



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

DEPARTMENT OF COMPUTER SYSTEMS

**VYUŽITÍ STROJOVÉHO UČENÍ PRO ZVÝŠENÍ ROBUST-  
NOSTI URČENÍ POZICE V BEZDRÁTOVÉM POZIČNÍM  
SYSTÉMU**

USING MACHINE LEARNING FOR IMPROVEMENT OF LOCATION ESTIMATION IN WIRELESS  
POSITIONING SYSTEM

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. ADAM MATUŠ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. VÁCLAV ŠIMEK**

BRNO 2022

## Zadání diplomové práce



Student: **Matuš Adam, Bc.**  
Program: Informační technologie a umělá inteligence  
Specializace: Inteligentní systémy  
Název: **Využití strojového učení pro zvýšení robustnosti určení pozice v bezdrátovém pozičním systému**  
**Using Machine Learning for Improvement of Location Estimation in Wireless Positioning System**

Kategorie: Umělá inteligence

Zadání:

1. Seznamte se s poziční platformou od společnosti Sewio využívající technologii UWB (širokopásmová komunikace) a stávající verzi algoritmu pro určení pozice objektu na základě multilaterace pomocí metody rozdílu času (TDoA).
2. Na základě poznatků z bodu 1) zadání stanovte parametry algoritmu, jejichž optimalizace je relevantní z pohledu zvýšení přesnosti a robustnosti lokalizace.
3. Analyzujte formát vzorku lokalizačních dat poskytnutých společností Sewio. Dále proveďte v existující instalaci lokalizačního systému vlastní akvizici dat.
4. Zabývejte se aktuálními trendy v oblasti strojového učení. Na základě těchto poznatků zvolte metody vhodné pro optimalizaci parametrů z bodu 2) zadání.
5. Proveďte implementaci vámi zvolených technik. Za tímto účelem můžete pro samotnou realizaci využít existujících knihoven a frameworků.
6. Implementované techniky otestujte v prostředí reálné instalace lokalizačního systému společnosti Sewio. Analyzujte jejich dopad na zpřesnění výpočtu pozic a celkové zvýšení robustnosti.
7. Zhodnoťte dosažené výsledky a pokuste se navrhnout případná rozšíření či další perspektivní směry vývoje.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Šimek Václav, Ing.**

Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 18. května 2022

Datum schválení: 29. října 2021

## Abstrakt

Tato diplomová práce se zabývá seznámením se s poziční platformou Sewio a technikami širokopásmové komunikace UWB, které platforma využívá pro určování pozic objektů. Stávající technologie je založena na měření časových intervalů příchodu signálů a multilateraci pomocí časových rozdílů. Platforma generuje a ukládá historická data z proběhlé lokalizace objektů. Datovou sadu tvoří sekvence pozičních dat, které kromě relativních souřadnic obsahují i relevantní signálové parametry bezdrátové komunikace. Po analýze pozičních dat z reálných instalací byl navržen a implementován systém strojového učení založený na technikách klasifikace Gaussovským rozložením a predikce lineární regresi. Systém funguje jako komponenta, jejíž vstupem jsou poziční data existujícího RTLS systému a výstupem je robustnější odhad pozic. Evaluační výsledky ukazují zlepšení stability pozic a odstranění konfliktních a odskočených souřadnic.

## Abstract

This thesis describes the Sewio platform and the communication techniques of the ultra-wideband technology standard, which the platform uses to determine the position of objects. The technology is based on measuring signal arrival time intervals and multilateration using time differences. The platform generates and stores historical data from past positioning of objects. The dataset consists of sequences of position data which, in addition to the monitored environment, contain relevant signal parameters of wireless communication. A system of machine learning techniques based on Gaussian models and linear regression was implemented to classify and predict real-time position data with the goal of improving position estimation stability and robustness. The system functions as a downstream component, which accepts RTLS position data and outputs improved position estimates. The evaluation results show that the implemented system can successfully improve position stability and robustness.

