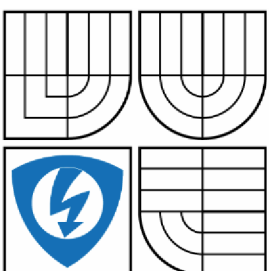


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SBĚR DAT A DETEKCE ANOMÁLIÍ PŘES MOBILNÍ ZAŘÍZENÍ

MOBILE BASED DATA ACQUISITION AND ANOMALY DETECTION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

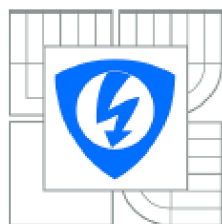
Bc. MICHAEL ONDRÁŠEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR HONZÍK, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Michael Ondrášek

ID: 133138

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Sběr dat a detekce anomálií přes mobilní zařízení

POKyny PRO VYPRACOVÁNÍ:

Cílem diplomové práce je navrhnout a realizovat systém pro měření, bezdrátový sběr, archivaci, zpracování a vizualizaci dat využívající běžně dostupná mobilní zařízení jako hlavní komunikační a vizualizační rozhraní. Při realizaci se zaměřte na problematiku detekce anomálií. Navrhněte pro tyto účely jednoduchý laboratorní přípravek vhodný pro testování jak funkčnosti celého systému, tak prezentování výstupů implementovaných algoritmů pro detekci anomálií.

1. Seznamte se s algoritmy používanými pro detekci anomálií, zpracujte rešerši.
2. Realizujte jednoduchý laboratorní přípravek vhodný pro generování dat, na kterých bude možné testovat metody pro detekci anomálií. Jako součást přípravku vytvořte modul umožňující měření, sběr a bezdrátový přenos dat.
3. Realizujte serverovou část systému sloužící k archivaci naměřených dat, jejich zpracování a vyhodnocení. Implementujte vybrané algoritmy pro detekci anomálií včetně vizualizace výsledků.
4. Pod OS Android realizujte mobilní aplikaci pro komunikaci a přenos dat mezi měřicím modulem a serverem a vizualizaci výsledků analýz realizovaných na straně serveru.
5. Navrhněte na přípravku experimenty, na kterých bude možné prezentovat jak funkčnost celého systému, tak vybrané algoritmy pro detekci anomálií.

DOPORUČENÁ LITERATURA:

Chandola V. et. al.: Outlier detection: a survey. Technical report, University of Minnesota, 2007.

Termín zadání: 9.2.2015

Termín odevzdání: 18.5.2015

Vedoucí práce: Ing. Petr Honzík, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

Abstrakt

Práce se zabývá realizací systému se specifickou architekturou pro detekci anomálií ve výuce nebo v komerčním použití. Systém se skládá ze tří částí: měřící modul, mobilní aplikace a serverová část. Přenos mezi měřícím modulem a serverem pro vyhodnocení je realizován současně s vizualizací na mobilním zařízení. Všechny části systému jsou realizovány s ohledem na minimální náklady a maximální rozšiřitelnost. Veškerý potřebný výpočetní výkon je soustředěn do serverové části z důvodu použitelnosti s více i současně pracujícími mobilními klienty. Důraz je kladen na architekturu řešení a možnost použití systému jako celku, nebo některých vybraných částí samostatně. Závěrem jsou navrženy experimenty pro prezentaci vybraných metod detekce anomálií.

Klíčová slova

detekce anomálií, Java, Android, REST, Bluetooth, OOP, Spring, Ibatis

Abstract

The work deals with the implementation of the specific architecture to detect anomalies in the classroom or in commercial use. The system consists of three parts: Measurement module, mobile applications and server part. Transmission between the measuring module of the server and the evaluation is carried out simultaneously with the visuals on the mobile device. All system components are implemented with the minimum cost and maximum expandability. All the necessary computing power is concentrated in the server part because of usability with multiple simultaneously operating mobile clients. Emphasis is placed on the solution architecture and the possibility of using the system as a whole, or selected portions separately. Finally, experiments are designed for the presentation of selected methods for anomaly detection.

Keywords

anomaly detection, Java, Android, REST, Bluetooth, OOP, Spring, Ibatis

Bibliografická citace:

ONDRÁŠEK, M. Sběr dat a detekce anomálií přes mobilní zařízení. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 69 s. Vedoucí diplomové práce Ing. Petr Honzík, Ph.D..

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Sběr dat a detekce anomálií přes mobilní zařízení jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 18. května 2015

.....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Petru Honzíkovi, Ph.D. a Ing. Ondřeji Hynčicovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: 18. května 2015

.....
podpis autora

Obsah

1	Úvod.....	11
2	Teoretický rozbor.....	12
2.1	Co je to Anomálie	12
2.2	Typy anomálií	14
2.3	Vyhodnocování	14
2.4	Výstup detekce anomálií	15
2.5	Extrakce příznaků.....	15
2.5.1	Hodnota spektra s největší amplitudou.....	16
2.5.2	Hodnota maximální složky spektra.....	17
2.5.3	Celkové harmonické zkreslení signálu	17
2.5.4	Hodnota součtu složek celého spektra	18
2.5.5	Součet amplitud spektra rozdělený do x částí.....	18
2.5.6	Odchylka signálu od jeho klouzavého průměru	18
2.5.7	Počet průchodu signálu nulou.....	19
2.5.8	Průměrná hodnota signálu	19
2.6	Klasifikační metody	20
2.6.1	Neuronové sítě	21
2.6.2	Bayesovské sítě.....	21
2.6.3	Support Vector Machines	21
2.6.4	Rozhodovací pravidla	22
2.7	Metody pro detekci a jejich využití.....	22
2.7.1	Nejbližší soused	22
2.7.2	Shluková analýza	24
2.7.3	Statistické metody	25
2.7.4	Kontextové metody.....	28
2.8	Použití detekce anomálií	28
2.8.1	Detekování narušení/vniknutí	28
2.8.2	Detekce podvodů	28
2.8.3	Oblast zdravotnictví	29
2.8.4	Průmyslové využití	29
2.8.5	Počítačové vidění	29
3	Návrh systému	30
4	Laboratorní přípravek	32
4.1	Požadavky	32
4.2	Technický návrh.....	32
4.3	Modul pro sběr dat	32
4.3.1	Použité technologie.....	35
4.3.2	Software	35
4.4	Realizace	37
5	Serverová část.....	39

5.1	Požadavky	39
5.2	Návrh.....	39
5.2.1	Použité technologie a nástroje	44
5.2.2	Datový model.....	45
5.2.3	Objektový návrh	47
5.3	Realizace	48
6	Mobilní aplikace	52
6.1	Požadavky	52
6.2	Návrh.....	52
6.2.1	Použité technologie.....	52
6.2.2	Objektový návrh	53
6.2.3	GUI	53
6.3	Realizace	54
7	Realizované experimenty.....	57
7.1	První experiment – klasifikace.....	58
7.2	Druhý experiment – míra odlišnosti.....	61
8	Závěr	63

Seznam obrázků

Obrázek 1 Ukázka anomálních hodnot.....	12
Obrázek 2 Typy anomálií	14
Obrázek 3 Ukázka frekvenčního spektra	16
Obrázek 4 Hlavní frekvenční složka.....	17
Obrázek 5 Ukázka odchylky signálu od jeho ideálního tvaru	18
Obrázek 6 Zero crossing [38]	19
Obrázek 7 Klasifikace do pěti tříd	21
Obrázek 8 Grafické znázornění části výpočtu LOF [7].....	24
Obrázek 9 Normalizované Gausovy funkce [11]	26
Obrázek 10 Normální rozdělení [12]	27
Obrázek 11 Klasická koncepce pro sběr dat a vyhodnocení	30
Obrázek 12 Navržená koncepce pro sběr dat a vyhodnocení	30
Obrázek 13 Sekvenční digram návrhu řešení	31
Obrázek 14 Zrealizovaný přípravek pro sběr dat.....	32
Obrázek 15 Modul s akcelerometrem ADXL345 [20][35]	33
Obrázek 16 Freescale KL25Z Kinetis KL2x MCU [21]	34
Obrázek 17 Bluetooth modul HC-06 [19]	35
Obrázek 18 Ukázka skutečné komunikace pro I ² C sběrnici	35
Obrázek 19 Diagram smyčky modulu pro sběr dat	37
Obrázek 20 Měřicí modul	38
Obrázek 21 Porovnání odezvy podle komunikačního protokolu [40].....	40
Obrázek 22 Struktura frameworku Spring [43]	41
Obrázek 23 MVC	42
Obrázek 24 Schéma rozložení aplikace do vrstev	43
Obrázek 25 Ukázka prostředí Netbeans IDE	44
Obrázek 26 Návrh datového modelu	46
Obrázek 27 Ukázka objektového modelu pro část DAO vrstvy.....	47
Obrázek 28 Struktura tříd serverové aplikace.....	48
Obrázek 29 Knihovny třetích stran integrované do projektu.....	49
Obrázek 30 Aplikace SoapUI pro testování komunikace.....	50
Obrázek 31 Ukázka vygenerované dokumentace Javadoc k aplikaci	51
Obrázek 32 Objektový model mobilní aplikace	53
Obrázek 33 Ukázka uživatelského rozhraní	54
Obrázek 34 Nastavení API Android Studia	55
Obrázek 35 Struktura projektu pro mobilní telefon.....	56
Obrázek 36 Ukázka naměřených průběhů a FFT pro jedno měření v prostředí Matlab	57
Obrázek 37 Vizualizace vypočtených příznaků pro 180 měření v Matlab.....	58
Obrázek 38 Model v RM pro k-NN.....	60
Obrázek 39 Detail bloku Forward Selection.....	60
Obrázek 40 Detail bloku Validation	60

Obrázek 41 Výsledek výpočtu k-NN pro 5 příznaků	60
Obrázek 42 Model pro LOF.....	61
Obrázek 43 Výsledek metody LOF	62

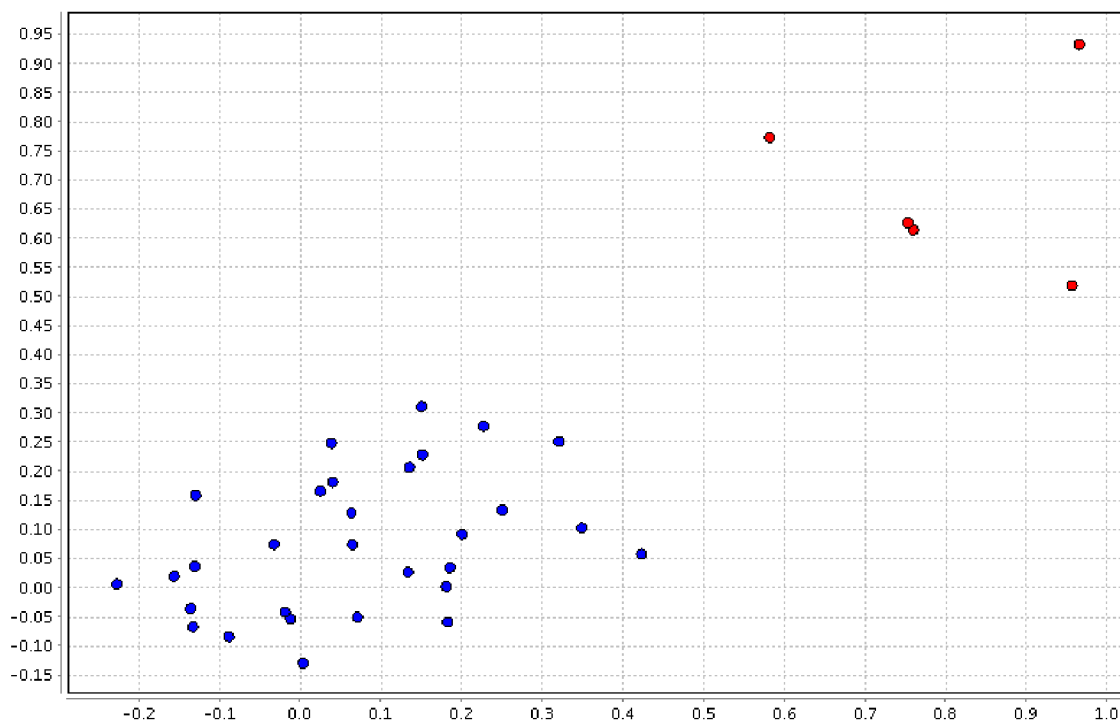
1 ÚVOD

Detekce anomálií, která bývá často v literatuře označována jako detekce odlehlých hodnot, se v dnešní době uplatňuje v mnoha oborech lidské činnosti. Od odhalování podezřelých transakcí v bankovníctví, přes detekci narušení v počítačových sítích až po detekci změn chování průmyslových zařízení. Posledně věnované problematice se věnuje tato práce. Po teoretickém úvodu, kde jsou popsány různé typy anomálií, příprava dat pro zpracování, generování příznaků a jejich následné vyhodnocení následuje část praktická. Jejím cílem je navrhnout a zrealizovat laboratorní přípravek pro generování dat využitelných pro další zpracování jako je například detekce anomálií. Naměřená data se budou bezdrátově přenášet z měřicího modulu do mobilního zařízení (mobilního telefonu, tabletu) na platformě Android. Vlastní výpočty detekce anomálií budou probíhat na serveru z důvodu jejich výpočetní náročnosti. Mobilní zařízení bude s tímto serverem obousměrně komunikovat. Přenášet na něj data naměřená na přípravku a zpětně zobrazovat výsledky zpracování. Na těchto realizovaných částech budou navrženy experimenty pro prezentaci celého systému pro přenos naměřených dat včetně vybraných realizovaných algoritmů pro detekci anomálií.

2 TEORETICKÝ ROZBOR

2.1 Co je to Anomálie

„Anomálie je definovaná jako vzor, který se odlišuje od očekávaného normálního chování“ [2]. Přístup k detekci anomálií spočívá v definování oblastí reprezentujících normální chování a prohlášení, že jakákoliv vyhodnocená data, která nepatří do těchto normálních oblastí, označíme jako anomálie.



Obrázek 1 Ukázka anomálních hodnot

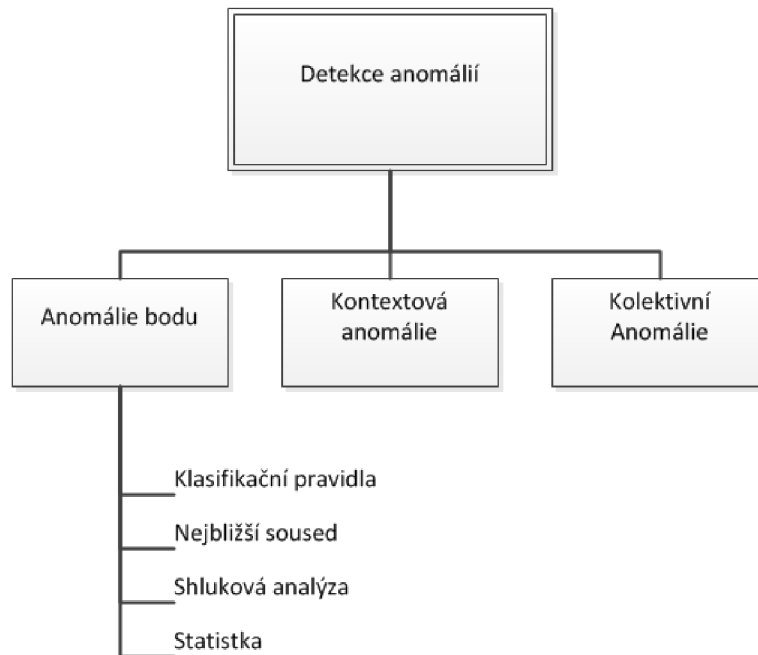
Tento na první pohled jednoduchý proces je díky různým faktorům [1] často hodně náročný:

- Definování normální oblasti, která zahrnuje všechna možná normální chování, je velmi obtížné. Navíc hranice mezi normálním a anomálním chováním je často obtížně definovatelná. Chování, které leží blízko této hranice, může být interpretováno jako normální i anomální a tím dochází ke snížení přesnosti detekce.
- V případě, že jsou anomálie výsledkem úmyslných škodlivých činností, škodu způsobující subjekty se často přizpůsobují proto, aby jejich anomální chování nebylo rozpoznáno a pohybovali se na mezi normálních hodnot. Tímto chováním se úkol definovat normální chování stává obtížnější.

- Sledované systémy jsou typicky dynamické a v čase se mění. Většinou je výskyt normálního chování reprezentováno většinou dostupných dat, nicméně aktuální normální chování nemusí být dostatečně reprezentativní v budoucnu.
- Význam míry odlišnosti se může lišit u různých aplikačních oblastí. V lékařství i malá odchylka teploty od normálu může být anomálie, zatímco stejná odchylka ceny na akciovém trhu může být považována za normální. Použití techniky detekce anomálií vyvinuté v jedné oblasti nemusí být vhodné pro detekci v oblasti jiné.
- Nedostupnost dostatečného množství klasifikovaných dat pro učení/validaci modelů používaných pro detekce anomálií.
- Data často obsahují šum, který má tendenci se podobat skutečné anomálii, a tudíž je obtížné ho detekovat a eliminovat.

Vzhledem k uvedeným problémům není snadné problém detekce anomálií v jeho obecné formě vyřešit. Ve skutečnosti většina existujících technik detekce anomálií řeší specifické definice problému. Definice je reprezentována různými faktory jako jsou povaha dat, dostupnost označených dat, typy anomálií, které mají být detekovány, atd. Výsledkem výzkumu v této oblasti pak bylo zavedení řady konceptů pocházejících z různých oborů jako jsou statistika, strojové učení, dolování dat, informační teorie, spektrální teorie a aplikuje je k řešení specifických problémů.

2.2 Typy anomálií



Obrázek 2 Typy anomálií

- Anomálie bodu. V případě, že hodnota neodpovídá zbytku dat, je tato hodnota označena jako anomální. Jedná se o nejjednodušší způsob detekce anomálií, který lze použít pouze v případě, že data neobsahují vzájemnou závislost na pořadí nebo čase. Příkladem je vyhodnocování naměřených dat např. hodnota odporu připojeného k měřicímu přístroji.
- Kontextová (podmíněná) anomálie. Jedná se o anomálii, kde je vyhodnocení zda se jedná nebo nejedná o anomální hodnotu v určitém kontextu s ostatními daty. Jako příklad lze uvést např. závislost teploty nebo intenzity světla na denní době. Kontext dat musí být určen v datovém souboru, aby ho bylo možné vyhodnotit. Atributy zpracovávaných dat se dělí na dvě skupiny: kontextové atributy a behaviorální atributy. V některých případech není možné kontext snadné identifikovat a popsat a vyhodnocení anomálie se realizuje jinou metodou.
- Kolektivní anomálie je rozšířená metoda kontextové anomálie. Oproti ní se ale vyhodnocuje kolekce hodnot, zda je v souladu s kontextem. Typickým příkladem je např. vynechání části, nebo celé periody v jinak periodickém signálu.

2.3 Vyhodnocování

- S učitelem (supervised), kdy obsluha označí anomální hodnoty v souboru dat. Vyhodnoceny budou jako anomální pouze ty hodnoty, které byly označeny.

Hrozí zde riziko neodhalených anomálií, které nebyly v označovaném souboru dat.

- Částečně s učitelem (semisupervised). V tomto případě obsluha označí normální hodnoty. Což znamená, že všechny odlišné hodnoty jsou vyhodnoceny jako anomální.
- Bez učitele (unsupervised), kde musí detekční systém sám vyhodnotit, zda a které hodnoty v souboru dat jsou anomální. Nejčastěji se používají statistické metody, kdy se předpokládá větší výskyt normálních hodnot než anomálních. Tím, že není potřeba zásah učitele, jsou tyto systémy velmi rozšířené a autonomní.

2.4 Výstup detekce anomálií

Důležitým hlediskem pro zjišťování anomálií je způsob, jakým jsou anomálie vyhodnoceny. Obvykle [1] jsou výstupy metod detekce anomálií jedním z těchto dvou typů:

- Míra (skóre)

Výsledkové metody přiřazující anomálním hodnotám hodnotu, která je závislá na míře, ve které je tento testovaný případ považován za anomálii. Výstupem takových metod je zpravidla seznam testovaných případů seřazených podle míry výsledné anomálie. Z takto seřazených případů lze definováním prahové míry vybrat množinu anomálních případů.