## Klíčová slova

Strojové učení, lokalizační systém v reálném čase, širokopásmová komunikace (UWB), lokalizace uvnitř budov, klasifikace, predikce

## Keywords

Machine learning, real-time location system (RTLS), ultra-wideband (UWB), indoor positioning, classification, prediction

## Citace

MATUŠ, Adam. *Využití strojového učení pro zvýšení robustnosti určení pozice v bezdrátovém pozičním systému*. Brno, 2022. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

# Využití strojového učení pro zvýšení robustnosti určení pozice v bezdrátovém pozičním systému

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka. Další technické informace mi poskytl pan Jiří Zvěřina z firmy Sewio. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Adam Matus  
13. května 2022

## Poděkování

Chtěl bych poděkovat vedoucímu práce Václavu Šimkovi za organizační rady a panu Jiřímu Zvěřinovi z firmy Sewio za poskytnutí technických prostředků.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Lokalizační platforma</b>	<b>4</b>
2.1	Bezdrátová technologie UWB . . . . .	4
2.2	Lokalizační techniky . . . . .	6
2.3	Komunikace a synchronizace . . . . .	7
2.4	Typy instalací . . . . .	9
2.5	Signálové charakteristiky . . . . .	10
2.5.1	Indikátor síly přijímaného signálu . . . . .	10
2.5.2	Viditelnost . . . . .	10
2.5.3	Vícecestná propagace . . . . .	12
<b>3</b>	<b>Techniky strojového učení</b>	<b>14</b>
3.1	Typy strojového učení . . . . .	14
3.2	Vlastnosti dat pro strojové učení . . . . .	15
3.3	Učení bez učitele . . . . .	17
3.4	Lineární regresní model . . . . .	18
3.5	Hluboké učení . . . . .	19
3.6	Nástroje a frameworky . . . . .	20
<b>4</b>	<b>Analýza lokalizačních dat</b>	<b>21</b>
4.1	Analýza datasetu . . . . .	21
4.1.1	Analýza signálových parametrů . . . . .	23
4.1.2	Analýza označeného vzorku . . . . .	24
<b>5</b>	<b>Implementace řešení</b>	<b>27</b>
5.1	Klasifikace Gaussovským modelem . . . . .	28
5.1.1	Hyperparametry . . . . .	29
5.1.2	Analýza modelů . . . . .	29
5.2	Predikce trajektorie . . . . .	33
5.2.1	Hyperparametry . . . . .	33
5.3	Vizualizační nástroje . . . . .	36
<b>6</b>	<b>Evaluační řešení</b>	<b>38</b>
6.1	Parametry modelů . . . . .	38
6.2	Skóre stability . . . . .	39
6.3	Eliminace odskoků . . . . .	40
6.4	Měření výkonu . . . . .	42

6.5	Perspektivy budoucí práce . . . . .	43
6.5.1	Kooperace UWB a lidarů . . . . .	43
6.5.2	Simulace prostředí . . . . .	44
<b>7</b>	<b>Závěr</b>	<b>45</b>
	<b>Literatura</b>	<b>46</b>
<b>A</b>	<b>Obsah paměťového média</b>	<b>48</b>
<b>B</b>	<b>Použití vizualizačního nástroje</b>	<b>49</b>

# Kapitola 1

## Úvod

Ve světě moderního průmyslu se stále více uplatňují systémy umělé inteligence jako nepostradatelná součást inteligentní platformy. Stávající systémy se obohacují technikami strojového učení, kde bývá například cílem automatizace procesu kalibrace zařízení nebo modelování inteligentního systému, který dokáže na základě změn prostředí adaptovat své chování. Jednou z komponent průmyslového systému může být lokalizační systém, který sleduje a odhaduje pozice objektů v reálném čase v monitorovaném prostoru. Tímto prostorem jsou například skladové nebo výrobní haly, ve kterých nelze použít venkovní lokalizaci typu GPS. Požadavky na takový lokalizační systém jsou přesnost v řádech centimetrů, zpracování v reálném čase a v neposlední řadě cena a jednoduchost instalace.

Tato diplomová práce se konkrétně zaměřuje na lokalizační platformu od společnosti Sewio. Firma již mnoho let vyvíjí a prodává řešení průmyslové lokalizace s názvem Real-Time Location System (RTLS). Produkt je založen na standardizované bezdrátové komunikační technologii ultra-wideband (UWB). Instalaci tvoří sada UWB zařízení, která jsou připevněna v monitorovaném areálu. Na pohybujícím se objektu je potom umístěno jiné UWB zařízení, které signalizuje svoji pozici pomocí krátkých UWB signálů. Zachycené signály zpracovává centrální RTLS server, který pomocí multilateračního algoritmu dokáže určit pozici objektu v kartézském souřadném systému. Tento způsob určování pozice obvykle funguje korektně v instalacích, které zaručují dostatečné pokrytí přijímači. Společnost Sewio se nyní snaží implementovat spolehlivou instalaci, kde je pokrytí slabší. V těchto instalacích zpravidla dochází k lokalizačním nepřesnostem a odskokům pozic, které současný systém nedokáže spolehlivě řešit.