- Klasifikace do tříd

Metody v této kategorii přiřazují klasifikační třídu normální nebo anomální každé testované instanci. Tyto metody detekce anomálií umožňují definovat konkrétní třídu k detekci nejvýznamnějších anomálií. Metody s binární klasifikací toto definování neumožňují. Nastavení mezí detekce se provádí nepřímo volbou parametrů metody.

2.5 Extrakce příznaků

Pro extrakci příznaků jsou použity metody, které získávají data z vlastního průběhu signálu nebo z jeho frekvenčního spektra. Pro výpočet spektra je použita Fourierova transformace [14][15].

$$c_m = \frac{1}{N} \sum_{k=0}^{N-1} f(k) e^{-jm \frac{2\pi}{N} k}$$

kde m je index koeficientu $m = 0, 1, 2, \dots, N - 1$

c_m je váhový koeficient

f_k je signál

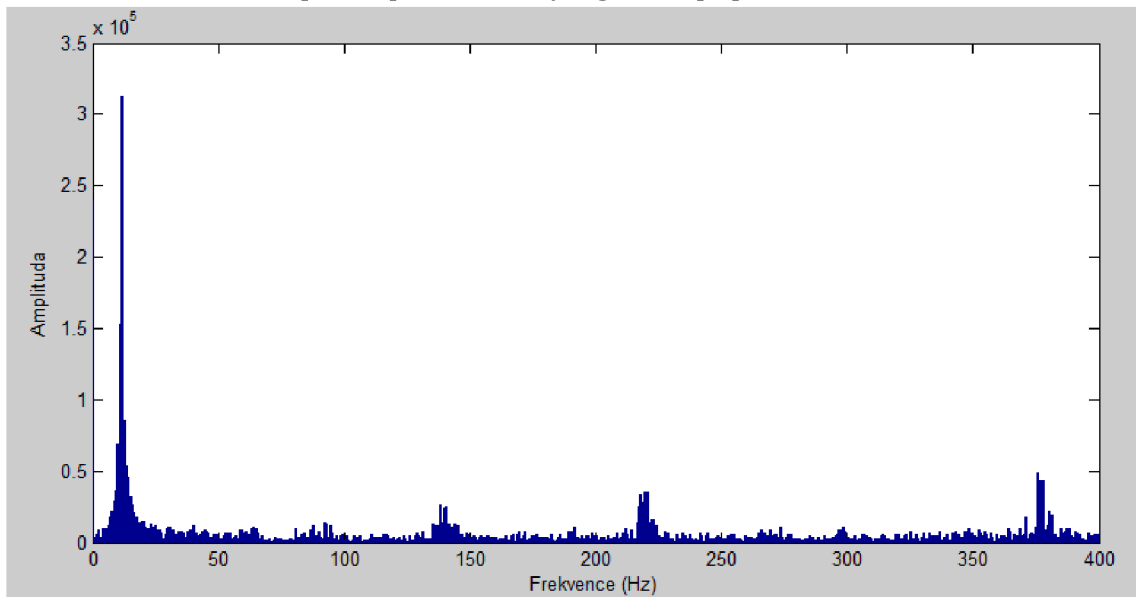
k je pořadové číslo vzorku

Vzhledem k tomu, že koeficienty c_m jsou komplexní čísla, tak pro další zpracování pracujeme s jejich amplitudou $|c_m|$.

Pro výpočet diskrétní Fourierovy transformace lze použít různé algoritmy.

- Cooley-Tukey – nepoužívanější algoritmus typu rozděl a panuj, kdy jsou vstupní data rozdělována rekurzivně vždy na poloviny. Některé z praktických implementací tohoto algoritmu již nevyužívají explicitní rekurzi.
- Algoritmy, které nevyužívají princip Cooley-Tukey jako jsou např.: Prime-factor algoritmus, Bruun algoritmus, Rader algoritmus a Bluestein algoritmus.

Ukázka frekvenčního spektra pro naměřený signál na přípravku:

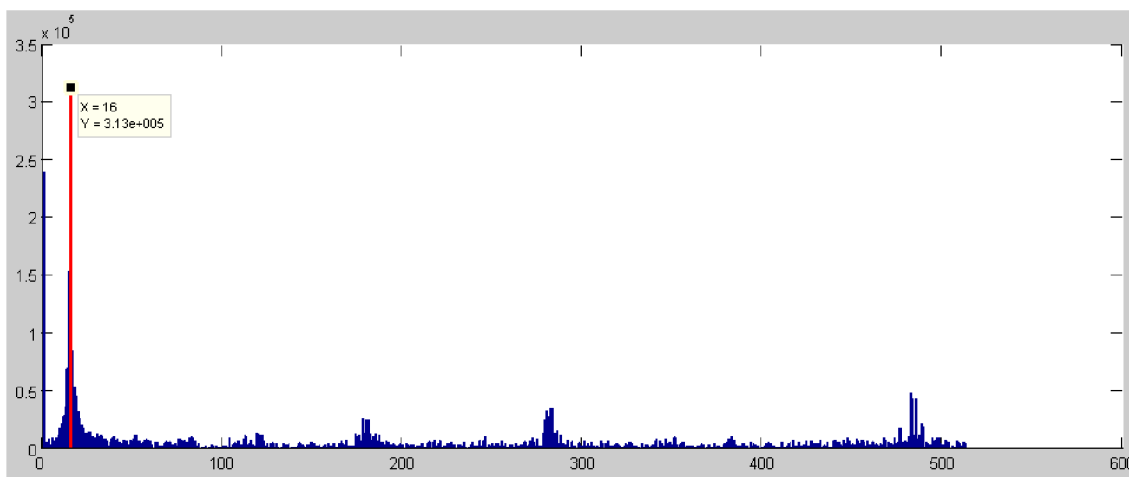


Obrázek 3 Ukázka frekvenčního spektra

2.5.1 Hodnota spektra s největší amplitudou

Vyhodnocení probíhá z frekvenčního spektra. Ze spektra signálu určíme pořadí složky s maximální hodnotou.

$$\max(A_n) = \sqrt{a_n^2 + b_n^2}$$



Obrázek 4 Hlavní frekvenční složka

Postup algoritmu výpočtu:

Ze všech hodnot spektra se vybere ta frekvenční složka, jejíž hodnota má největší amplitudu. Jako příznak se použije frekvence této vybrané složky. Nepočítá se s hodnotou stejnosměrné složky.

2.5.2 Hodnota maximální složky spektra

Vyhodnocení probíhá z frekvenčního spektra. Jedná se o nejvyšší hodnotu složky spektra.

$$\max(A_n) = \sqrt{a_n^2 + b_n^2}$$

Postup algoritmu výpočtu:

Ze všech hodnot spektra se vybere ta frekvenční složka, jejíž hodnota má největší amplitudu. Nepočítá se s hodnotou stejnosměrné složky.

2.5.3 Celkové harmonické zkreslení signálu

Vyhodnocení probíhá z frekvenčního spektra.

$$THD_u = \frac{\sqrt{U_2^2 + U_3^2 + U_4^2 + U_5^2 + \dots}}{U_1} \cdot 100$$

Postup algoritmu výpočtu:

Ze všech hodnot spektra se vybere ta frekvenční složka, jejíž hodnota má největší amplitudu. Nepočítá se s hodnotou stejnosměrné složky. Tato frekvenční složka se bere jako hlavní. Zjistí se, kolik dalších násobků hlavní složky je obsaženo ve spektru. Postupně se tyto násobky prochází a sčítají mocniny jejich hodnot. Tento součet se odmocní a vydělí hodnotou hlavní složky. Výsledek se násobí hodnotou 100.

2.5.4 Hodnota součtu složek celého spektra

Vyhodnocení probíhá z frekvenčního spektra.

$$A_n = \sum_{n=1}^{max} \sqrt{a_n^2 + b_n^2}$$

Postup algoritmu výpočtu:

Prochází se frekvenční složky mimo stejnosměrné a jejich hodnoty se sčítají.

2.5.5 Součet amplitud spektra rozdělený do x částí

Vyhodnocení probíhá z frekvenčního spektra. Výsledkem je součet hodnot spektra rozděleného na 10 úseků.

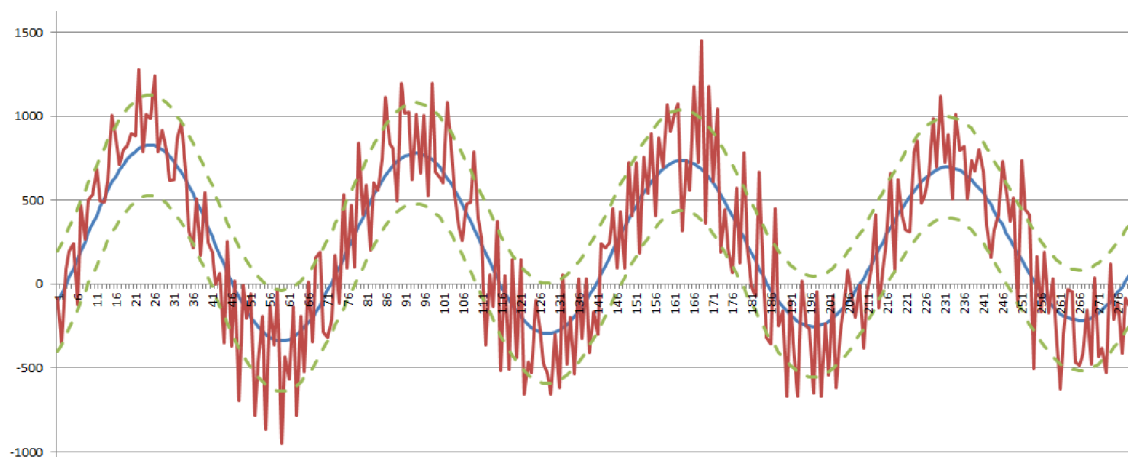
$$A_n = \sum_{n=0}^{max/10} \sqrt{a_n^2 + b_n^2}$$

Postup algoritmu výpočtu:

Nejdříve se zjistí počet hodnot ve frekvenčním spektru. Tento se rozdělí na 10 částí a v jednotlivých částech se sečtou jeho hodnoty.

2.5.6 Odchylka signálu od jeho klouzavého průměru

Vyhodnocení probíhá přímo ze změřených hodnot. Výsledkem je počet, kolikrát signál opustí definované okolí klouzavého průměru.



Obrázek 5 Ukázka odchylky signálu od jeho ideálního tvaru

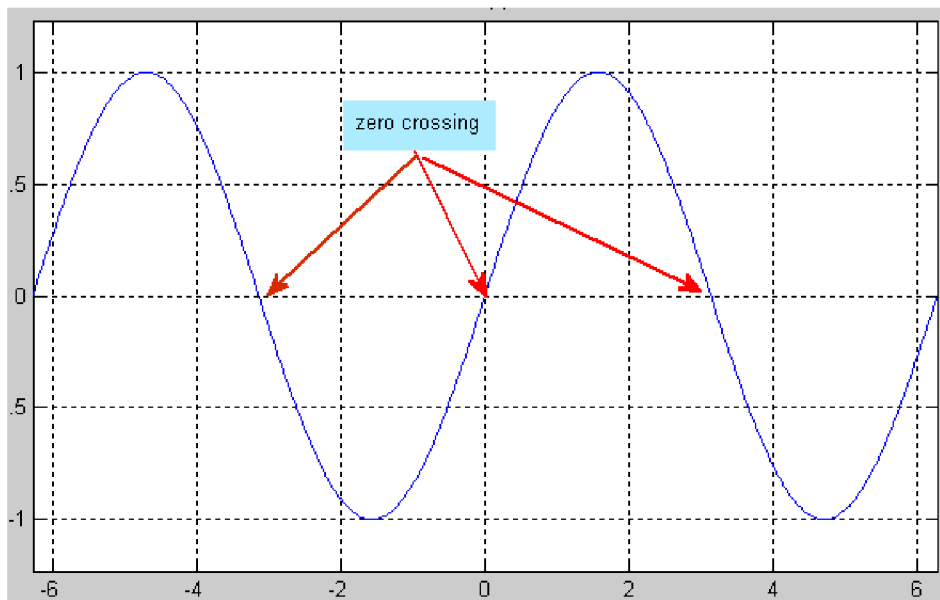
Postup algoritmu výpočtu:

Definuje se počet hodnot pro klouzavý průměr a okolí měřeného signálu. Na základě počtu hodnot se spočte hodnota klouzavého průměru. V případě že aktuální hodnota je mimo definované okolí, přičteme do příznaku jedničku.

2.5.7 Počet průchodu signálu nulou

Vyhodnocení probíhá přímo ze změřených hodnot. Výsledkem je počet, kolikrát signál překročil (oběma směry) nulovou hodnotu [37].

$$zcr = \frac{1}{T-1} \sum_{t=1}^{T-1} I_{\{s_t s_{t-1} < 0\}}$$



Obrázek 6 Zero crossing [38]

Postup algoritmu výpočtu:

Nejprve spočteme průměrnou hodnotu signálu jako součet všech hodnot dělený počtem. Následně procházím všechny naměřené hodnoty a v případě, že minulá hodnota je pod průměrnou hodnotou a aktuální nad průměrnou hodnotou přičteme do příznaku jedničku. Stejně přičteme jedničku i v opačném případě.

2.5.8 Průměrná hodnota signálu

Vyhodnocení probíhá přímo ze změřených hodnot.

$$s_{prumer} = \frac{1}{T} \sum_{t=1}^T s_t$$

Postup algoritmu výpočtu:

Spočteme průměrnou hodnotu signálu jako součet všech hodnot podělený počtem.

2.6 Klasifikační metody

Klasifikace je proces, kdy na základě tréninkových dat naučený model klasifikuje nová data do jedné ze tříd. Klasifikace, pro použití v metodách detekce anomálií, pracuje v dvoufázovém postupu. V učicí fázi se učí model na základě dostupných definovaných tříd. Ve druhé, testovací fázi, se data označují jako normální nebo anomální podle výsledné klasifikace do tříd.

Klasifikátor může pracovat pouze v takovém prostoru příznaků, na které byl naučen.

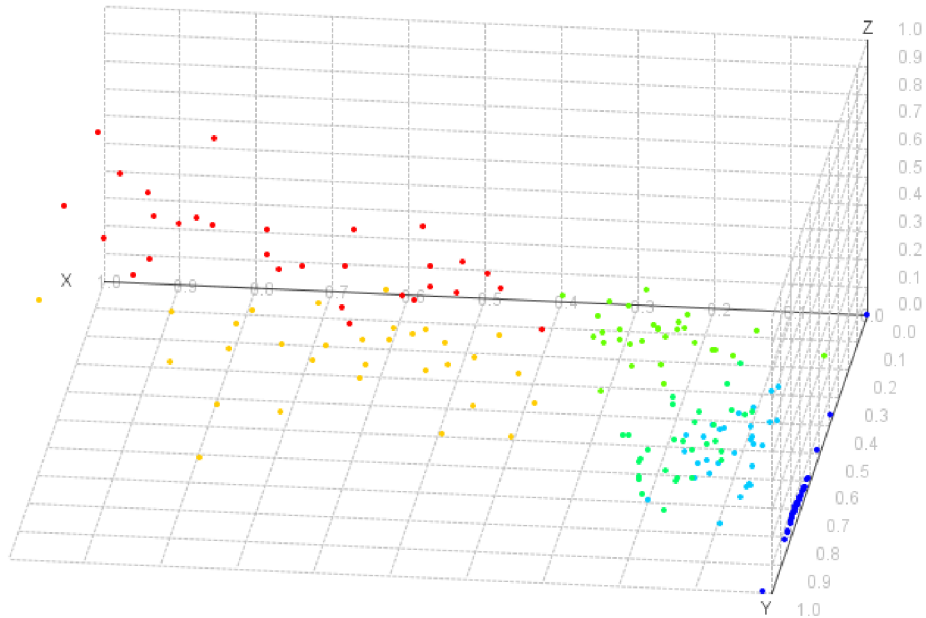
Na základě množství klasifikačních tříd pro učicí fázi rozlišujeme dvě základní kategorie:

- Klasifikace do více tříd u metod detekce anomálií vychází z předpokladu, že učicí data obsahují definované případy patřící do několika běžných tříd. Tyto metody detekce anomálií učí klasifikátor rozlišovat mezi normálními třídami a těmi zbývajících (anomálními). Vybrané metody klasifikace do více tříd pracují způsobem, že pokud žádný z klasifikátorů neklasifikuje případ jako normální, potom je tento případ prohlášen za anomální.
- Klasifikace do jedné třídy u metod detekce anomálií předpokládá, že všechny zkušební případy mají pouze jednu definovanou třídu. Taková metoda má diskriminační hranici okolo normálních případů definovanou použitím klasifikačního algoritmu. Jakýkoliv testovaný případ, který nespadá do oblasti definované touto hranicí, je označen jako anomální.

Výpočetní složitost klasifikačních technik je závislá na klasifikačním algoritmu. Použití rozhodovacích stromů je rychlejší než jiné metody. [2] Výhody a nevýhody klasifikačních technik [1]:

- Výhody
 - Techniky klasifikace do více tříd, mohou využívat silný algoritmus, který může rozlišovat mezi případy patřícími k různým třídám ne pouze normální/anomální.
 - Testovací fáze je rychlá, protože každý testovaný případ je porovnáván s vypočítaným modelem.
- Nevýhody

Techniky klasifikačních metod předpokládají přesné definování pro různé třídy, což často není možné.



Obrázek 7 Klasifikace do pěti tříd

2.6.1 Neuronové sítě

Použití neuronové sítě je pro klasifikaci do více tříd. Realizuje se jako dvou krokové. V prvním kroku je síť naučena a v druhém potom po přivedení testovacích dat umí klasifikovat do tříd, na které nebyla naučena.

2.6.2 Bayesovské sítě

Jsou používány pro detekci anomálií s klasifikací do více tříd. Metoda na základě naivního Bayesovské sítě odhaduje na základě rozložení, zda se data vyskytují v oblasti normálních nebo anomálních hodnot. Provádí se výběr atributů, které popisují vstupní data a nad těmito atributy se provádí klasifikace. Přesnost odhadování je závislá na testovacích datech. U těchto dat se předpokládá, že jsou vzájemně nezávislá.

2.6.3 Support Vector Machines

Je používána pro detekci do jedné třídy. Pracuje jako lineární klasifikátor. Využívá se princip, kdy vstupním datům přidáme dimenze tak, aby šla data lineárně separovat. V případě, že testovaná data následně patří do definované oblasti, jsou označena jako

normální. Další varianta hledá hyper koule ve vícerozměrném prostoru a určuje, zda testovaná data leží v těchto hyper koulích.

Ukázková aplikace využívající tuto metodu včetně vizualizace lze najít například zde: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

2.6.4 Rozhodovací pravidla

Jedná se o metodu, kdy jsou vytvořena pravidla pro normální hodnoty. V případě, že hodnoty neodpovídají nadefinovaným pravidlům, jedná se o anomální hodnoty. Základní metoda pro klasifikaci do více tříd je dvoufázová, kdy v první fázi se provádí učení, jehož výsledkem je např. rozhodovací strom. V druhé fázi se na rozhodovací strom aplikují neznámé hodnoty a výsledkem je informace zda hodnota je či není anomální.

2.7 Metody pro detekci a jejich využití

2.7.1 Nejbližší soused

Metoda nejbližšího souseda spoléhá na předpoklad, že normální hodnoty jsou si blízko, naopak anomální hodnoty jsou vzdálenější. Vyhodnocuje se tedy vzdálenost mezi jednotlivými hodnotami. Tuto vzdálenost lze počítat různými metodami. Některé metody počítají nejen se vzdáleností, ale i s polohou.

2.7.1.1 k-NN

Jedná se o základní metodu typu nejbližších sousedů. Velikost anomálie je vyjádřena na základě vzdáleností k nejbližším sousedům. Čím větší je tato vzdálenost, tím více je hodnota anomální. Před vlastním výpočtem je potřeba provést normalizaci vstupních hodnot.

Typy normalizací [36]:

- Lineární

$$x_{norm} = \frac{x - X_{min}}{X_{max} - X_{min}}$$

typické intervaly (0; 1), (-1; 1)

- Střední hodnotou a rozptylem

$$x_{norm} = \frac{x - X_{min}}{X_{max} - X_{min}}$$

při normálním rozložení 99% v intervalu -3; 3

- Logitovou funkcí

$$x_{norm} = \frac{x - X_{min}}{X_{max} - X_{min}}$$

interval (0; 1) nebo (-1; 1)

Způsoby výpočtu vzdáleností [4][36]:

- Euklidovská

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2}$$

- Hammingova

$$d(x_i, x_j) = \sum_{r=1}^n |a_r(x_i) - a_r(x_j)|$$

- Překrytí

$$d(x_i, x_j) = \sum_{r=1}^n [1 - \partial(a_r(x_i), a_r(x_j))]$$

- Kosínová

$$d(x_i, x_j) = \frac{\sum_{r=1}^n (a_r(x_i) \cdot a_r(x_j))}{\sqrt{\sum_{r=1}^n (a_r(x_i) \cdot a_r(x_i)) \cdot \sum_{r=1}^n (a_r(x_j) \cdot a_r(x_j))}}$$

Pro klasifikaci se počítá příslušnost podle vztahů [36]:

$$\hat{f}(x_q) = \arg \max_{g \in G} \sum_{i=1}^k \partial(g, f(x_i))$$

$$\partial(a, b) = 1, \text{ když } a = b, \text{ jinak } \partial(a, b) = 0$$

2.7.1.2 Relativní hustota [5]

Metoda detekující anomálii v případě, že v okolí je malá hustota jiných hodnot. Předpokládá se, že normální hodnoty jsou hustěji než anomální. Způsoby výpočtů[5]:

- Inverzní vzdálenost

$$density(x, k) = \left[\frac{\sum_{y \text{ in } N(x, k)} distance(x, y)}{|N(x, k)|} \right]^{-1}$$

kde $N(x, k)$ je množina k nejbližších sousedů x
 $|N(x, k)|$ je velikost množiny

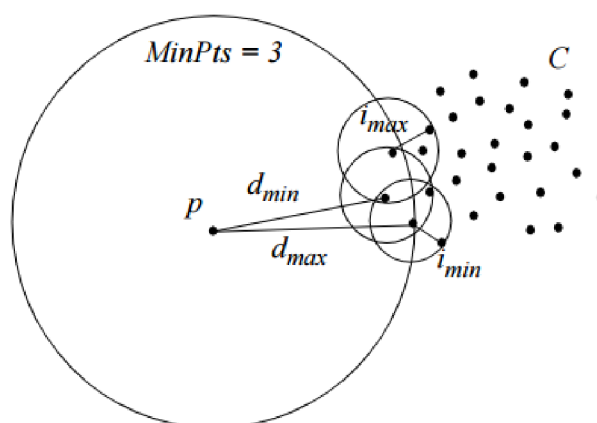
y je nejbližší soused

- Počet bodů v okruhu
 $density(x, \epsilon) = \text{počet objektů v } \epsilon \text{ vzdálenosti od } x$

- Průměrná relativní hustota [39]

$$avg_rel_density(x, k) = \frac{density(x, k)}{\left[\frac{\sum_{y \in N(x, k)} density(y, k)}{|N(x, k)|} \right]}$$

Metodami výpočtu odlehlosti pomocí algoritmů LOF (local outlier factor) se blíže zabývá článek [3] a [8].



Obrázek 8 Grafické znázornění části výpočtu LOF [7]

2.7.2 Shluková analýza

„Shluková analýza patří mezi metody učení bez učitele. Jejím cílem je v dané množině objektů nalézt její podmnožiny – shluky objektů – tak, aby si členové shluku byli navzájem podobní, ale nebyli si příliš podobní s objekty mimo tento shluk.“ [6]

Po stanovení typů shluků, kterých chceme dosáhnout, je dále nutné vybrat shlukovací algoritmus. Většinou je potřeba zařadit objekty do určitého počtu shluků nebo vytvoření hierarchie shluků. [23] Z tohoto hlediska se rozlišuje shlukování ploché (flat) a hierarchické. Pro ploché je také používán termín nehierarchické. Je potřeba stanovit, zda má být objekt zařazen k jednomu nebo do více shluků. U jednoho je používána informace o příslušnosti právě k jednomu shluku, pro více shluků je každému objektu přiřazena míra příslušnosti ke každému ze zadaného počtu shluků.

Z hlediska postupu rozlišujeme metody, které vycházejí z matice vzdáleností a metody vycházejí přímo ze vstupní datové matice. [2] Pro tyto dva typy se používají termíny: metody založené na vzdálenostech (distance-space methods) a metody vektorového

prostoru (vector-space methods). K prvnímu typu patří metody hierarchického shlukování a některé metody nehierarchické. Techniky vektorového prostoru jsou většinou používány ke shlukování objektů. Dělíme je na inkrementální (incremental), které procházejí postupně jednotlivé objekty a optimalizační, které vycházejí z počátečních shluků. Metody můžeme také různě kombinovat.

Při použití nehierarchického shlukování je potřeba zadání počtu shluků. Počet shluků často není známý. Je možné proto provádět shlukování pro různé počty shluků a z toho vyvodit optimální počet. K tomu bylo navrženo mnoho přístupů.

Pro měření podobnosti existuje mnoho koeficientů. Jejich použití závisí na tom, zda chceme shlukovat objekty nebo proměnné a také na typu proměnných.

2.7.3 Statistické metody

Z pohledu statistických metod je jako anomální hodnota označena t_a , jejíž výskyt má nízkou pravděpodobnost v souboru všech dat.

2.7.3.1 Gaus

Gausovo rozložení [10] umožňuje zjistit, v jaké vzdálenosti vyjádřené v násobcích směrodatné odchylky se vzorky nachází od průměru. Hodnoty od určitého násobku se označují jako anomální. Nejprve je potřeba zjistit statistické rozdělení vstupních dat. Gausovo rozložení má tvar zvonu. Definice obecné normalizované gausovy křivky [36]:

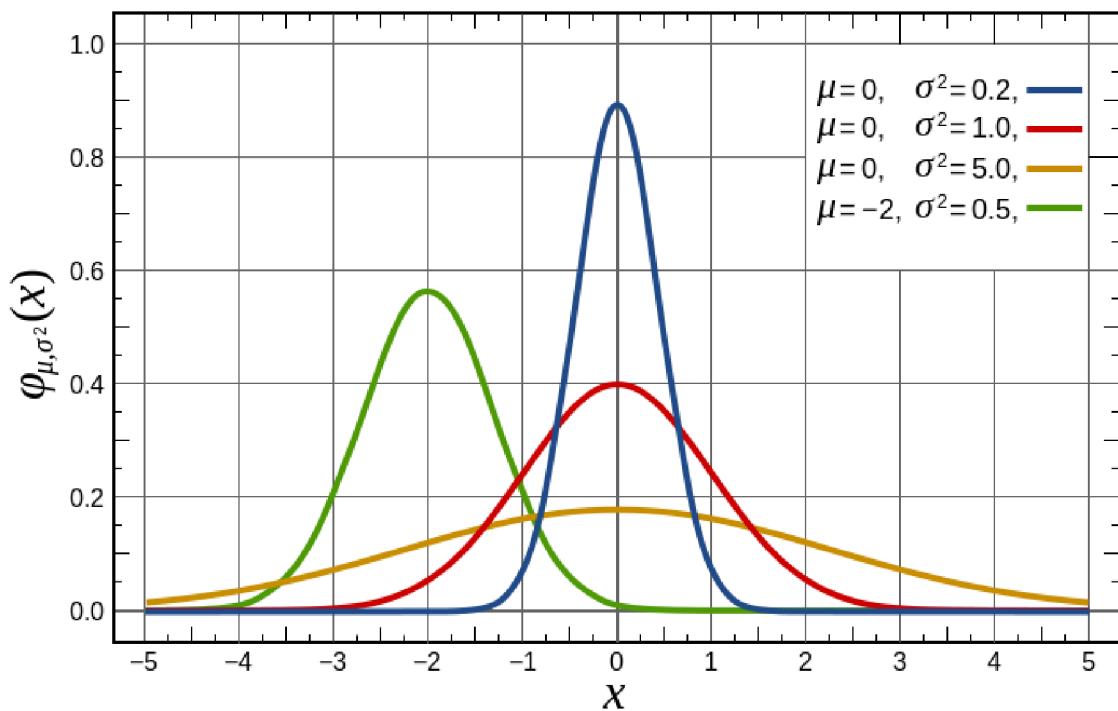
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]}$$

σ je směrodatná odchylka

σ^2 je rozptyl

μ je střední hodnota

e je Eulerovo číslo



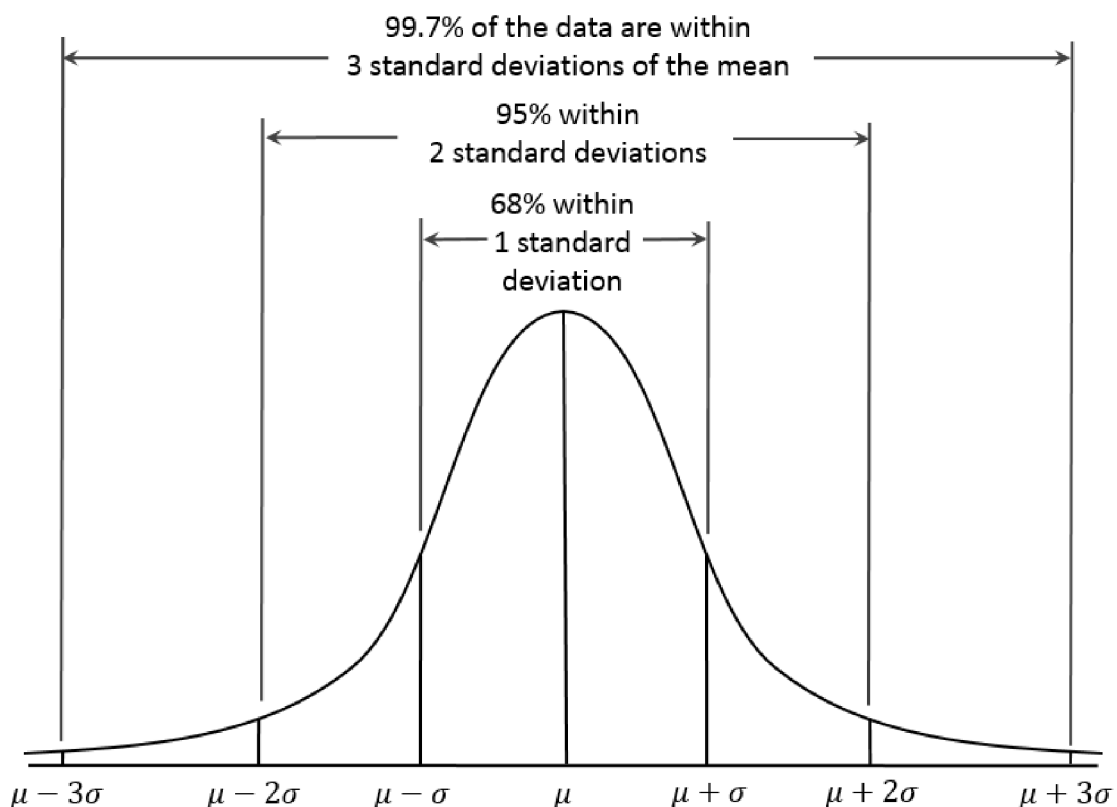
Obrázek 9 Normalizované Gausovy funkce [11]

Na následujícím obrázku je zobrazeno normální rozdělení s rozdělením podle vzdálenosti udané v počtu směrodatných odchylek. Zhruba 68% hodnot je do vzdálenosti jedné směrodatné odchylky, 95 % do dvou směrodatných odchylek a 99,7% do tří.

Data mohou být označena např. Gruberovým testem[1][9], pro výpočet hodnoty z-skóre.

$$z = \frac{x - \mu}{\sigma}$$

V případě že z-skóre má hodnotu větší jak 3 hovoříme o anomální hodnotě.



Obrázek 10 Normální rozdělení [12]

2.7.3.2 Regresní model

Regresní model slouží k detekci anomálií v datových řadách. Je to dvou kroková metoda, kde v prvním kroku dochází k naučení regresního modelu a v druhém jsou testované vzorky porovnávány s modelem. Míra neshody definuje hodnotu skóre anomálie. Vzhledem k tomu, že anomálie se může vyskytovat již v datech, která jsou použita pro naučení modelu, je možné, že takto naučený model může vykazovat nepřesné výsledky. Aby k tomuto jevu nedocházelo, používá se tzv. robustní regrese. Robustní regrese umí „skrýt“ anomálie v prvním kroku při načítání regresního modelu a umožňují již v tomto kroku detekovat anomálie. Základní regresní modely byly navrženy pro zpracování vícerozměrných časových datových souborů na principu klouzavých průměrů.

2.7.3.3 Histogramy

Metoda využívá model, který je vytvořen ze zpracovávaných dat. V této dvou krokové metodě je v prvním kroku na základě tréninkových dat vypočten histogram. V druhém kroku jsou vstupní data porovnávána, zda odpovídají tréninkovému histogramu. Pokud ne jsou označena jako anomální. Pokud trénovací data obsahují velké množství dat, může dojít k tomu, že bude odhaleno méně anomálií a naopak. Kvůli tomu je potřeba

správně určit množství dat pro trénovací množinu tak, aby množství falešných anomálií a falešných správných výsledků bylo vyvážené. Histogramy jsou ve většině metod [2] v prvním kroku vytvořeny na základě normálních daty. Jiné metody naopak vytváří testovací histogram z anomálních dat a zpracování výsledků je potom opačné. Histogramy lze využít i pro zpracování vícerozměrných dat, kdy výsledné skóre je součet jednotlivých skóre.

2.7.4 Kontextové metody

Algoritmy pro detekci kontextové anomálie vyžadují, aby zpracovávané údaje obsahovaly kontextové atributy (k definování kontextu), a soubor atributů chování (pro detekci anomálie v rámci kontextu). Některé ze způsobů, kterými kontextové atributy mohou být definovány, jsou: prostorové atributy, hrany, sekvence nebo profily. Většina technik uvedených v [1] usiluje o převod z hledání anomálií v kontextu na hledání anomálií bodu.

2.8 Použití detekce anomálií

V následujících kapitolách je uveden výběr aplikací detekce anomálií [2].

2.8.1 Detekování narušení/vniknutí

Oblast použití je zejména v IT, kde slouží k odhalení nebezpečného chování zpravidla v počítačové síti. Může se jednat o působení trojských koní, virů, nebo např. pokusy o stahování velkého množství dat.

Detekování narušení lze provádět i na základě sledování běhu operačního systému počítače. Princip spočívá v detekování posloupností příkazů, které jsou nestandardní.

Dalším využitím je detekce narušení bezpečnosti síťového provozu. V těchto případech je nutné detekovat kolektivní anomálie. Problémem je změna chování a úpravy algoritmů útočníka poté co byl odhalen, nebo zjistil, že jeho chování je detekováno. Je potřeba neustále aktualizovat a upřesňovat algoritmy pro detekci.

2.8.2 Detekce podvodů

Primárně určená pro detekci podvodů s krádeží identity a stahováním dat, ke kterým nemá běžně uživatel přístup. Oblast použití je hlavně v bankách a pojišťovnách. Jedná se hlavně o tyto podvody:

Podvody s kreditními kartami, kdy lze na základě předchozího chování, nebo jiného vstupu odhalit anomální transakce.

Detekce pojistných podvodů. V tomto případě je vyhodnocována např. zvýšená četnost výskytů stejných osob nebo stejného majetku.

Kontrola transakcí na burzách. Zde se provádí detekce časových souvislostí, kdy by obchodní transakce neměly nastat před uveřejněním tajných obchodních informací,

které na tyto obchody mají vliv. Slouží primárně k ochraně obchodních společností a burzy jako takové.

Detekce neoprávněného stahování informací. Zpravidla se jedná o veřejné zdroje, kde pomocí „robotů“ se pokouší útočníci načíst veškerá publikovaná data. Např. registr ekonomických subjektů, katastrální úřady a podobně.

2.8.3 Oblast zdravotnictví

V oblasti zdravotnictví slouží detekce anomálií k podpoře stanovení diagnózy. Jedná se o proces, kdy je potřeba velmi důkladně zvážit přínosy a důsledky použití automatického zpracování. Využívá se proto vyhodnocování s částečným učitelem. Další využití je ve zpracování výsledků dlouhodobých vyšetření jako aktivit mozku, činnosti srdce, nebo krevního tlaku.

2.8.4 Průmyslové využití

Primární snahou je detekovat úroveň opotřebení zařízení tak, aby nedošlo k havarijním stavům. V případě zjištění anomálního chování je potřeba provést vyhodnocení a učinit opatření k eliminaci tohoto chování. Této oblasti se věnuje praktická část diplomové práce.

- Detekce poruch mechanických komponent. U detekce může a nemusí být znám obsluze aktuální stav zařízení. Jedná se tedy o možné učení s nebo bez učitele. Podle toho je potřeba správně zvolit detekční metodu. Jedná se zpravidla o kontextové anomálie.
- Detekce konstrukčních vad. Slouží k odhalení anomálií ve strukturách jednotlivých materiálů, nebo celých konstrukčních celků.

2.8.5 Počítačové vidění

Slouží k detekci anomálních objektů v obrazu nebo k detekci pohybu, případně anomálního pohybu. Tyto systémy jsou realizovány v místech, kde je například velký počet lidí (letištní terminály, nádraží, muzea), nebo kde je potřeba sledovat provoz (tunely, dálnice). Vyhodnocení probíhá buď po jednotlivých snímcích v určitém časovém intervalu, nebo kontinuálně v reálném čase.

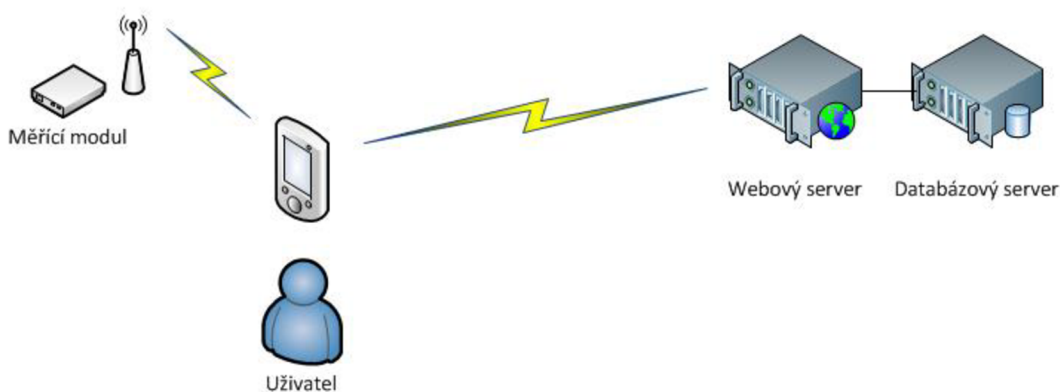
3 NÁVRH SYSTÉMU

Koncepční návrh celého systému vytvořeného v rámci řešení diplomové práce se opírá o požadavek na mobilitu a jednoduchost při použití. Pro výpočet detekce anomálií je zapotřebí dostatečný výkon Hw na kterém jsou uložena data a probíhají výpočty. Z toho důvodu je navrženo řešení, které nahrazuje klasickou koncepcí pro detekci anomálií jak je zobrazena na následujícím obrázku.



Obrázek 11 Klasická koncepce pro sběr dat a vyhodnocení

Navržené řešení rozděluje činnost do tří samostatných částí. První část je modul pro sběr dat. Druhou je server, kde jsou uchovávána naměřená data a probíhají výpočty detekce anomálií. A třetí část, kterou je mobilní telefon sloužící pro přenos dat z měřicího modulu na server a pro vizualizaci výsledků.