Řešení implementované v této práci pracuje s výstupními daty stávajícího RTLS řešení. Nejdříve jsou data modelována jako Gaussovská rozdělení v prostoru signálových parametrů, které popisují kvalitu pozice. Model je následně použitý jako klasifikátor, který je nasazen pro klasifikaci a hodnocení nově vznikajících pozic. Pozice jsou dále zpracovány jako časová řada, ze které je možné odhadovat trajektorii pohybu objektu. Navržený systém využívá technik strojového učení pro věrohodnostní odhad parametrů Gaussovského modelu a pro výpočet vážené lineární regrese. Systém je vyhodnocen na datech, která pocházejí z reálných lokalizačních instalací. Porovnáním s původními pozicemi je ukázáno, že systém dokáže řešit konfliktní pozice a eliminovat většinu odskoků.

Text práce je strukturován následovně: kapitola 2 se zabývá charakteristikou lokalizační platformy. Kapitola 3 diskutuje techniky strojového učení, zpracování dat a použité knihovní implementace. V kapitole 4 je analyzován vzorek datové sady z RTLS instalace a na základě těchto poznatků je navržen systém v kapitole 5. Kapitola 6 se zabývá evaluací a vyhodnocením systému a navrhuje možná budoucí směřování práce.

## Kapitola 2

# Lokalizační platforma

Tato kapitola se zabývá strukturou platformy RTLS (Real-time location system) Sewio. Platforma je postavena na technologii UWB (Ultra-wideband, širokopásmová komunikace) a nabízí veškerý hardware a software pro průmyslové instalace. Jádrem hardwarového řešení je integrovaný čip DW1000 DecaWave, který implementuje UWB komunikaci. Podstatné pro tuto práci je pochopit techniku multilaterace používanou při měření vzdáleností pomocí signálových impulsů. Vlastní software platformy bude zmíněn jen okrajově. Informace v této kapitole jsou čerpány z veřejných dokumentací společností Sewio [15] a Decawave [4].

### 2.1 Bezdrátová technologie UWB

UWB je bezdrátová technologie, která může fungovat při nízkých úrovních energie na krátké vzdálenosti. Přenos probíhá velmi krátkými impulzy, přenesená informace je tedy obsažena na širokém frekvenčním pásmu, běžně 500 MHz a více. UWB signál nezpůsobuje rušení běžných úzkopásmových radiokomunikačních systémů.

UWB má spoustu výhod pro komunikační aplikace, UWB nabízí konkrétně

- nízkou složitost a potenciálně nízkou cenu,
- charakteristiku signálu podobnou šumu,
- odolnost vůči rušení a vícecestné propagaci signálu a
- vysoké rozlišení v časové doméně.

Jak již bylo zmíněno, UWB využívá k přenosu informací spíše sekvenci impulsů než modulovanou sinusovku. Díky této vlastnosti je ideální pro přesné měření časových intervalů. Vzhledem k tomu, že puls zaujímá tak široké frekvenční pásmo, je jeho vzestupná hrana velmi strmá a to umožňuje přijímači přesně měřit dobu příchodu signálu. Jednotlivé impulsy jsou velmi úzké, obvykle ne více než dvě nanosekundy.

Tabulka 2.1 srovnává vlastnosti různých bezdrátových technologií z pohledu pozicování. Každá technologie má svoje kladné i záporné stránky a UWB není výjimkou. Přesnost pozic může být velmi závislá na použitém typu instalace a výskytu materiálů v monitorovaném prostředí. Signál může být například rušen kovovými materiály. Tento problém lze v praxi obvykle řešit kvalitním rozmístěním sensorových jednotek.



Technologie	Identifikace	Přesnost	Typ pozice	Spotřeba	Vzdálenost [m]
UWB	Ano	cm - dm	Absolutní	Nízká	50
Wi-Fi	Ano	m	Absolutní	Vysoká	50
Bluetooth	Ano	m	Absolutní	Nízká	20
RFID	Ano	dm - m	Absolutní	Nízká	50
Kamera	Ne	mm - dm	Absolutní	Vysoká	10
Laser	Ne	mm	Relativní	Vysoká	5
Infrared	Ne	m	Absolutní	Nízká	5
Ultrazvuk	Ne	cm	Absolutní	Nízká	10

Tabulka 2.1: Porovnání bezdrátových technologií z hlediska lokalizačních schopností