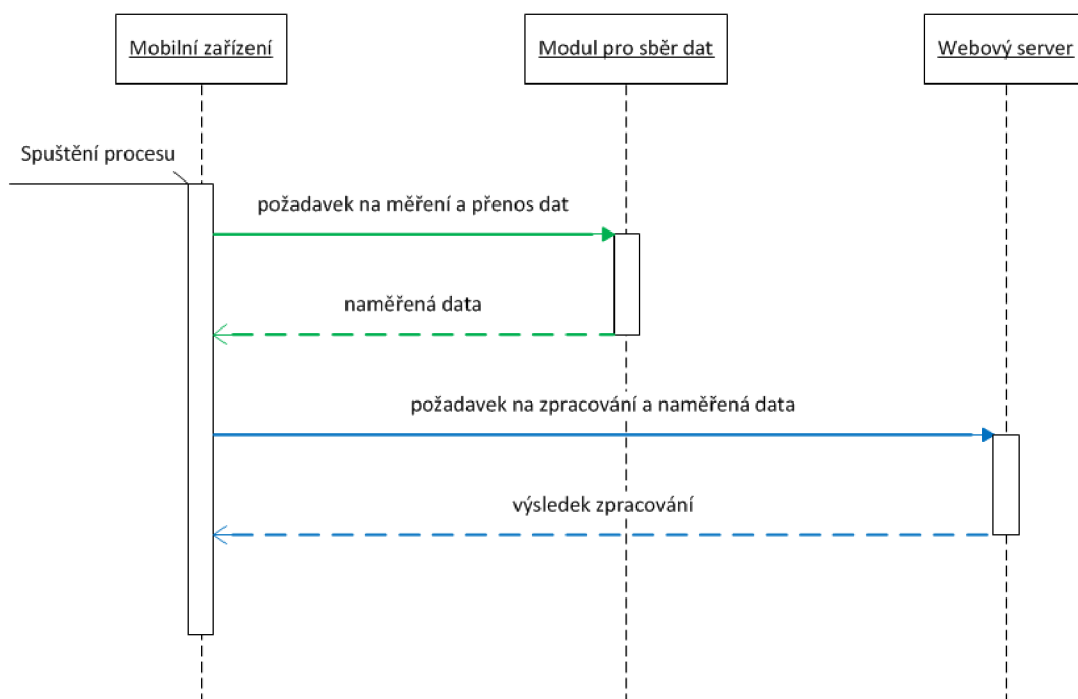


Obrázek 12 Navržená koncepce pro sběr dat a vyhodnocení

Výhodou navrženého řešení je zajištění dostatečné kapacity a výkonu HW pro zpracování. Vzhledem k tomu, že se jedná o webovou aplikaci, je tato globálně dostupná prostřednictvím internetu. Je tedy možné na jednom místě uchovávat naměřená data z více jednotek pro sběr dat a vyhodnocovat i např. opotřebení u

mechanických komponent za dlouhé časové období. Tato data a výsledky zpracování se mohou rovněž integrovat do systémů pro plánování a vyhodnocování údržby. Mobilní telefon v navrženém řešení musí umět komunikovat rozhraním Bluetooth a mít datové připojení k internetu. V dnešní době se jedná o zcela standardní smartphone.

Celý proces od sběru dat až po zobrazení výsledku je možné zobrazit ve zjednodušené formě jako sekvenční diagram. Zprávy označené zelenou barvou jsou posílány prostřednictvím Bluetooth a modré zprávy jsou RESTová volání v síti internet.



Obrázek 13 Sekvenční digram návrhu řešení

4 LABORATORNÍ PŘÍPRAVEK

4.1 Požadavky

Laboratorní přípravek navržený v rámci řešení diplomové práce by měl odolat i případnému nešetrnému zacházení během laboratorních cvičení a měl by být bezpečný. Z toho důvodu nebyly použity žádné točivé komponenty, kde by byla možnost zachycení oděvu nebo vlasů a v případě velkých otáček možnost odlétnutí části vlivem odstředivé síly.

4.2 Technický návrh

Jako nejjednodušší přípravek jsem navrhnul kmitající nosník. K tomuto nosníku je možné namontovat držáky, kam lze umístit kuličky z různých materiálů a různých velikostí. Během kmitání nosníku budou tyto kuličky narážet do stěn a nosníku a způsobovat vibrace.

Konstrukce je tedy jednoduchá, a navíc jí bylo docíleno s nízkými výrobními náklady.



Obrázek 14 Zrealizovaný přípravek pro sběr dat

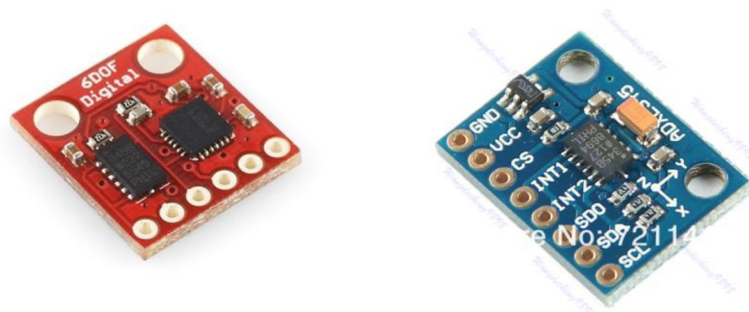
4.3 Modul pro sběr dat

Pro měření vibrací jsem použil měřící modul osazený akcelerátorem mems ADXL345. Jeho cena se pohybuje v nízkých jednotkách dolarů. Akcelerátor komunikuje po sběrnici I²C.

Parametry ADXL345 [17]:

- Ultranízká spotřeba: méně než 40 μA v měřicím módu a 0.1 μA v standby módu při napájecím napětí $V_S = 2.5 \text{ V}$
- Spotřeba proudu se přizpůsobuje automaticky podle přenášeného pásma
- Uživatelem volitelná rozlišení
- Pevné 10 bitové rozlišení

- Plné rozlišení, kde se zvyšuje rozlišení s g rozsahem, až 13-bitovým rozlišením na ± 16 g (zachování 4 mg / LSB rozlišení ve všech rozsazích g)
- Vestavěná, patentovaná FIFO technologie minimalizuje zátěž hlavního procesoru
- Detekce jedno nebo dvoj tapnutí
- Monitoring aktivity/nečinnosti



Obrázek 15 Modul s akcelerometrem ADXL345 [20][35]

K vlastnímu zpracování dat z akcelerátoru jsem vybíral z těchto HW platform:

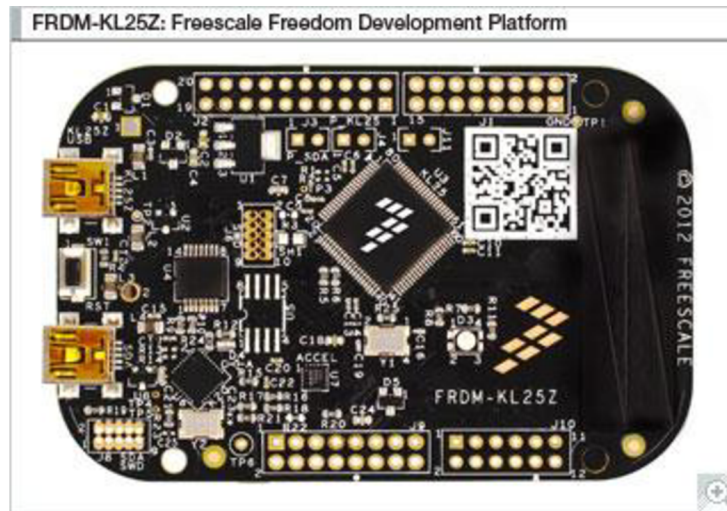
- Arduino je otevřená platforma určená pro jednoduché a rychlé použití. Její nevýhodou je cena, a vlastní programovací jazyk.
- Raspberry Pi. Jedná se o plnohodnotný počítač s operačním systémem na bázi Linuxu. Nevýhodou je, v případě použití některého ze standardních jazyků (C, Java), není zaručený dostatečně rychlý přístup k GPIO konektoru pro připojení periférií.
- Vývojový modul FRDM-KL25Z, který je osazen ARM® procesorem. Jeho cena se pohybuje okolo 50% v oproti předchozím platformám. SW je možno psát v jazyce C/C++.

Z uvedených platform jsem vybral modul FRDM-KL25Z z důvodu ceny a možnosti programování v jazyce C/C++. Jazyk C/C++ je součástí výuky a je tedy možné, aby studenti prováděli v případě potřeby úpravy přímo v měřicím modulu jako např. změnu vzorkovací frekvence nebo definovali množství načtených dat. Další výhodou je fakt, že má stejné rozmístění vývodů jako Arduino, a je tedy možné k němu připojovat moduly, které jsou vyrobeny přímo pro tuto platformu.

Parametry FRDM-KL25Z [18] :

- Freescale KL25Z Kinetis KL2x MCU (MKL25Z128VLK4)
- Vysoce výkonné ARM® Cortex™-M0+ jádro
- 48MHz, 16KB RAM, 128KB FLASH
- USB (Host/Device)
- SPI (2)

- I2C (2)
- UART (3)
- PWM (TPM)
- ADC (16 bit)
- DAC (1x 12bit)
- Touch Sensor
- GPIO (66)



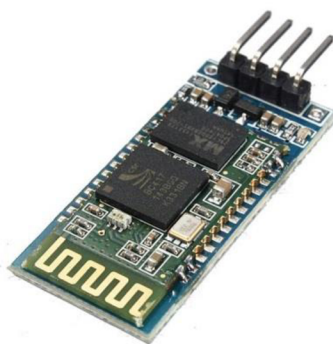
Obrázek 16 Freescale KL25Z Kinetis KL2x MCU [21]

Pro přenos dat do mobilního telefonu bude použita bezdrátová technologie Bluetooth. Touto technologií je vybavena většina chytrých mobilních telefonů (pro něž se běžně užívá pojem smartphone).

Z dostupných Bluetooth zařízení jsem použil modul HC-06, který bude komunikovat rozhraním RS232 s modulem FRDM-KL25Z.

Parametry HC-06 [19]:

- Bluetooth verze: V2.0+EDR
- Pracovní napětí: 3.3V
- Napájecí napětí 3.6 - 6V
- Defaultní Baud Rate: 9600,8,1,n.
- Maximální Baud Rate: 115200,8,1,n.
- Integrovaná anténa, dosah až 9m.

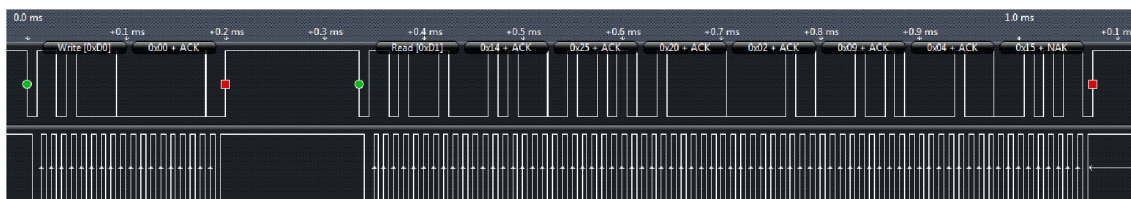


Obrázek 17 Bluetooth modul HC-06 [19]

4.3.1 Použité technologie

4.3.1.1 I²C [34]

Jedná se o sériovou sběrnici někdy označovanou jako IIC (Internal-Integrated-Circuit Bus). Slouží k připojování nízkorychlostních (100kHz nebo 400kHz) obvodů k procesoru. Pro adresování slouží standardně 7 bitů, takže lze připojit 128 zařízení. Adresa je dána výrobcem zařízení. Komunikace probíhá po dvou vodičích SDA (data) a SCL (clock - hodiny).



Obrázek 18 Ukázka skutečné komunikace pro I²C sběrnici

4.3.1.2 Bluetooth [22]

Jedná se o otevřený standard bezdrátové komunikace. Patří do kategorie osobních počítačových sítí. V dnešní době je používán v souvislosti s bezdrátovým přenosem zvuku např. v automobilech při telefonování nebo poslouchání hudby. Zařízení je potřeba spárovat výměnou definovaného hesla. Bluetooth může komunikovat různými protokoly např.: LMP, L2CAP, HCI, RFCOMM.

4.3.2 Software

Jako vývojové prostředí byl použit on-line nástroj „ARM mbed Developer Site“ dostupný na adrese <http://developer.mbed.org/>. Toto vývojové prostředí obsahuje i knihovny určené pro komunikaci s různými perifériemi. V našem případě i knihovnu pro I²C komunikaci s akcelerometrem ADXL345.

Aplikace obsahuje smyčku, kdy očekává zaslání spouštěcí sekvence po rozhraní RS-232 prostřednictvím adaptéru Bluetooth. V případě jejího obdržení spouští indikaci, která

signalizuje přijetí požadavku a následuje signalizace, že bude spuštěn sběr dat ze snímače. Intervaly indikací jsou nastaveny na 1 sekundu. V cyklu je ze snímače načteno 1024 hodnot. Toto množství je určeno na základě dostupné paměti mikrokontroleru a požadavku pro výpočet FFT, kdy je potřeba aby číslo bylo mocninou čísla 2.

Rychlost měření je 800 měření za sekundu. Její omezení je rychlost použité sběrnice I²C a množství přenášených dat. Naměřené hodnoty jsou uloženy v paměti RAM. Po načtení všech vzorků hodnot (1024 x 3 hodnoty z akcelerometru), jsou tato postupně zasílána po RS-232 do mobilní aplikace.

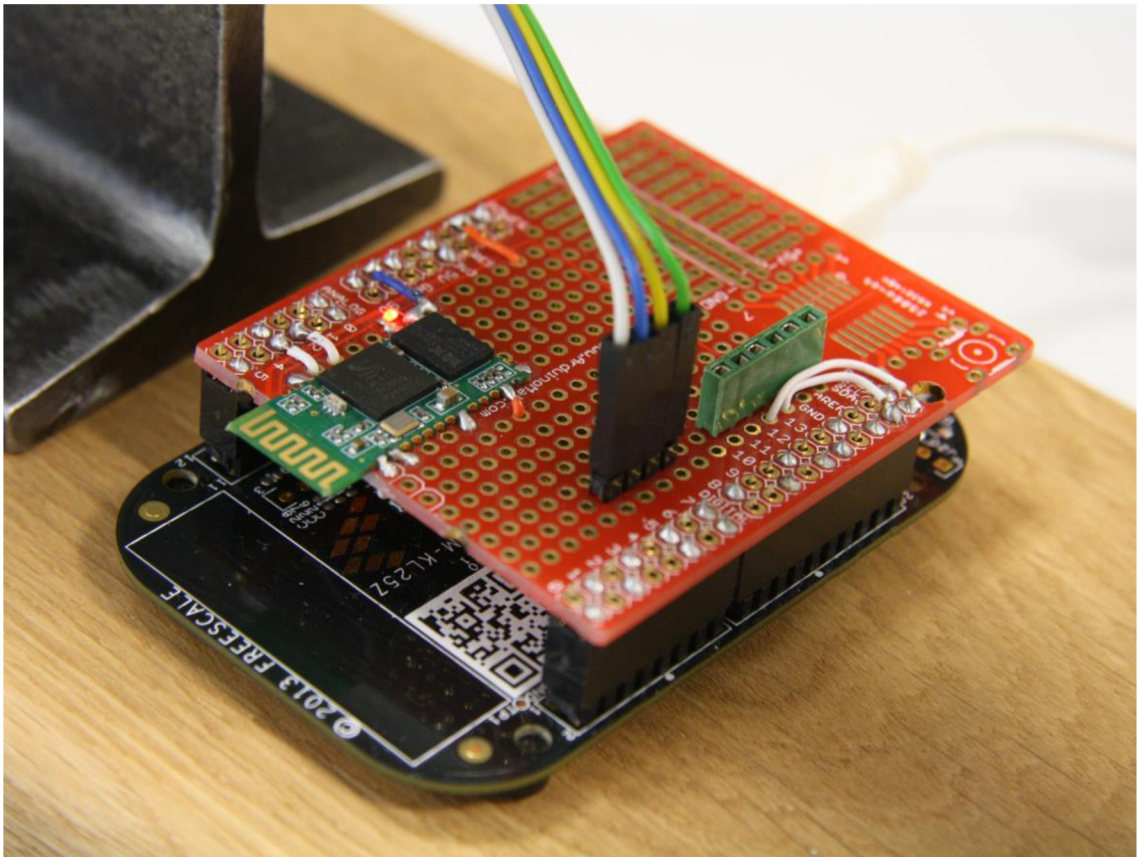
SW je možné upravit například i pro případy, kdy by bylo potřeba měřit hodnoty v dlouhém časovém období. Modul by v pravidelných časových intervalech měřil a ukládal hodnoty do paměti RAM. Staré hodnoty by přemazával v nekonečné smyčce. Po přijetí požadavku by zaslal poslední sadu naměřených hodnot do okamžiku přijetí požadavku.



Obrázek 19 Diagram smyčky modulu pro sběr dat

4.4 Realizace

Měřicí přípravek je zrealizován na univerzální desce určené pro Arduino kompatibilní zařízení. Na této desce je připevněn komunikační modul pro Bluetooth komunikaci a konektor pro připojení akcelerometru. Přívod k akcelerometru, který je upevněn na konci nosníku, je provede ohebným přívodem. Jako indikátor stavu je použita tříbarevná led dioda umístěná přímo na vývojovém modulu.



Obrázek 20 Měřící modul

5 SERVEROVÁ ČÁST

Serverová část je zrealizována jako webový server komunikující s mobilní aplikací ve formátu JSON RESTovým rozhraním. Vlastní výpočet je prováděn přímo v aplikaci. V případě potřeby lze na server nainstalovat SW třetí strany např. Rapid Miner a algoritmy pro výpočet použít právě voláním této aplikace se zachováním stejného rozhraní směrem k mobilnímu klientovi.

5.1 Požadavky

Část aplikace umístěná na webovém serveru má za úkol shromažďovat všechna naměřená data, provádět výpočtu hodnot příznaků a následně vyhodnocovat naměřená data vhodnými algoritmy strojového učení. Výstupem je výsledek klasifikace. Vstupní a výstupní data jsou ve formátu, který minimalizuje množství přenášených dat.

5.2 Návrh

Pro ukládání naměřených dat lze použít ukládání naměřených dat do souborového systému serveru, nebo použít relační databázi. Aby bylo možné k naměřeným datům ukládat další informace jako např. vypočtené příznaky a pro rychlejší práci s hledáním je výhodnější použití relační databáze.

Dostupné nejčastěji používané relační databáze jsou:

- MS SQL
- Oracle
- MySQL

Předpokládané množství uložených dat je pro 1 měření cca 20kB, kdy počet měření je řádově ve stovkách až tisících. Jedná se tedy o nízké desítky MB uložených dat. Prostředí, kde bude serverová aplikace nasazená, obsahuje všechny tři uvedené typy databází. Pro snadnou přenositelnost na běžný komerční webhosting jsem vybral databázi MySQL. Tato databáze je používaná poskytovateli zejména díky ceně za tuto databázi.

Vlastní aplikaci pro zpracování a vyhodnocení měření lze vytvořit v různých programovacích jazycích. Jelikož se jedná o aplikaci webovou, i když nebude mít žádné klasické webové formuláře, komunikující prostřednictvím webového rozhraní lze ji vytvořit v mnoha programovacích jazycích.

Mezi nejpoužívanější podle [25] patří:

- C#
- PHP
- Java

Jazyk C# běžící na frameworku .NET je vhodný a zpravidla se tak i používá ve spojení s databází MS SQL a vůbec technologií Microsoftu. Běží na aplikačním serveru Internet Information Services. Nevýhodou tohoto řešení je vyšší cena webhostingu v případě veřejného nasazení a závislost na MS platformě.

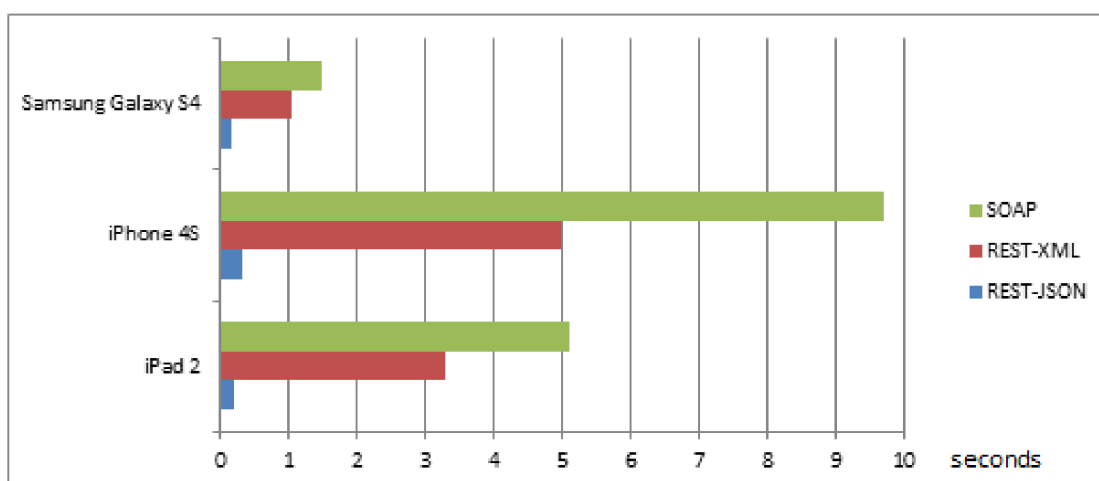
PHP je skriptovací jazyk[24]pro tvorbu webových aplikací, který lze použít i pro objektově orientované programování. Nevýhodou je nutnost integrování frameworků třetích stran v případě potřeby jednoduchého použití např. komunikace různými webovými protokoly.

Java [26] je od počátku jejího vývoje objektově orientovaný jazyk, který není závislý na platformě operačního systému. Pro jeho výběr hovoří i to, že ve stejném jazyce je vytvořena mobilní aplikace, která je součástí celkového řešení.

Aplikační server pro běh aplikace je od poskytovatele Javy doporučován AS Glassfish [28]. Tento AS je poskytován i jako Open Source [29]. Aplikační server je nainstalován s výchozím nastavením. Pro přístup do MySQL databáze je instalována knihovna pro tuto databázi.

Komunikace mezi mobilní a serverovou aplikací je možné řešit vícero způsoby. Jedná se např. o komunikace webovými službami protokolem SOAP nebo protokolem JSON v architektuře REST [31].

Protokol SOAP tak jak je definován jako standard [30] obsahuje ve zprávě velké množství znaků, které slouží k definování struktury, ale nenesou vlastní informaci. Díky tomu je i přenášená zpráva, která nese minimum informací v poměru k těmto informacím (podle typu) i 10 násobně větší než vlastní přenášené informace. Oproti tomu je protokol JSON navržen jako úspornější. Porovnáním protokolů se věnuje článek z února 2015 [40].



Obrázek 21 Porovnání odezvy podle komunikačního protokolu [40]

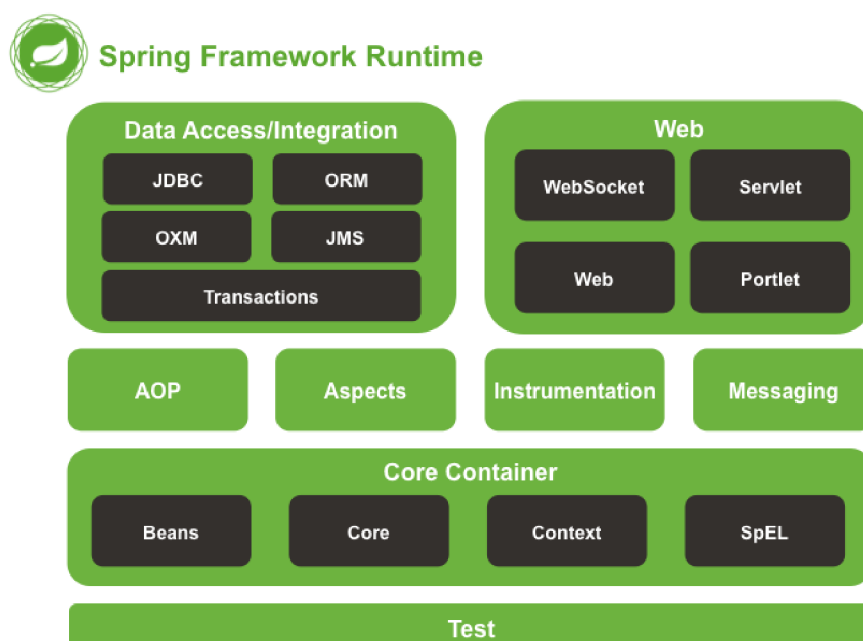
V architektuře REST je hlavní použití http metod jako GET, POST, PUT a DELETE.

Pro naše použití bude vhodná hlavně metoda POST, které slouží ke vložení dat na server.

Pro zpracování naměřených dat do frekvenčního spektra je použita knihovna [32] Kolumbijské univerzity, která v porovnání s knihovnou [33] z Princetonské univerzity má menší požadavky na paměť a je rychlejší.

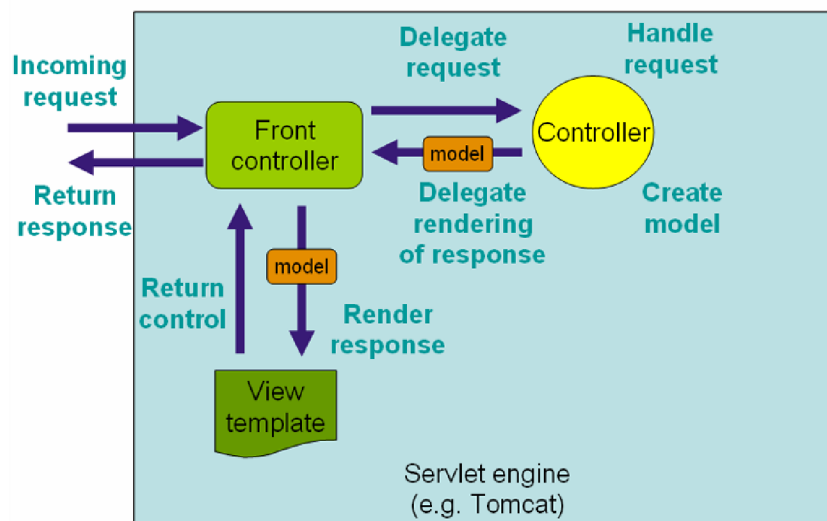
Struktura aplikace – použití frameworků

Serverová aplikace využívá pro svůj běh prostředí frameworku Spring verze 3.2.11 [41]. Seznam oblastí, které Spring řeší je uveden na následujícím obrázku.



Obrázek 22 Struktura frameworku Spring [43]

Pro účely aplikace se využívají různé části např.: Beans, Core, Context a Web. Pro prezentační vrstvu se využívá návrhový vzor MVC (Model View Controller), který zprostředkovává komunikaci mezi kontrolery (obsahují vlastní programový kód) a prezentační vrstvou (JSP stránky). Oproti nižším verzím Springu zde již lze využít vlastnost, kdy namísto konfigurace Front controlleru se používají anotace u jednotlivých metod.



Obrázek 23 MVC

Pro komunikaci s databází je použit perzistentní framework Ibatis verze 2.3.4. Tento framework [42] umožňuje psát v případě potřeby vlastní SQL dotazy na rozdíl např. od frameworku Hibernate. Optimalizace dotazů při jejich vykonávání v případě dolování dat je nutná pro zvýšení početního výkonu a zkrácení odezvy.

Vícevrstvá architektura

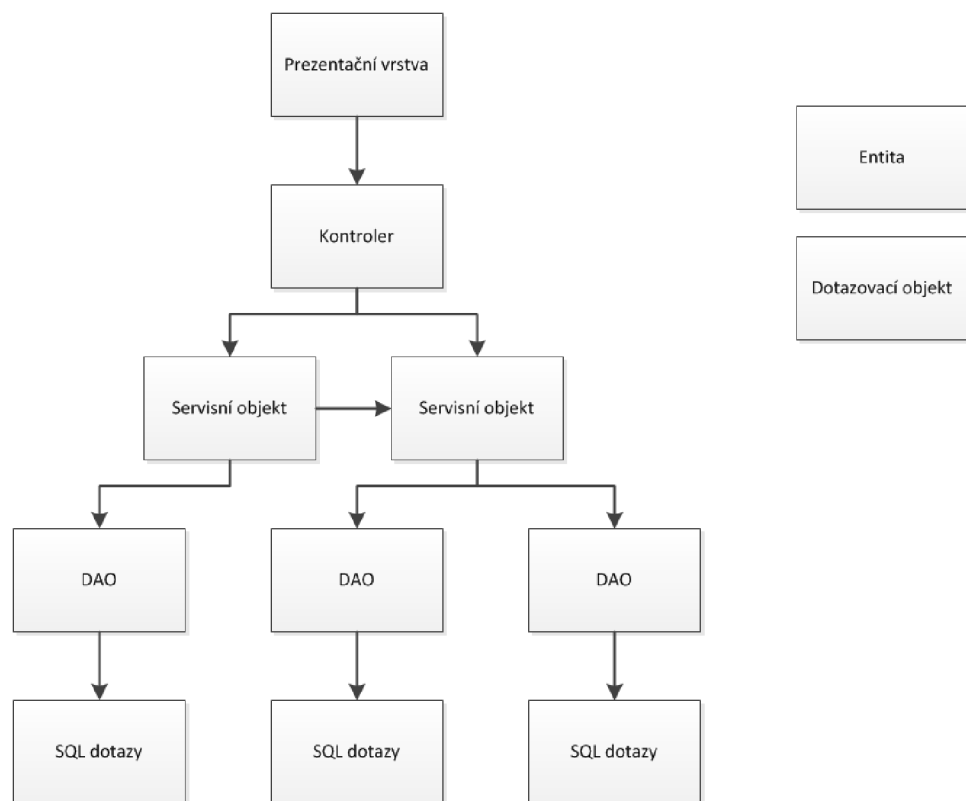
Aplikace je rozdělena do více vrstev z důvodu lepší orientace v projektu a snadné rozšiřitelnosti.

Jedná se o následující vrstvy (směrem odspodu):

- Vrstva SQL dotazů – obsahuje základní sadu SQL příkazů pro práci s tabulkami. Zpravidla to jsou příkazy nad jednou tabulkou: insert, select, update a delete. V případě potřeby je optimální právě do této vrstvy zrealizovat složitější dotazy.
- Vrstva DAO. Tato vrstva zastřešuje předchozí vrstvu do objektu s definovanými metodami. Tyto metody jsou zpravidla v základní množině stejné, proto se zde hodně využívá interfejsů. Interfejsy definují základní operace nad těmi DAO, které je implementují.
- Servisní vrstva. Obsahuje hlavní řídicí kód aplikace. Jedná se zpravidla o komplexní metody, která pro svou potřebu využívají DAO vrstvy, případně jiné servisní třídy. Logika implementování metod do této vrstvy spočívá v tom, že je lze volat z libovolného místa aplikace, kde je jich potřeba.
- Kontrolery. Jedná se o část návrhového vzoru MVC, která zajišťuje interakci mezi prezentační vrstvou a servisní vrstvou. Jejím úkolem je reagovat na podněty od klienta (zpravidla webový prohlížeč nebo klient webové služby), vyhodnocovat správnost dotazů a předávat řízení do servisní vrstvy. Výsledky potom jsou ve formě modelu vráceny do prezentační vrstvy.
- Prezentační vrstva. Obsahuje kód, který se zasílá na klienta. Jedná se zpravidla o HTML kód doplněný o části psané v jazyce Java. Tyto části se během

zpracování na serveru na aplikačním serveru doplní o hodnoty zaslané modelem z kontroleru. Tato vrstvy v případě webových stránek bývá doplněna zpravidla o Javascript, který řeší reakce na události přímo v prohlížeči na klientovi.

- Mimo vrstvy se ještě vyskytují objekty typu Entita (jejím obrazem je logická entita dat – zpravidla jeden řádek z tabulky) a dotazovací objekt, který slouží k předávání parametrů od prezentační vrstvy až do vrstvy DAO. Výsledkem zpracování dotazu na základě dotazovacího objektu je zpravidla právě entita, nebo pole entit (podle typu dotazu).



Obrázek 24 Schéma rozložení aplikace do vrstev

Servisní vrstva bude v realizované aplikaci zajišťovat výpočet atributů, uložení dat (ve spolupráci s DAO), načtení všech dat daného pokusu (DAO), normalizaci a vlastní výpočet.

Realizují se vybrané algoritmy a to k-NN pro klasifikaci do tříd a LOF pro výpočet míry odlehlosti.

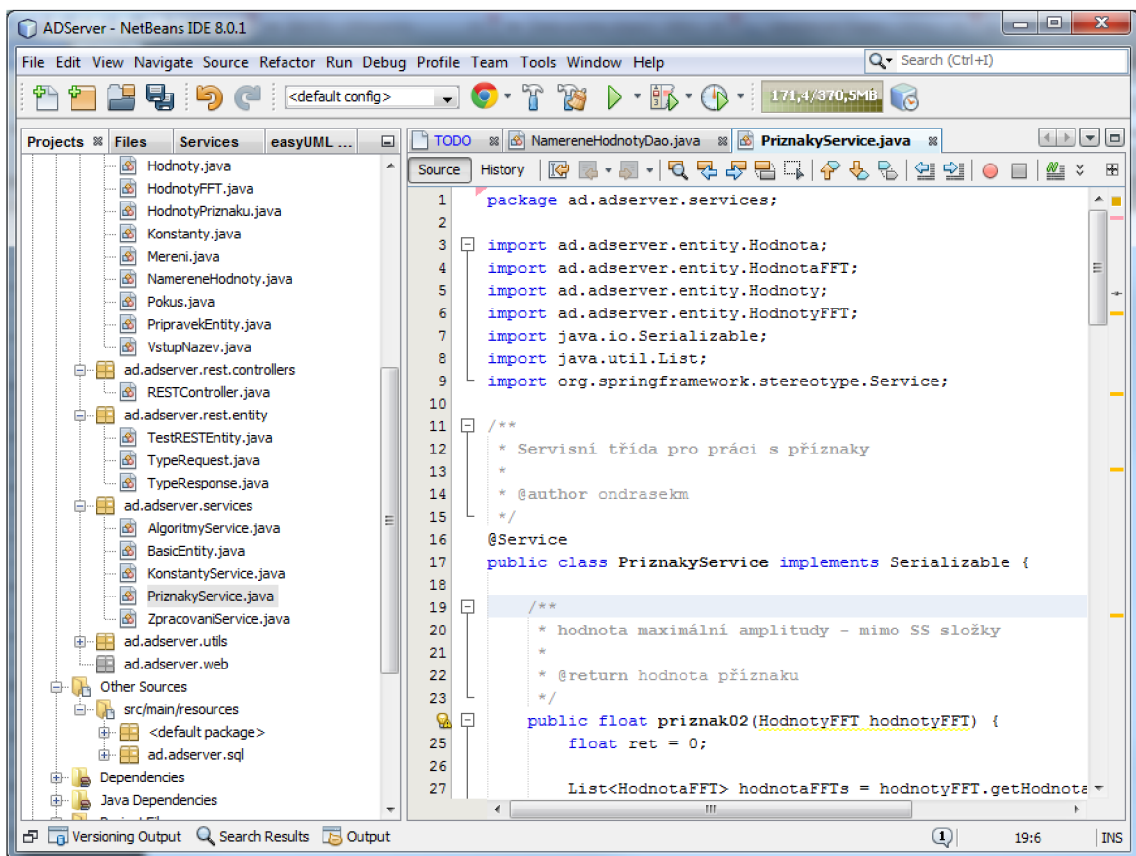
5.2.1 Použité technologie a nástroje

5.2.1.1 Java

Serverová aplikace je ve verzi Java EE 7 WEB. Pro vývoj je použita platforma JDK 1.7, kdy projekt je kompilován i pro běh v prostředí verze 1.6. Vše dostupné na adrese <https://www.oracle.com/java/index.html> .

5.2.1.2 Netbeans IDE

Vývojové prostředí pro vývoj v různých jazycích. Dostupné na adrese: <https://netbeans.org/> . Toto vývojové prostředí je rozšiřitelné o komponenty třetích stran, kterými lze upravit funkčnost tohoto IDE. Pro realizaci byla použita verze 8.0.1.



Obrázek 25 Ukázka prostředí Netbeans IDE

5.2.1.3 Aplikační server Glassfish

Je držitelem označení „World's first Java EE 7 Application Server“ [28]. Pro běh použita verze 4.0 dostupná z <https://glassfish.java.net/download.html> .

5.2.1.4 Spring

Framework je popsán v kapitole 5.2. Dostupný je na adrese: <http://spring.io/> .

5.2.1.5 Ibatis 2.3.4 [42]

Framework je popsán v kapitole 5.2. Dostupný je na adrese <http://blog.mybatis.org/> .

5.2.1.6 MySql

Relační databáze [46], která je dostupná ve free verzi a komerční verzi. Pro účel práce byla použita free verze dostupná na adrese: <http://www.mysql.com/> . Lze ji provozovat na různých operačních systémech.

5.2.1.7 REST API

Použití popsáno v kapitole 5.2.

5.2.1.8 JSON

Formát pro výměnu dat mezi systémy. Použití popsáno v kapitole 5.2.

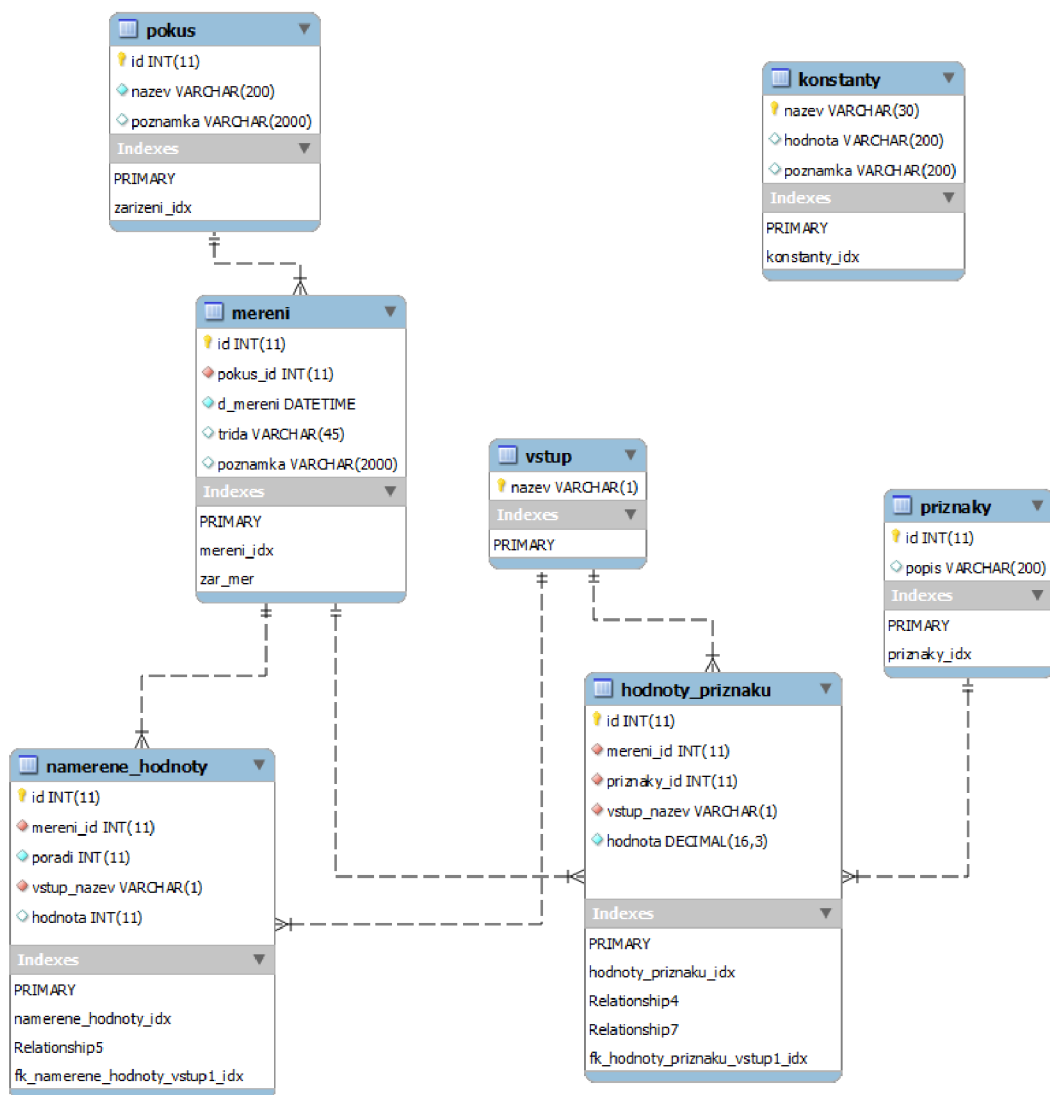
5.2.2 Datový model

Datový model je navržen v nástroji MySQL Workbench ve verzi 6.0.7, který je dostupný zdarma na adrese <https://www.mysql.com/products/workbench/> . Teno nástroj umožňuje jak databáze navrhovat, tak je i spravovat a administrovat. Z takto navrženého modelu se vygenerováním skriptu vytvoří potřebná struktura přímo v určeném databázovém schématu (v našem případě „addb“). Struktury vygenerované z datového modelu obsahují všechna naměřená primární data a vypočítané hodnoty příznaků. Uchování příznaků v databázi namísto on-line výpočtů za běhu má výhodu zrychlení algoritmů v případě zpracování velkého množství měření. V případě potřeby je možné dopočítat nové příznaky, nebo přepočítat již jednou vytvořené. Směr vazeb 1:N mezi entitami je zobrazen graficky.