## Standard IEEE

Technologie UWB je definovaná ve standardu IEEE 802.15.4 [8]. Definuje linkovou i fyzickou vrstvu v referenčním modelu ISO/OSI. Původní standard zavedl nové způsoby komunikace v malých sítích WPAN (Wireless Personal Area Networks) s krátkým dosahem a nízkou přenosovou rychlostí. Technologie, jako je například ZigBee, jsou běžně používány pro budování bezdrátových senzorových sítí. Novela 802.15.4a přidala do standardu UWB, což doplnilo vlastnosti, které původnímu standardu chyběly: vyšší datovou propustnost a také možnost velmi přesného určování vzdálenosti. V roce 2011 byla představena revize IEEE 802.15.4–2011, která slouží jako revize původních standardů za účelem jejich sloučení do jednoho dokumentu. Specifikuje tři frekvenční pásma, sub-gigahertz, nízké a vysoké pásmo poskytující celkem 16 rádiových kanálů. Samozřejmě záleží na regulačním úřadu každého státu, které z těchto pásem lze nebo nelze geograficky využít. Norma specifikuje čtyři datové rychlosti pro UWB: 110 kbps, 850 kbps, 6,8 Mbps a 27 Mbps.

## Hardware

Platforma je založena na integrovaném obvodu DW1000 od společnosti DecaWave. Využívá celkem 6 standardizovaných UWB rádiových kanálů, a to jak v nízkých, tak vysokých pásmech UWB. To dává čipu flexibilitu, protože pokud dojde k rušení v jednom kanálu, lze čip snadno přepnout na jiný rádiový kanál. Kanály 1, 2, 3 a 5 jsou definovány šířkou pásma 500 MHz, zatímco kanály 4 a 7 nabízejí šířku pásma přes 1 GHz.

Platforma Sewio ze standardu IEEE 802.15.4 využívá datové rychlosti 110 kbps, 850 kbps a 6,8 Mbps a pracuje pouze na kanále 2 s operační frekvencí v rozsahu 3774 – 4243.2 MHz a kanále 5 s operační frekvencí v rozsahu 6240 – 6739.2 MHz.

Jsou zde 2 druhy fyzických zařízení, které platforma využívá pro lokalizaci.

- Na sledovaném objektu je připevněn tzv. **tag**. Toto malé zařízení funguje jako vysílač impulzů, které zaznamenávají upevněné přijímače. Tagy se běžně instalují na vozidla, například vysokozdvížné vozíky, nebo se integrují do výrobního materiálu, například do dílů automobilů.
- Signály vysílané tagem přijímají tzv. **kotvy** (anchors). Tyto jednotky tvoří sadu upevněných zařízení instalovaných v monitorovaném prostoru. Kotvy přeposílají data serveru, který ze získaných dat počítá pozici tagu. Systém je závislý na LOS (line-of-sight) propagaci signálu, což znamená, že pro ideální fungování musí být sady kotev vzájemně viditelné.

## 2.2 Lokalizační techniky

Systémy využívající UWB jsou skvělými kandidáty pro kombinovanou komunikaci a prostorovou lokalizaci. Mají velmi krátké impulsy v časové doméně, protože šířka pásma přenášeného signálu je nepřímo úměrná době trvání pulsu. Pokud je s přiměřenou přesností znám okamžik příchodu impulsu, lze poměrně přesně odhadnout vzdálenost, kterou impuls urazí od zdroje. K odhadu pozic lze použít jednoduché triangulační techniky kombinací odhadů vzdálenosti z více přijímačů.

Pro představu uvažujme UWB pásmo 7.5 GHz. Jeden signálový puls má v tomto případě šířku 133 pikosekund, z čehož lze předpokládat lokalizační přesnost na přibližně 4 centimetry. Obecně lze z frekvenčního pásma odvodit prostorovou přesnost podle jednoduchého vzorce:

$$s = \frac{c}{f} \quad (2.1)$$

kde rychlost světla  $c = 3 \cdot 10^8$  a  $f$  je frekvence pásma.

Jednou z nejběžnějších metod pro určení pozice je řešení multilaterace pomocí časových rozdílů příchodu signálů, anglicky *Time difference of arrival (TDoA)*. Pro zjištění pozice objektu v dvourozměrném prostoru jsou zapotřebí minimálně 3 stanice a pro třírozměrný prostor jsou potřeba 4 stanice. Na obrázku 2.1 je znázorněn příklad 3 přijímacích stanic, pomocí kterých je možné v ideálním prostředí získat přesnou 2D pozici objektu. Podle časů příchodu signálů od vysílače na sledovaném objektu jsou odvozeny vzdálenosti  $d_1$ ,  $d_2$ ,  $d_3$ . Tyto vzdálenosti lze promítnout do abstraktní kružnice, kde v ideálním případě existuje pouze 1 průsečík všech tří kružnic.

V praxi dochází častěji k případu na obrázku 2.2, kde vzdálenosti nevytvářejí situaci s jedním společným průsečíkem, ale tvoří spíše oblast – *confidence area*. Oblast vznikne protažením (případně zkrácením) skutečných vzdáleností o vzdálenosti  $e_1$ ,  $e_2$  a  $e_3$ . Tento problém je reálně řešitelný různými metodami interpolace nebo optimalizačními algoritmy.

### Přímý výpočet

Uvažujme ideální model, kdy vypočtené vzdálenosti odpovídají reálným vzdálenostem. Mějme 4 stanice v trojrozměrné kartézské soustavě souřadnic, vzdálenost mezi stanicí  $i$  a lokalizovaným tagem je

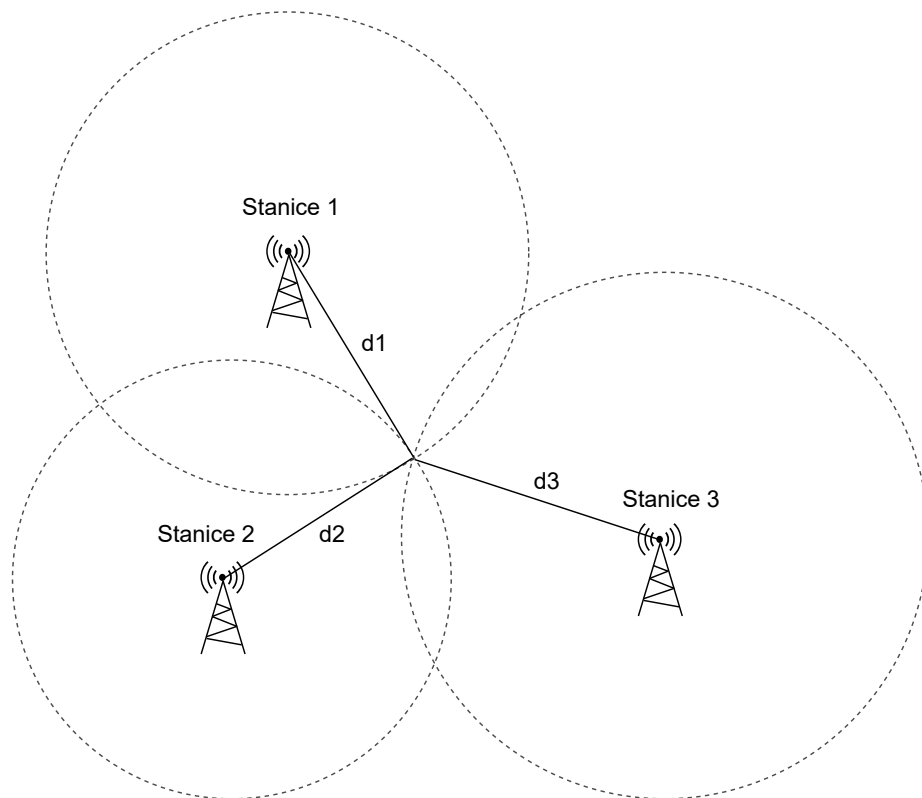
$$\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} = c(t_i - t_0), \quad i = 1, 2, 3, 4 \quad (2.2)$$

kde  $(x, y, z)$  jsou souřadnice tagu,  $(x_i, y_i, z_i)$  jsou souřadnice  $i$ -té stanice,  $c$  je rychlost světla,  $t_i$  je čas příchodu signálu na stanici, a  $t_0$  je neznámý čas vysílání signálu lokalizovaného tagu.

Rozdíl časů příchodu signálu (TDoA) stanic  $i, j$  je

$$\Delta t_{ij} = t_i - t_j \quad (2.3)$$

Z rovnic 2.2 a 2.3 je možné odvodit vztah pro výpočet pozice tagu  $(x, y, z)$ . Celý postup odvození je poměrně složitý a je detailně rozepsaný v knize *UWB Theory and Applications* v kapitole Positioning [11]. Řešení rovnic je několik, ale ve skutečnosti může existovat pouze jedno správné. Pokud jsou řešení velmi blízko sebe, je výhodné vypočítat jejich průměr a ten vzít jako výsledek. Výsledky, které jsou mimo monitorovaný prostor, je možné zahodit, stejně tak výsledky v komplexních číslech. V nejhorsím případě jsou všechna řešení



Obrázek 2.1: Příklad ideální 2D multilaterace

neaplikovatelná a je nutné použít data z více senzorů (pokud jsou k dispozici) nebo použít předchozí měření.