DM obsahuje tyto entity (tabulky):

- *pokus* – obsahuje informace k jednomu prováděnému pokusu ve smyslu sady měření na jednom místě. Tímto místem se myslí měření např. jednoho studenta, nebo měření na různých místech.
- *mereni* – obsahuje informace konkrétního jednoho měření v rámci vybraného pokusu. Obsahuje datum a čas měření, název třídy u klasifikace pokud je znám a poznámku k měření.
- *namerene_hodnoty* – obsahují jednotlivé naměřené hodnoty. U každé hodnoty je určeno pořadí a vazba na entitu *vstup* podle konkrétní osy měření (data z akcelerometru jsou ve třech osách). V případě použití jiného typu snímače, nebo sady snímačů zde může být identifikace konkrétního snímače.

- *hodnoty_priznaku* – ke každému měření jsou zde uloženy všechny vypočtené příznaky. V případě měření ve více osách, nebo z více snímačů je zde opět vazba na entitu *vstup*
- *vstup* – je číselníková tabulka, která obsahuje výčet řad naměřených hodnot (osy, identifikátory snímačů)
- *priznaky* – číselníková tabulka obsahující seznam typů příznaků, které jsou v tabulce *hodnoty_priznaku* uloženy.
- *konstanty* – obsahují nastavení aplikace. Jedná se např. o parametry pro výpočet příznaků, nebo globální nastavení aplikace.

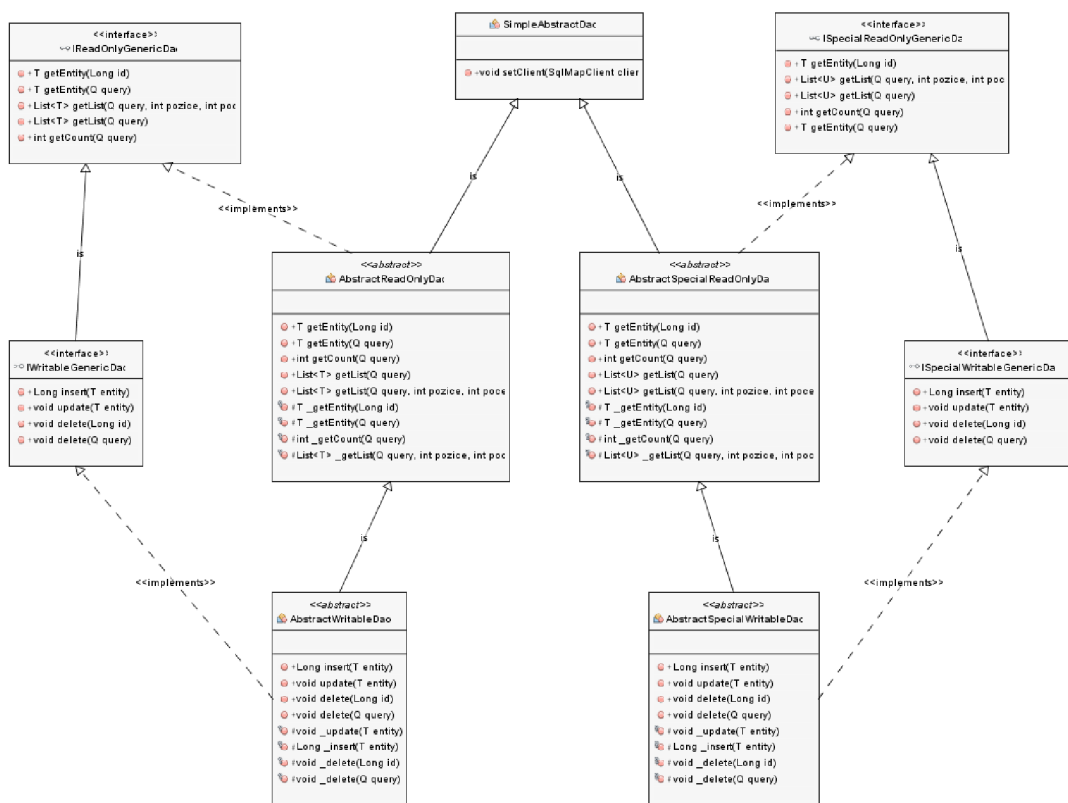


Obrázek 26 Návrh datového modelu

5.2.3 Objektový návrh

Objektový návrh [16] by měl být prováděn u každé aplikace. Často je v praxi zaměňován za jednodušší návrh vrstev ve vícevrstvé architektuře. Oproti návrhu architektury jde ale objektový návrh do hloubky a řeší strukturu aplikace z pohledu koncepce aplikace [47]. Snaží se kopírovat reálné úlohy do objektů, tak jak je známe. Velká množina řešení opakujících se programovacích úloh je popsána návrhovými vzory [13]. Část frameworku Spring WEB je postavena právě na jednom z návrhových vzorů MVC.

Objektový model byl realizován před vlastním vývojem a v průběhu realizace byl upřesňován.



Obrázek 27 Ukázka objektového modelu pro část DAO vrstvy

Ukázka třídy AbstractWritableDao:

```
package ad.adserver.dao.generic;
```

```
/**
 * Spolecny predek pro DAO. Krome cteni umi i zapisovat, updatovat, mazat.
 * Potomci implementuji metody s prefixem "_", ktere resi samotny
 * dotaz do DB - volani odpovidajiciho mapovani iBatisu
 * @author ondrasekm
 */
```

```

public abstract class AbstractWritableDao<T, Q> extends AbstractReadOnlyDao<T,
Q> implements IWritableGenericDao<T, Q> {

    /**
     * {@inheritDoc }
     */
    @Override
    public Long insert(T entity) {

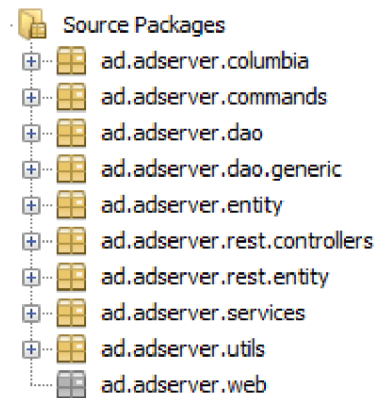
        try {
            return _insert(entity);
        }
        catch(RuntimeException ex) {
            throw ex;
        }
    }
}

```

Objektový model celé serverové aplikace obsahující více jak 40 tříd je uveden vzhledem ke své velikosti v samostatné příloze.

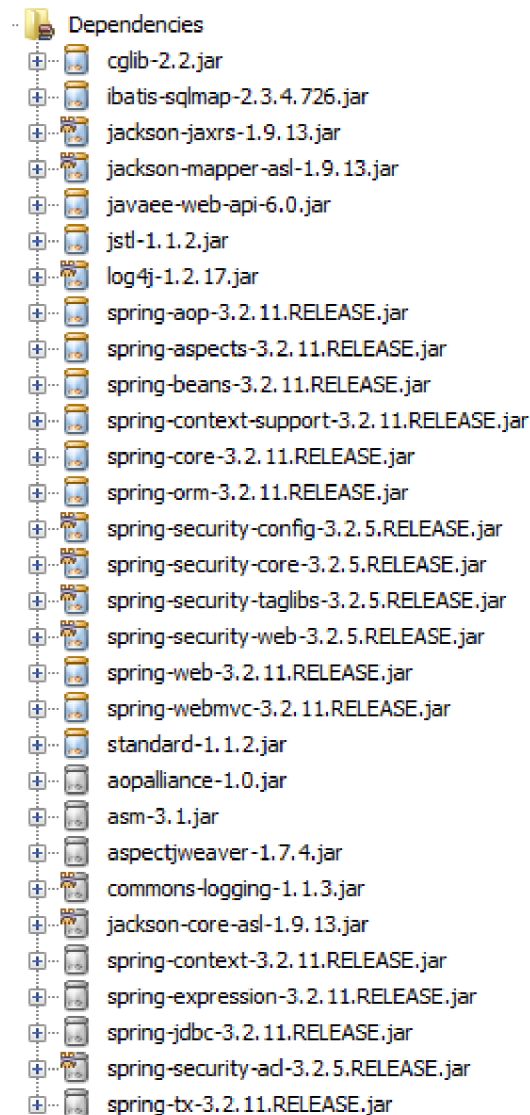
5.3 Realizace

Jednotlivé třídy aplikace jsou uloženy v balíčcích podle vrstvy a použití.



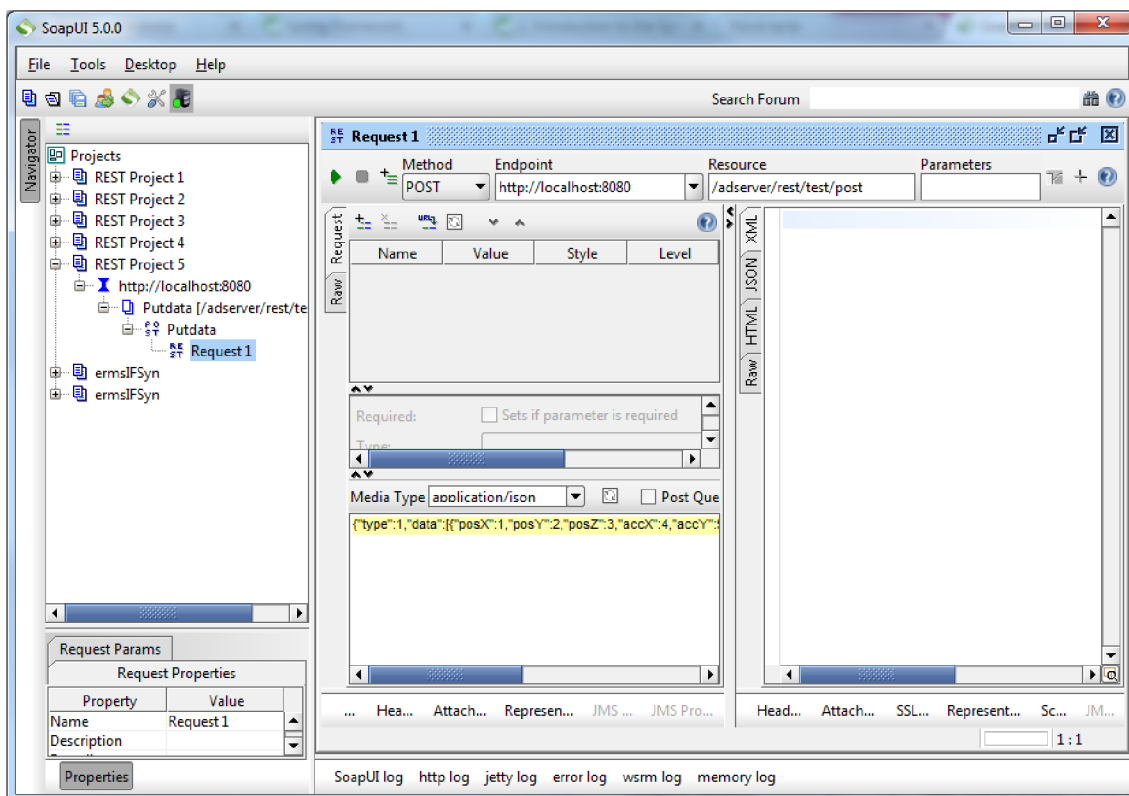
Obrázek 28 Struktura tříd serverové aplikace

Pro vlastní běh aplikace bylo nutno integrovat do projektu i knihovny třetích stran. Jejich výčet je zobrazen na následujícím obrázku. Využití každé konkrétní knihovny lze dohledat podle názvu třídy buď v IDE Netbeans nebo na internetu.



Obrázek 29 Knihovny třetích stran integrované do projektu

Během vývoje aplikace bylo potřeba testovat serverovou část. Pro potřeby testovacího volání byl použit nástroj SoapUI ve verzi 5.0.0 dostupný na adrese <http://www.soapui.org/> . Tento nástroj umožňuje simulovat volání různých metod různými formáty (SOAP, JSON, ..). Na obrázku je ukázka volání metody POST , kde se zasílá objekt ve formátu JSON. Jedná se o simulaci volání z mobilní aplikace a předávají se naměřená data.



Obrázek 30 Aplikace SoapUI pro testování komunikace

Pro celou aplikaci je na základě vkládaných komentářů a popisů vygenerovaná dokumentace utilitou Javadoc. Tato dokumentace má stejnou strukturu jako aplikace a je vygenerovaná do formátu HTML. Vložené komentáře a popisy využívá i přímo IDE pro snadnější orientaci vývojáře během práce s metodami při programování. Vzhledem k velkému rozsahu je umístěna v příloze.

The screenshot displays the Javadoc documentation for the class `ad.adserver.dao.generic.AbstractSpecialReadOnlyDao<T,U,Q>`. The interface is organized as follows:

- Navigation:** Overview, Package, **Class**, Use, Tree, Deprecated, Index, Help.
- Class Path:** ad.adserver.dao.generic
- Class Name:** **Class AbstractSpecialReadOnlyDao<T,U,Q>**
- Imports:**
 - java.lang.Object
 - org.springframework.dao.support.DaoSupport
 - org.springframework.orm.ibatis.support.SqlMapClientDaoSupport
 - ad.adserver.dao.generic.SimpleAbstractDao
 - ad.adserver.dao.generic.AbstractSpecialReadOnlyDao<T,U,Q>
- All Implemented Interfaces:**
 - ISpecialReadOnlyGenericDao<T,U,Q>, org.springframework.beans.factory.InitializingBean
- Direct Known Subclasses:**
 - AbstractSpecialWritableDao
- Source Code:**

```
public abstract class AbstractSpecialReadOnlyDao<T,U,Q>
extends SimpleAbstractDao
implements ISpecialReadOnlyGenericDao<T,U,Q>
```
- Description:** Spolecný predek pro read-only DAO. Resí obecné logování dle návrhového vzoru template-method. Potomci implementují metody s prefixem "_", které resí samotný dotaz do DB - volání odpovídajícího mapování iBatisu. T - entita s detaily, U - entita osekána o nepotřebné properties (treba pro použití v RecordListu), Q - dotazovací objekt.
- Author:** ondrasekm

Obrázek 31 Ukázka vygenerované dokumentace Javadoc k aplikaci

6 MOBILNÍ APLIKACE

6.1 Požadavky

Požadavkem je jednoduchost aplikace z důvodu snadné přenositelnosti na různé platformy používané pro mobilní zařízení.

6.2 Návrh

Aplikace je zrealizována pro platformu Android [27]. Důvodem je velká rozšířenost zařízení s tímto operačním systémem a fakt, že aplikace lze programovat v jazyce Java stejně jako serverovou část. Jako API pro tuto aplikaci je použito API19. Aplikace vytvořené s touto verzí API lze spouštět na většině dnes dostupných zařízení s tímto operačním systémem.

6.2.1 Použité technologie

6.2.1.1 Java

Pro vývoj je použita platforma JDK 1.7 s použitím Android API19 verze 4.4.2 (KitKat), JDK je dostupné na adrese <https://www.oracle.com/java/index.html> . API19 je možné nainstalovat v SDK manažeru vývojového prostředí Android Studio nebo přímo z adresy: <https://developer.android.com/sdk/index.html> .

6.2.1.2 BT – RS 232

Pro komunikaci rozhraním Bluetooth se využívá metod knihovny API. Pro použití je potřeba nastavit povolení aplikace k tomuto hardwaru následující konfigurací v souboru AndroidManifest.xml:

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

6.2.1.3 REST API

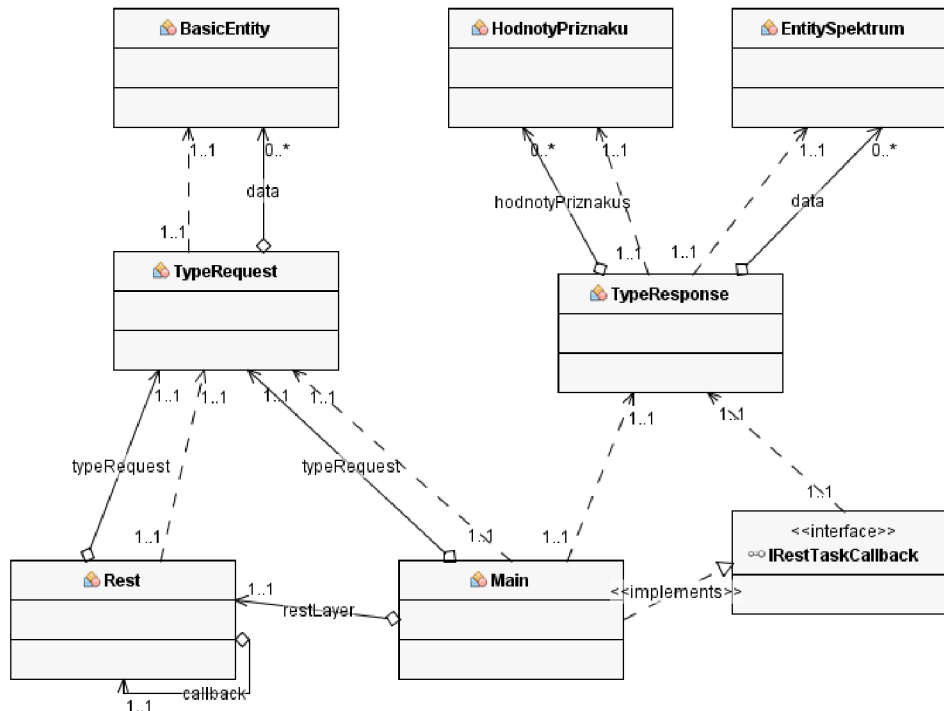
Pro komunikaci je využita knihovna třetí strany gson ve verzi 2.2.4. Knihovna provádí serializaci a deserializaci objektů do řetězce JSON pro komunikaci REST. Knihovna je dostupná na adrese: <https://code.google.com/p/google-gson/> .

6.2.1.4 Zobrazení průběhů a grafů

Zobrazení naměřených hodnot je naprogramované s použitím knihovny GraphView. Jedná se o knihovnu pro platformu Android pro použití v prezentační vrstvě k vytvoření grafických prvků průběhů nebo grafů. Dostupná je na adrese: <http://www.android-graphview.org/> .

6.2.2 Objektový návrh

Vzhledem k požadované jednoduchosti mobilní aplikace je tato rozdělena pouze do osmi tříd, z nich pět obsahuje strukturu komunikačních souborů (stejně soubory jsou i na straně serveru), jedna obsahuje rozhraní a pouze dvě třídy mají vlastní výkonný kód.