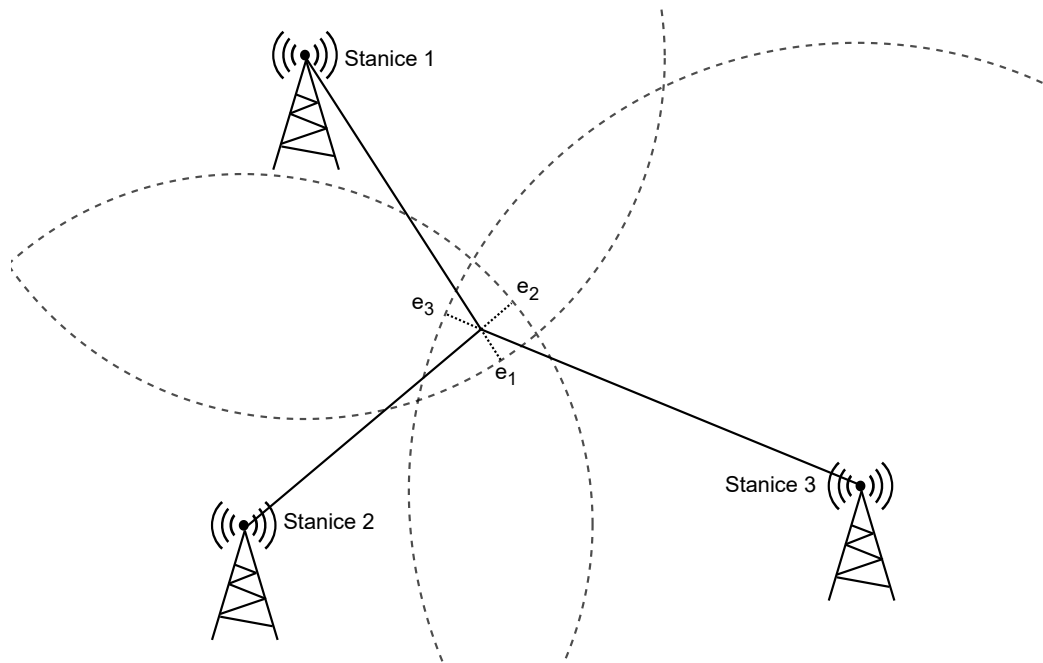
## 2.3 Komunikace a synchronizace

Platforma Sewio je založeno na přesném měření časového rozdílu mezi signály UWB kotvami, které musí být přesně synchronizovány.

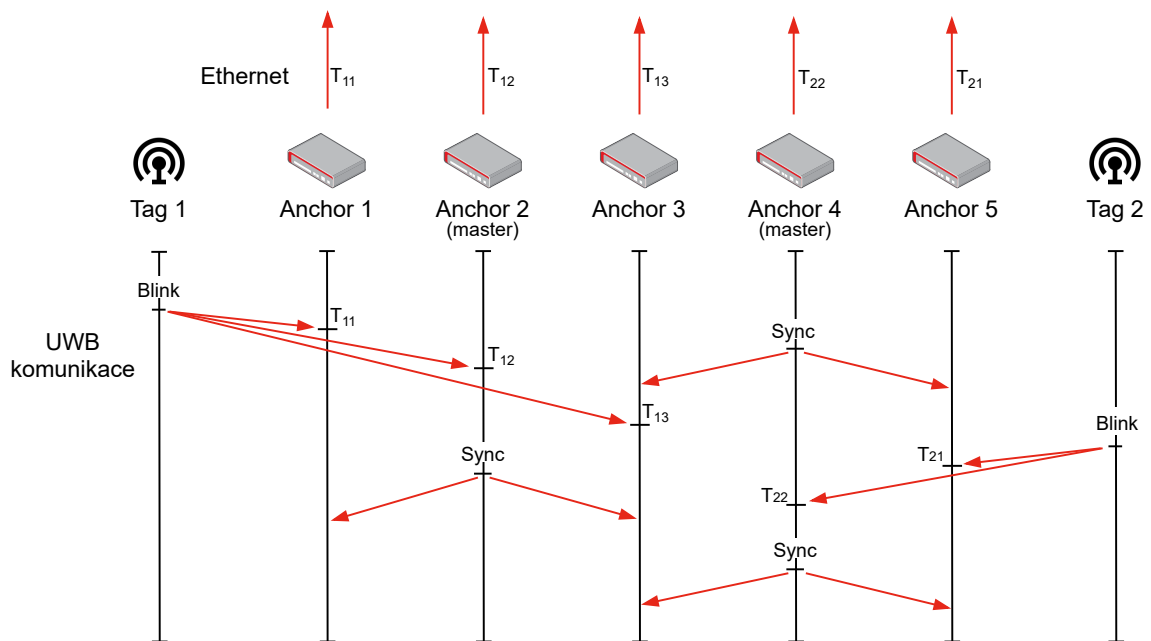
Tag nekomunikuje s kotvami jednotlivě, ale posílá broadcast *blink* zprávy. Tag tedy nemusí znát adresy ani počet anchorů v prostředí. Tento způsob komunikace maximalizuje životnost baterie tagu. Z pohledu škálování systému to také přináší výhodu, protože tag nemusí být přenastaven při změnách v konfiguraci kotev.