Obrázek 32 Objektový model mobilní aplikace

6.2.3 GUI

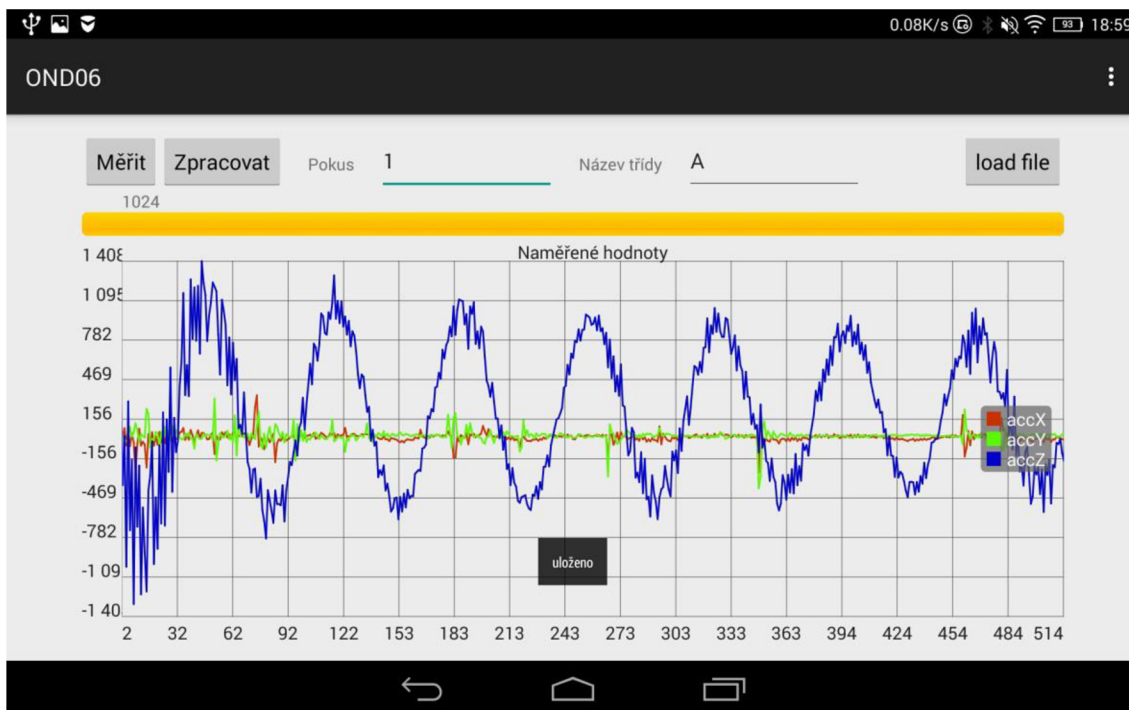
Vzhledem k tomu, že aplikace pro mobilní telefon má být co nejjednodušší, běží celá aplikace pouze s jednou obrazovkou.

Při zapnutí aplikace dojde k zapnutí bluetooth adaptéru (pokud neběží) a je o tom informován uživatel pomocí vyskakovacího dialogového okna ve spodní části obrazovky (na ilustračním obrázku je zde zobrazen text „uloženo“). Stejným způsobem se informuje uživatel v případě potřeby a probíhajících činnostech. Obrazovka je rozdělena na dvě hlavní části. První je část formulářové, která slouží k interakci s uživatelem. Obsahuje tlačítka, editační pole a identifikátor stavu načítání. Druhá část, již je vyhrazena zbylá plocha (pod horizontálním indikátorem stavu načítání) je dynamické pole, které slouží k zobrazení grafických nebo textových informací podle potřeby.

Formulářová část obsahuje vstupní pole pro označení pokusu. Označením „pokus“ se myslí skupina souvisejících měření. Dále vstupní pole pro označení názvu třídy v případě klasifikace do tříd. Dále se zde zobrazují tlačítka pro spuštění vlastního

měření měřicím modulem na přípravku. Po stisknutí tohoto tlačítka dojde k odeslání řídicí sekvence do měřicího modulu. Naměřené hodnoty jsou po měření nahrávány do aplikace a průběh jejich načítání se zobrazuje na horizontálním indikátoru. Počet vzorků je též zobrazen jako číslo pod tlačítkem měřit.

Po načtení všech dat se změřená data vykreslí formou grafu v dynamické (druhé) oblasti. Graf se automaticky nastaví tak, aby zobrazoval všechny naměřené hodnoty na celé ploše dynamické oblasti a obsahoval legendu. V grafu se lze v časové ose pohybovat tažením grafu.



Obrázek 33 Ukázka uživatelského rozhraní

Tlačítko „Zpracovat“ odešle načtená vizualizovaná data ke zpracování na server. Po zpracování na serveru a obdržení odpovědi se v dynamické oblasti zobrazí odpovídající výstup zpracování

Podrobné použití aplikace je popsáno v uživatelské příručce.

6.3 Realizace

Vlastní vývoj mobilní aplikace je v prostředí vývojového nástroje Android Studio IDE verze 0.9 a později jeho novější verze 1.0.1. Pro tento nástroj jsem se rozhodl z důvodu jednoduché instalace vývojového prostředí a snadné konfigurace pro vlastní vývoj oproti jiným nástrojům např. Eclipse. Vývojové prostředí je zdarma a je určeno pouze pro mobilní platformu Android. Je dostupné na adrese

<https://developer.android.com/sdk/index.html>.

Po jeho instalaci je potřeba zvolit verzi Androidu, pro který bude vývoj prováděn. Na základě požadavků je vybraná verze API 19 což je verze Android 4.4.2. Následně je vybrané SDK doinstalováno do Android Studia.

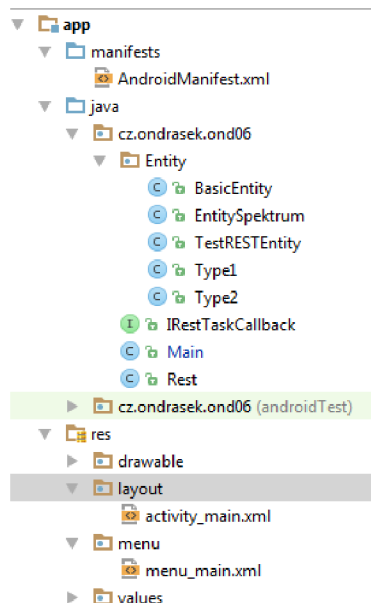
Android 4.4.2 (API 19)				
<input type="checkbox"/>	SDK Platform	19	4	Installed
<input type="checkbox"/>	Samples for SDK	19	6	Installed
<input type="checkbox"/>	ARM EABI v7a System Image	19	2	Installed
<input type="checkbox"/>	Intel x86 Atom System Image	19	2	Installed
<input checked="" type="checkbox"/>	Google APIs (x86 System Image)	19	10	Update available: rev. 12
<input checked="" type="checkbox"/>	Google APIs (ARM System Image)	19	10	Update available: rev. 12
<input type="checkbox"/>	Glass Development Kit Preview	19	11	Not installed
<input type="checkbox"/>	Sources for Android SDK	19	2	Installed

Obrázek 34 Nastavení API Android Studia

Pro vývoj bez vlastního Android zařízení lze nastavit emulátor, na kterém lze aplikaci během vývoje ladit a testovat. Vzhledem k omezení, že emulátor neumí komunikovat pomocí Bluetooth, musel být vývoj a ladění realizováno na externím připojeném zařízení. Jedná se konkrétně o mobilní telefon Samsung Galaxy Young a tablet Lenovo Yoga 2. Aplikace je otestovaná a běží na obou zařízeních. Vzhledem k velikosti displeje jsou veškeré ukázky aplikace snímány na tabletu.

Aplikace se skládá ze dvou hlavních typů souborů. První jsou vlastní .java soubory s aplikační logikou a druhou .xml s konfigurací prezentační vrstvy GUI. Soubory definující komunikační třídy předávané mezi aplikací a serverem jsou použity stejně, jako v serverové části. Pouze jsou umístěny v jiném balíku.

Objektový návrh aplikace je jednoduchý pro snadnou přenositelnost na jiné mobilní platformy.



Obrázek 35 Struktura projektu pro mobilní telefon

Popis aplikace

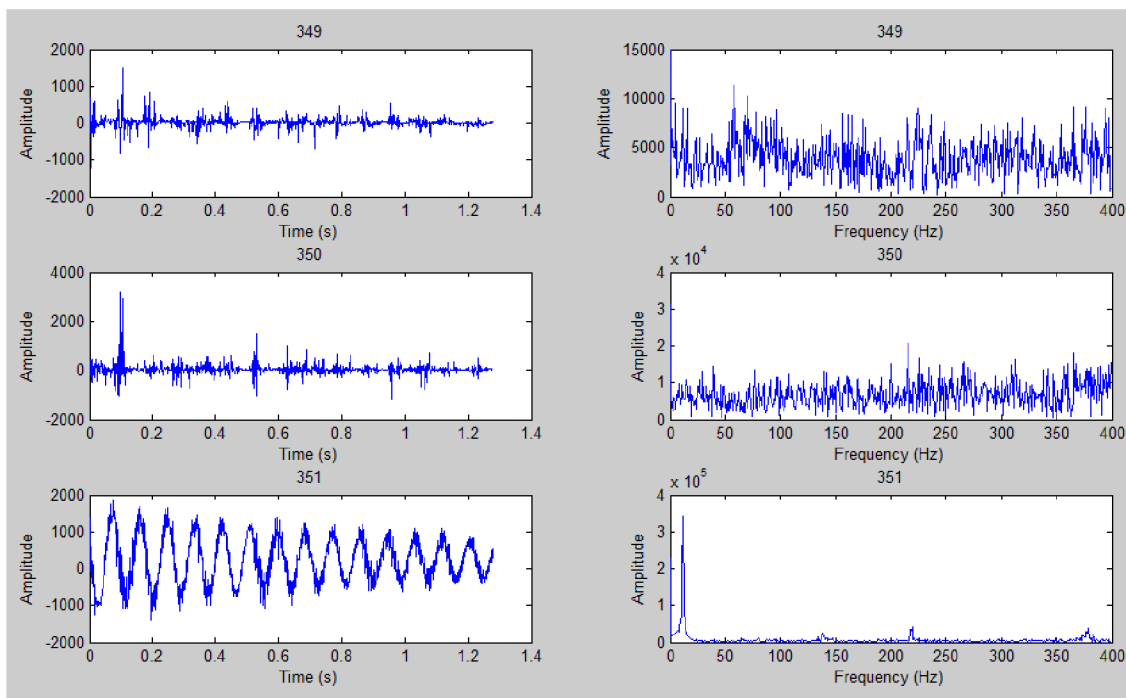
Po spuštění aplikace se kontroluje, zda je spuštěný Bluetooth na zařízení a zda ho zařízení vůbec obsahuje. V opačném případě je Bluetooth spuštěn, nebo informován uživatel a aplikace je ukončena. Spárování mobilního zařízení a měřicího modulu je nutno provést přímo v operačním systému mobilního zařízení. Rychlost komunikace RS-232 rozhraním Bluetooth je nastavena na maximum což je 115200bps. Stisknutím tlačítka „Měřit“ dojde k odeslání dat směrem do měřicího modulu. Systém nani asynchronně čeká na příjem dat. Postu příjmu je indikován na obrazovce uživatele. Přijímaná dat se ukládají do objektu TypeRequest jako pole hodnot definované třídou BasicEntity.

Po obdržení všech dat (množství je nastaveno v aplikaci na 1024 měření) jsou tato získaná data uložena na souborový systém zařízení. Pro snadnou orientaci v uložených souborech je struktura názvu souboru následující: „/sdcard/OND/pokus_třída_yyyyMMdd_HHmms.csv“. Parametry *pokus* a *třída* obsahují uživatelem zadané řetězce. Následuje datum a čas v uvedeném formátu. Načtená data jsou zobrazena ve formě grafu, kde každá osa akcelerometru je vykreslena jinou barvou.

Stisknutím tlačítka „Zpracovat“ se spustí serializace objektu TypeRequest do RESTového řetězce. Objekt TypeRequest obsahuje kromě naměřených dat i informaci o identifikaci pokusu a klasifikační třídě. Tento serializovaný objekt je zaslán asynchronní metodou post do serverové části. V nižších verzích SDK pro Android bylo možné komunikovat i synchronně. Z důvodů vytížení hlavního vlákna byl tento způsob v minulosti zakázán a není již podporován. Při příjmu odpovědi (objekt TypeResponse) je tato zpracována a podle typu dat následně zobrazena uživateli.

7 REALIZOVANÉ EXPERIMENTY

Na laboratorním přípravku byly navrženy a zrealizovány dva experimenty z oblasti detekce anomálií. Pro tyto experimenty byly vytvořeny algoritmy v serverové části, které podle postupů z teoretické části práce zpracovávají a vyhodnocují naměřená data z měřicího modulu. U obou experimentů mohou studenti ve cvičení nad změřenými daty sami realizovat potřebné algoritmy k detekci anomálií metodami, jak jsou popsány u vybraných experimentů. Tyto kroky by realizovaly jednak v prostředí aplikace Matlab, kde jako součást práce jsou vytvořeny vzorové algoritmy pro výpočet příznaků a v prostředí RapidMiner kde jako součást práce jsou vytvořeny modely pro výpočty obou experimentů. Zdrojové kódy pro výpočty příznaků v Matlab i předpřipravené modely pro RapidMiner jsou uloženy jako příloha.



Obrázek 36 Ukázka naměřených průběhů a FFT pro jedno měření v prostředí Matlab

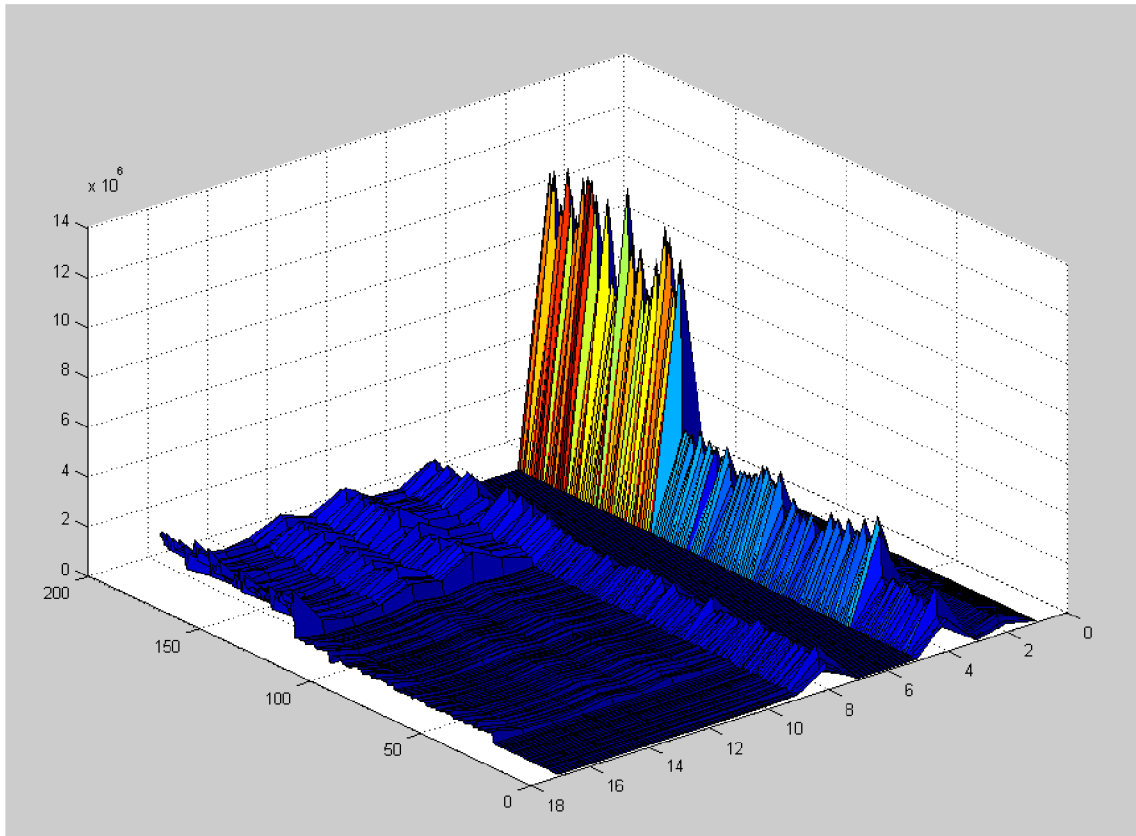
Ukázka algoritmu pro výpočet celkového harmonického zkreslení v prostředí Matlab:

```
%3 - THD
for i=1:sloupce
    p1 = priznaky(1,i); %kmitočet
    p2 = priznaky(2,i); %hodnota
    pocet = floor((Nsamps/2)/p1); %počet harmonických co mám k
    dispozici
    c = 0; %součet harmonických
    for j=2:Nsamps/2
        for k=2:pocet %pro jednotlivé harmonické
            if j == k*p1
                c = c + f(j,i)^2;
            end
        end
    end
end
```

```

        end
    end
end
priznaky(3,i) = sqrt(c)*100/p2; %THD
end

```



Obrázek 37 Vizualizace vypočtených příznaků pro 180 měření v Matlab

7.1 První experiment – klasifikace

Jedná se o učení s učitelem, kde pro klasifikaci je použita metoda k-NN popisovaná v teoretické části v kapitole 2.7.1.1. Nastavení k-NN je na počet 3.

Popis a postup realizace experimentu

1. Všechna měření tohoto experimentu jsou prováděna s nastavením mobilní aplikace na pokus s názvem „1“ (první experiment).
2. Získání dat z přípravku pro fázi učení
 - a. Jsou měřena data pomocí prázdného přípravku. V mobilní aplikaci je nastaven název třídy „000“ (bez kuliček – prázdný přípravek). Je provedeno 30 měření a data jsou vždy odeslána ke zpracování na serveru.

- b. Jsou měřena data přípravkem, do něž byla vložena jedna kovová kulička. V mobilní aplikaci je nastaven název třídy „00lv1“. Je provedeno 30 měření a data jsou vždy odeslána ke zpracování na serveru.
- c. Jsou měřena data přípravkem, do něž byly vloženy dvě kovové kuličky. V mobilní aplikaci je nastaven název třídy „00lv2“. Je provedeno 30 měření a data jsou vždy odeslána ke zpracování na serveru.
- d. Jsou měřena data přípravkem, do něž bylo vloženo pět kovových kuliček. V mobilní aplikaci je nastaven název třídy „00lv5“. Je provedeno 30 měření a data jsou vždy odeslána ke zpracování na serveru.
- e. Jsou měřena data přípravkem, do něž byla vložena jedna skleněná kulička. V mobilní aplikaci je nastaven název třídy „00s1“. Je provedeno 30 měření a data jsou vždy odeslána ke zpracování na serveru.
- f. Jsou měřena data přípravkem, do něž byly vloženy dvě skleněné kuličky. V mobilní aplikaci je nastaven název třídy „00s2“. Je provedeno 30 měření a data jsou vždy odeslána ke zpracování na serveru.

3. Extrahování a výběr příznaků

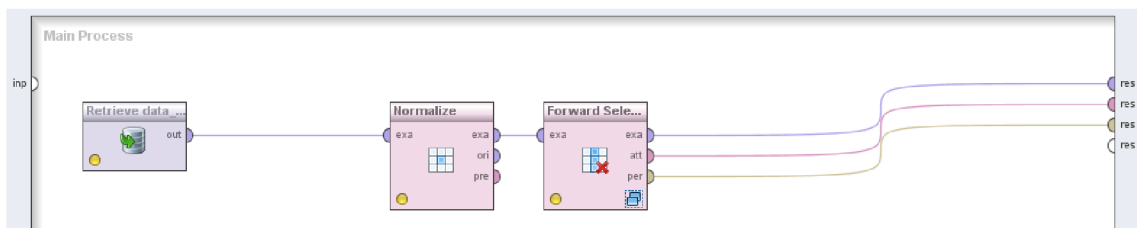
- a. Z dat uložených na serveru byly v prostředí aplikace Matlab vyextrahovány příznaky pro každou osu akcelerometru algoritmy uvedenými v kapitole 2.5. Příznaky byly označeny ve formátu „fx01“. Kde „x“ značí osu měření a „01“ číslo příznaku. Příznaky byly číslovány takto:
 - i. „01“ Hodnota spektra s největší amplitudou (kapitola 2.5.1)
 - ii. „02“ Hodnota maximální složky spektra (kapitola 2.5.2)
 - iii. „03“ Celkové harmonické zkreslení signál (kapitola 2.5.3)
 - iv. „04“ Hodnota součtu složek celého spektra (kapitola 2.5.4)
 - v. „05“ Odchylka signálu od jeho klouzavého průměru (kapitola 2.5.6)
 - vi. „06“ Počet průchodu signálu nulou (kapitola 2.5.7)
 - vii. „07“ Průměrná hodnota signálu (kapitola 2.5.8)
 - viii. „08“ – „17“ Součet amplitud spektra rozdělený do 10 částí (kapitola 2.5.5)

Celkem bylo tedy vyextrahováno $3 * 17 = 51$ příznaků pro jedno měření („fx01“ – „fx17“, „fy01“ – „fy17“ a „fz01“ – „fz17“).

- b. Výběr příznaků, které mají největší vliv na dosažení nejlepší přesnosti metody k-NN byl proveden v aplikaci Rapid Miner (dále jen RM) ve verzi 5.3.013, která je dostupná na adrese: <https://rapidminer.com/>. Výsledkem byla přesnost metody 94,48% pro počet příznaků 11. Pro implementaci byl snížen počet vybíraných příznaků na 5, jako kompromis mezi počtem nutných příznaků a dosaženou přesností (pro 5 příznaků přesnost 91,16%). Vybrány byly příznaky: fx04, fz02, fy05, fx06 a fx05.

4. Na serveru byly vypočteny hodnoty příznaků: fx04, fz02, fy05, fx06 a fx05 pro data z fáze učení
5. Vyhodnocovací fáze
 - a. Na měřicím přípravku bylo provedeno měření s vloženou jednou kovovou kuličkou. Název třídy v mobilní aplikaci nebyl vyplněn. Po zaslání dat na server k vyhodnocení na naučeném modelu byla obdržena odpověď: „Název třídy 00lv1“ což odpovídá vložené jedné kovové kuličce.

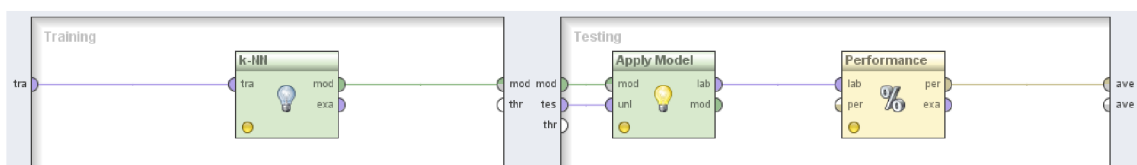
Ukázka výběru atributů v prostředí RapidMiner:



Obrázek 38 Model v RM pro k-NN



Obrázek 39 Detail bloku Forward Selection



Obrázek 40 Detail bloku Validation

PerformanceVector

```

PerformanceVector:
accuracy: 91.16% +/- 28.39% (mikro: 91.16%)
ConfusionMatrix:
True:  000      00lv1    00lv2    00lv5    00s1    00s2
000:   30       0         0         1         0         0
00lv1: 0        25         6         0         0         0
00lv2: 0         5        24         2         0         0
00lv5: 0         0         0        27         0         0
00s1:  0         0         0         0        29         0
00s2:  0         0         0         0         1        30
  
```

Obrázek 41 Výsledek výpočtu k-NN pro 5 příznaků

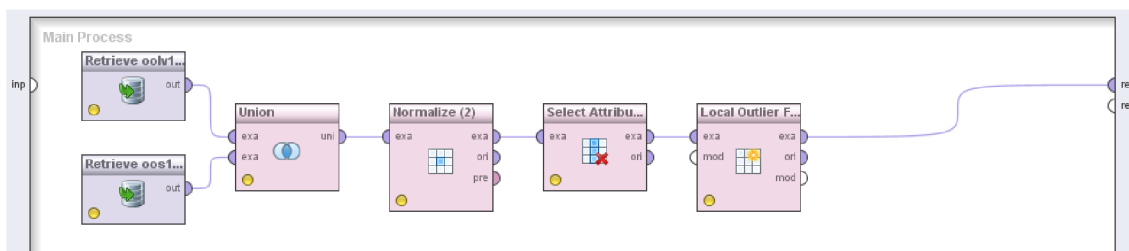
7.2 Druhý experiment – míra odlišnosti

Druhý navržený experiment implementuje algoritmus pro výpočet LOF (Local Outlier Factor). Jedná se o metodu učení bez učitele. Na základě využití stejné množiny příznaků jako v prvním experimentu je nad množinou měření stejného pokusu pro nové měření počítána hodnota příznaku LOF. Nastavení LOF (minimální počet měření) je v algoritmu na serveru nastaven na počet 20.

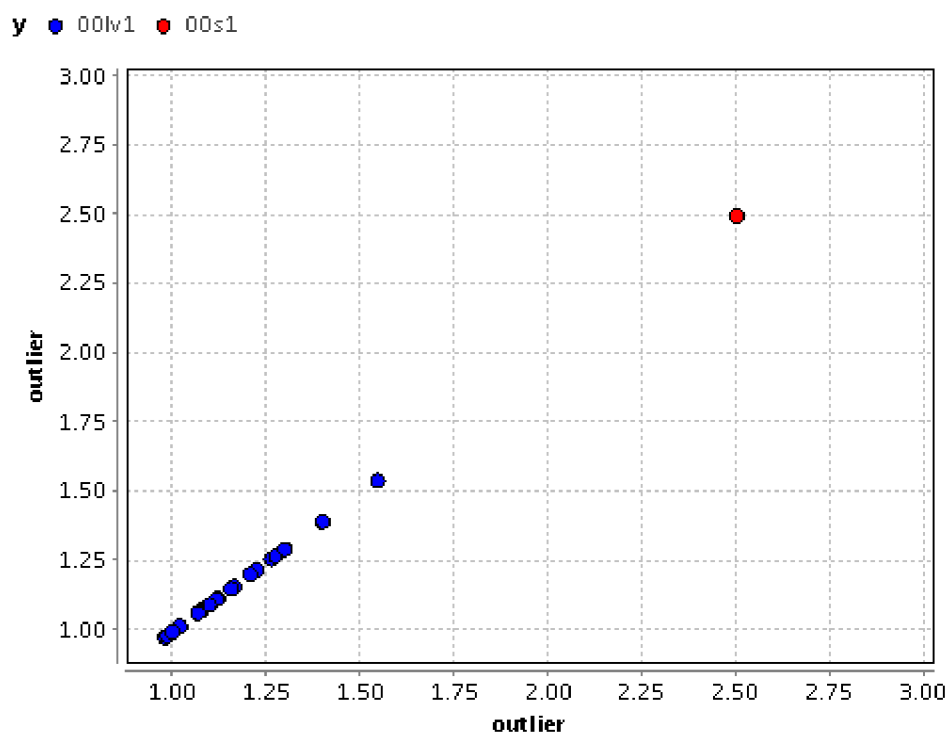
Popis a postup realizace experimentu

1. Všechna měření tohoto experimentu jsou prováděna s nastavením mobilní aplikace na pokus s názvem „2“ (druhý experiment). Název třídy není vyplněn (nemá vliv na tento experiment).
2. Je provedeno 20 měření pomocí přípravku s vloženou jednou kovovou kuličkou. Data jsou odeslána ke zpracování na server.
3. Je provedeno měření s vloženou jednou kovovou kuličkou. Po zaslání dat na server k vyhodnocení byla obdržena odpověď: „Hodnota LOF: 1,145“.
4. Je provedeno měření s vloženou jednou skleněnou kuličkou. Po zaslání dat na server k vyhodnocení byla obdržena odpověď: „Hodnota LOF: 2,500“. Byl detekována větší míra odlišnosti.

Ukázka řešení v prostředí RapidMiner:



Obrázek 42 Model pro LOF



Obrázek 43 Výsledek metody LOF

8 ZÁVĚR

Práci tvoří část teoretická a praktická. V rámci teoretické části jsou uvedeny příklady využití detekce anomálií na různých praktických problémech. Následuje popis různých přístupů k extrakci příznaků zejména z dat dynamických a sekvenčních a dále pak stručný přehled metod používaných pro detekci outlierů, respektive anomálních hodnot. Následuje část praktická, která začíná koncepčním návrhem celého systému od sběru dat po vizualizaci výsledků. Návrh systému je rozdělen do tří částí, které spolu komunikují bezdrátovými technologiemi. Systém tedy umožňuje provádět sběr dat i bez přímého kontaktu mezi uživatelem a měřicím modulem, který může být umístěn přímo v technologickém zařízení.

První část systému je navržený a zrealizovaný měřicí přípravek umožňuje získat reálná data pro další zpracování. Je tvořen kmitajícím nosníkem s připevněným akcelerometrem. Naměřené hodnoty načítá mikrokontroler, který komunikuje s mobilním zařízením pomocí rozhraní bluetooth.

Pro získání dat z měřicího modulu přípravku je navrženo použít mobilní zařízení. Pro účely použití ve školní laboratoři je možné vytvořit jednoduchou aplikaci běžící na PC, pro komunikaci s přípravkem bez nutnosti použití mobilního zařízení.

Struktura a funkcionalita mobilní aplikace je navržena s ohledem na možné rozšíření na další operační systémy mobilní platformy. Neobsahuje žádné speciální komponenty vázané na vybranou platformu. Vytvořená aplikace je pro platformu Android verze 4.4. a vyšší.

Serverová část je navržena v souladu s aktuálními trendy pro serverové aplikace umožňující provoz na libovolné platformě operačního systému. Je kompletně objektově navržena a zrealizovaná s využitím volně dostupných knihoven. Obsahuje metody pro serializaci a deserializaci dat do XML. Tyto uvedené vlastnosti jí zaručují snadnou správu, rozšiřitelnost a možnou výměnu dat s okolními systémy.

Provedené experimenty potvrdily funkčnost celého systému a již v tomto roce posloužily jako zdroj experimentálních dat pro další diplomovou práci. Za úspěšně dokončené považují laboratorní přípravek, mobilní aplikaci, serverové rozhraní a komunikační propojení mezi těmito částmi. Rezervy a prostor pro navazující práci spočívají v implementaci dalších algoritmů pro detekci anomálií a návrhy složitějších experimentů; rozsah realizované práce v této oblasti lze považovat za primárně ukázkový a návodný.

Literatura

- [1] CHANDOLA, Varun, Arindam BANERJEE a Vipin KUMAR. 2009. *Anomaly detection*. *ACM Computing Surveys*. 41(3): 1-58. DOI: 10.1145/1541880.1541882. ISSN 03600300. Dostupné také z: <http://www-users.cs.umn.edu/~banerjee/papers/09/anomaly.pdf>
- [2] Arindam Banerjee, Varun Chandola, Vipin Kumar, Jaideep Srivastava, *Anomaly Detection: A Tutorial* [online] [cit. 2015-03-20]
Dostupné z: <https://www.siam.org/meetings/sdm08/TS2.ppt>
- [3] Vanda Vintrová, Tomáš Vintr, Hana Řezanková, Vladimír Úradníček, *Porovnání vybraných algoritmů pro ohodnocení odlehlosti vícerozměrných pozorování* [online] březen 2014 [cit. 2015-05-03]
Dostupné z: http://www.cs.cas.cz/imapa/download/pubs/IB_1_2014.pdf
- [4] Mgr. Marta Vomlelová PhD. , *Učení založené na instancích* [online] [cit. 2015-03-20]
Dostupné z: <http://kti.mff.cuni.cz/~marta/sliInstance.pdf>
- [5] Carolina Ruiz, *Ruiz's Notes on Anomaly Detection* [online] [cit. 2015-03-20]
Dostupné z:
http://web.cs.wpi.edu/~ruiz/KDDRG/Resources/AnomalyDetection/ruiz_notes_on_anomaly_detection.doc/
- [6] Jan Kelbel, David Šilhán, *Shluková analýza* [online] [cit. 2015-04-08].
Dostupné z:
<http://www.fd.cvut.cz/personal/nagyivan/Projekty/Classification/ShlukovaAnaliza.pdf>
- [7] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander, *LOF: Identifying Density-Based Local Outliers*, [online] [cit. 2015-05-07].
Dostupné z: <http://www.it.iitb.ac.in/~deepak/deepak/courses/mtp/papers/LOF-identifying%20density-based%20local%20outliers.pdf>
- [8] Jeroen H.M. Janssens, Eric O. Postma, *One-Class Classification with LOF and LOCI: An Empirical Comparison* [online] [cit. 2015-04-06].
Dostupné z:
http://www.researchgate.net/profile/Eric_Postma/publication/228848576_One-class_classification_with_LOF_and_LOCI_An_empirical_comparison/links/09e41510fde2b72d81000000.pdf
- [9] Charu Aggarwal, *Outlier* [online] [cit. 2015-04-20]
Dostupné z: <http://charuaggarwal.net/outlierbook.pdf>
- [10] Trevor Whitney, *Anomaly detection* [online] [cit. 2015-04-20]
Dostupné z: http://Trevorwhitney.Com/Data_Mining/Anomaly_Detection
- [11] *Chapter 13. Web MVC Framework* [online] [cit. 2015-04-06].
Dostupné z: <http://docs.spring.io/spring-framework/docs/2.0.8/reference/mvc.html>

- [12] Dan Kernler , *A visual representation of the Empirical (68-95-99.7) Rule based on the normal distribution*. [online] 2014 [cit. 2015-03-20]
Dostupné z: http://commons.wikimedia.org/wiki/File:Empirical_Rule.PNG#mediaviewer/File:Empirical_Rule.PNG
- [13] Rudolf PECINOVSKÝ, *Návrhové vzory: 33 vzorových postupů pro objektové programování*, Computer Press 2007, ISBN 978-80-251-1582-4
- [14] *Fast Fourier transform* . [online] 2015 [cit. 2015-03-20]
Dostupné z: http://en.wikipedia.org/wiki/Fast_Fourier_transform
- [15] Prof. Ing. Pavel Jura, CSc., *Signály a systémy, Část 3: Diskrétní signály a diskrétní systémy* srpen 2010
- [16] *Objektově orientované programování* [online] [cit. 2015-04-06].
Dostupné z: http://cs.wikipedia.org/wiki/Objektov%C4%9B_orientovan%C3%A9_programov%C3%A1n%C3%AD
- [17] *ADXL345* [online] [cit. 2015-05-06].
Dostupné z: <http://www.analog.com/en/products/mems/mems-accelerometers/adxl345.html#product-overview>
- [18] *FRDM-KL25Z* [online] [cit. 2015-04-01].
Dostupné z: <http://developer.mbed.org/platforms/KL25Z/>
- [19] *Bluetooth BT převodník HC-06 USB RS232 TTL pro Arduino HC06* [online] [cit. 2015-04-06].
Dostupné z: <http://www.easyduino.cz/Bluetooth-BT-prevodnik-HC-06-USB-RS232-TTL-pro-Arduino-HC06-d102.htm?tab=description>
- [20] *IMU DIGITAL COMBO BOARD - 6 DEGREES OF FREEDOM ITG3200/ADXL345* [online] [cit. 2015-04-01].
Dostupné z: <http://electronhobbies.com/sensors/138-imu-digital-combo-board-6-degrees-of-freedom-itg3200-adxl345-sf-sen-10121.html>
- [21] *FRDM-KL25Z: Freescale Freedom Development Platform for Kinetis KL14, KL15, KL24, KL25 MCUs* [online] [cit. 2015-04-02].
Dostupné z: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FRDM-KL25Z&nodeId=015210045A#prettyPhoto
- [22] *Bluetooth* [online] leden 2015 [cit. 2015-04-02].
Dostupné z: <http://cs.wikipedia.org/wiki/Bluetooth>
- [23] ŘEZANKOVÁ, Hana, Dušan HÚSEK a Václav SNÁŠEL. *Shluková analýza dat*. 1. vyd. Praha: Professional Publishing, 2007, 196 s. ISBN 978-80-86946-26-9
- [24] *PHP* [online] květen 2015 [cit. 2015-04-06].
Dostupné z: <http://en.wikipedia.org/wiki/PHP>
- [25] *TIOBE Index for April 2015* [online] [cit. 2015-04-06].
Dostupné z: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

- [26] *Java (programovací jazyk)* [online] duben 2015 [cit. 2015-04-06].
Dostupné z: [http://cs.wikipedia.org/wiki/Java_\(programovac%C3%AD_jazyk\)](http://cs.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk))
- [27] *Android (operační systém)* [online] duben 2015 [cit. 2015-04-06].
Dostupné z:
http://cs.wikipedia.org/wiki/Android_%28opera%C4%8Dn%C3%AD_syst%C3%A9m%29
- [28] *Glassfish* [online] říjen 2014 [cit. 2015-04-06].
Dostupné z: <https://glassfish.java.net/index.html>
- [29] *Open Source Initiative* [online] [cit. 2015-04-06].
Dostupné z: <http://opensource.org/>
- [30] *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)* [online] duben 2007 [cit. 2015-04-06].
Dostupné z: <http://www.w3.org/TR/soap12/>
- [31] *RESTful Web services: The basics* [online] [cit. 2015-04-06].
Dostupné z: <https://www.ibm.com/developerworks/webservices/library/ws-restful/>
- [32] *FFT.java*, [online] únor 2007 [cit. 2015-04-06].
Dostupné z: http://www.ee.columbia.edu/~ronw/code/MEAPsoft/doc/html/FFT_8java-source.html
- [33] *InplaceFFT.java* [online] únor 2011 [cit. 2015-03-10].
Dostupné z: <http://introc.cs.princeton.edu/java/97data/InplaceFFT.java.html>
- [34] *Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroléru PIC16F877* [online] květen 2000 [cit. 2015-03-10].
Dostupné z: <http://www.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeprom-24lc256>
- [35] *ADXL345 3 Axis Digital Acceleration Of Gravity Tilt Module For Arduino free shipping* [online] [cit. 2015-03-10].
Dostupné z: <http://www.aliexpress.com/item/Free-Shipping-ADXL345-3-Axis-Digital-Acceleration-Of-Gravity-Tilt-Module-For-Arduino/1825481028.html>
- [36] Ing. Petr Honzík, Ph.D., *Metody statistiky používané ve strojovém učení*
- [37] *Zero-crossing rate* [online] březen 2015 [cit. 2015-04-06].
Dostupné z: http://en.wikipedia.org/wiki/Zero-crossing_rate
- [38] Toto Sugito, *C/C++ Zero Crossing code* [online] květen 2011 [cit. 2015-04-06].
Dostupné z: <http://toto-share.com/2011/05/cc-zero-crossing-code/>
- [39] *MySQL 5.0 Reference Manual :: 1 General Information* [online] [cit. 2015-04-06].
Dostupné z: <http://dev.mysql.com/doc/refman/5.0/en/introduction.html>
- [40] Steven Davelaar, *Performance Study – REST vs SOAP for Mobile Applications* [online] únor 2015 [cit. 2015-05-01].
Dostupné z: <http://www.ateam-oracle.com/performance-study-rest-vs-soap-for-mobile-applications/>

- [41] *Spring Framework* [online] [cit. 2015-05-01].
Dostupné z: <http://projects.spring.io/spring-framework/>
- [42] IBATIS [online] prosinec 2013 [cit. 2015-05-01].
Dostupné z: <http://cs.wikipedia.org/wiki/IBATIS>
- [43] *2. Introduction to the Spring Framework, Part I. Overview of Spring Framework* [online] [cit. 2015-05-01].
Dostupné z: <http://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/overview.html>

Seznam zkratek

GUI	Graphical User Interface – Grafické uživatelské rozhraní
SW	Software
HW	Hardware
AS	Aplikační server
SQL	Structured Query Language - Strukturovaný dotazovací jazyk
REST	Representational State Transfer – Architektura rozhraní
API	Application Programming Interface – Rozhraní pro programové aplikace
JSON	JavaScript Object Notation – JavaScriptový objektový zápis
JavaScript	Programovací jazyk pro webové prohlížeče
PHP	Hypertext Preprocessor – skriptovací programovací jazyk
SOAP	Simple Object Access Protocol – Jednoduchý objektový protokol pro výměnu dat ve formátu XML
XML	Extensible Markup Language - Rozšiřitelný značkovací jazyk
FFT	Fast Fourier transform - Rychlá Fourierova transformace
UML	Unified Modeling Language - grafický jazyk pro vizualizaci
J2EE	Java 2 Enterprise Edition – Java edice enterprise
DAO	Data access object – objekty pro komunikaci s databází
IDE	Integrated Development Environment – vývojové prostředí

Seznam příloh

Příloha 1. CD obsahující:

- elektronickou verzi diplomové práce
- zdrojové kódy aplikace měřicího modulu
- zdrojové kódy mobilní aplikace
- zdrojové kódy serverové aplikace
- objektový model serverové aplikace
- skript pro vytvoření tabulek v databázi na serveru
- ukázka naměřených dat
- změřený průběh komunikace na I²C sběrnici
- program pro Matlab na výpočet příznaků
- model pro RapidMiner pro experimenty