Vysílačem UWB signálu může být kotva, pokud probíhá synchronizace, nebo tag, pokud vysílá blink zprávu. Příjímačem jsou potom dále chápány vždy kotvy. V závislosti na kontextu jsou signálové parametry vztažené k roli kotvi, která signál zpracovala. Například MASTER-RSSI je naměřená hodnota RSSI na master kotvě.

Na obrázku 2.3 je sekvence UWB komunikace dvou tagů a několika kotev. Blink zprávy jsou zaslány viditelným kotvám. Tagy vzájemně nejsou synchronizované. Kotvy spolu komunikují jen v rámci synchronizačních impulsů. Kotvy potom komunikují s RTLS serverem pomocí ethernetu.



Obrázek 2.2: Příklad 2D multilaterace s oblastí chyby



Obrázek 2.3: Sekvenční diagram komunikace platformy

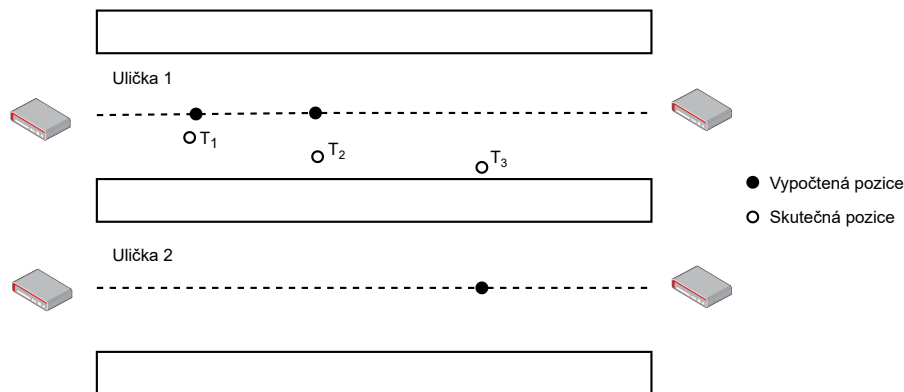
Synchronizace probíhá periodicky a vždy v rámci sady anchorů, kde je zvolena jedna z kotev jako master. Každá kotva musí mít viditelnost na alespoň jednu master kotvu, aby synchronizace mohla fungovat správně. Sada kotev se společnou master kotvou se nazývá buňka (*location cell*).

## 2.4 Typy instalací

Platforma rozlišuje 2 typy rozmístění kotev: 2D buňka a 1D buňka. Běžně používané je rozmístění v režimu 2D, kde jsou jednotlivé lokalizační buňky schopné určovat pozici objektů v dvourozměrném souřadném systému (x,y). Tato instalace se používá v otevřených prostorech. V této instalaci jsou obvykle potřeba minimálně 3 kotvy na buňku, a jejich instalace není jednoduchá. Při instalaci se musí brát v úvahu mnoho faktorů prostředí (nerovnost terénu, vzájemná viditelnost kotev) a je potřeba instalaci kalibrovat (výběr master kotev, výškový rozdíl kotev). Výhodou 2D instalace je poměrně dobrá robustnost a přesnost lokalizace. Instalaci je také možné doplnit redundantními kotvami a díky tomu dosáhnout větší stability a přesnosti vypočtené pozice.

1D buňky využívají pouze 2 kotvy, které jsou instalované naproti sobě. Pozice sledovaného objektu je potom bod (x,y), který leží na úsečce mezi kotvami. Aby tento typ instalace měl smysl, tak monitorovaný prostor musí být dostatečně úzký a zdi, který tento prostor svírají, nesmí propouštět signál. Vhodné jsou například chodby, tunely nebo uličky ve skladech. Právě skladové uličky mohou být problematické, protože zdi jsou zde obvykle kovové rámy s proměnlivým množstvím zboží. Signál tak může procházet do vedlejší chodby a lokalizační systém potom musí správně uhodnout, ve které uličce se lokalizovaný tag nachází.

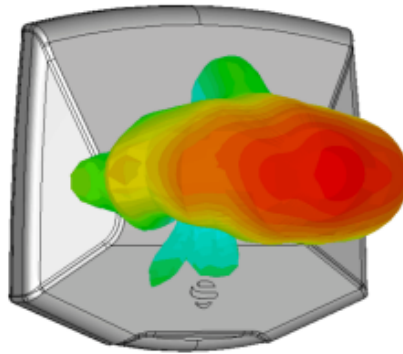
Příklad problematické situace ilustruje obrázek 2.4. Lokalizační 1D buňky jsou paralelně instalovány ve 2 uličkách ve skladovém prostoru. Překážkou jsou zde částečně propustné police se zbožím. Zjištěné pozice  $T_1$  a  $T_2$  jsou relativně správné, ale pozice  $T_3$  chybně odskočila do vedlejší uličky.



Obrázek 2.4: Paralelní instalace 1D buněk a chybná lokalizace

Instalace v režimu 1D vyžaduje kotvy typu DirectFive 2.5, které obsahují směrové přijímače. Tyto antény mají schopnost přijímat signál pouze z určitého úhlu - v horizontálním směru přibližně 90 stupňů a ve vertikálním směru 36 stupňů. Tento prostor lze popsat jako kužel, ze kterého přijímač dokáže zaznamenat UWB signál. Ve správné 1D instalaci jsou páry kotev na sebe přesně natočené v horizontálním směru, aby signál tagu v uličce dosahoval pokud možno nejlepších možných signálových parametrů. Instalace také musí dodržovat pravidla vzdáleností instalovaných kotev: minimální doporučená vzdálenost je 15 metrů a maximální je 30 metrů. Pokud by reálný prostor byl příliš dlouhý, je nutné lokalizaci rozčlenit do více buněk. Jedním z dalších aspektů instalace je vertikální sklon kotvy. Pokud jsou kotvy instalovány vysoko, musí mít dostatečný sklon směrem dolů, aby viděly na tagy blízko podlahy v co největší části monitorované uličky. Zároveň ale nesmí

být sklon k zemi příliš velký, jinak by nebyla možná dostatečně kvalitní synchronizace kotev navzájem.



Obrázek 2.5: Kotva typu DirectFive, která obsahuje směrovou anténu. Zvýrazněná oblast reprezentuje směr přijímání signálu. Obrázek byl převzat z dokumentace Sewio RTLS.

Hlavní nevýhodou těchto antén v instalaci paralelních 1D buněk je možnost konfliktní lokalizace, kdy signál tagu může přijmout více buněk ze sousedních uliček. Je to dáno právě poměrně širokým záběrem přijímací antény v horizontálním směru, a tedy pokud nejsou vzdálenosti uliček dostatečně velké, bude docházet ke kolizím. Současný systém lokalizace tento problém řeší jen částečně, například ručním laděním prahů signálových parametrů a jejich vah.

## 2.5 Signálové charakteristiky

Tato část kapitoly se stručně zabývá některými relevantními signálovými parametry a charakteristikami. Platforma Sewio využívá centrální RTLS server pro zpracování dat z anchorů a také umožňuje ukládat data o proběhlém procesu lokalizace objektů. Data pocházejí z přijímačů, které zaznamenávají signálové parametry. Platforma používá pojmy jako RSSI, FP, FPI a další, jejichž pochopení je podstatné pro další pokračování. Uvedený popis parametrů čerpá z technické a instalační dokumentace společnosti Sewio [15].

### 2.5.1 Indikátor síly přijímaného signálu

Indikátor síly signálu (nejlépe známý pod anglickou zkratkou **RSSI**) je míra energie přijatého signálu na přijímači. Jednotkou veličiny je dBm (decibel-miliwatt), která udává poměr výkonů na logaritmické škále přijatého a vysílaného signálu.

Platforma Sewio používá RSSI pro změření pokrytí signálu mezi kotvami. Hodnoty se pohybují v záporných číslech, kde vyšší hodnoty indikují lepší pokrytí. Hranicí kvalitní síly signálu je -90 dBm, ke které většinou dochází mezi anchory se vzájemnou přímou viditelností. Parametr RSSI je podstatný pro synchronizaci hodin *master* kotvy s podřazenými *slave* kotvami.

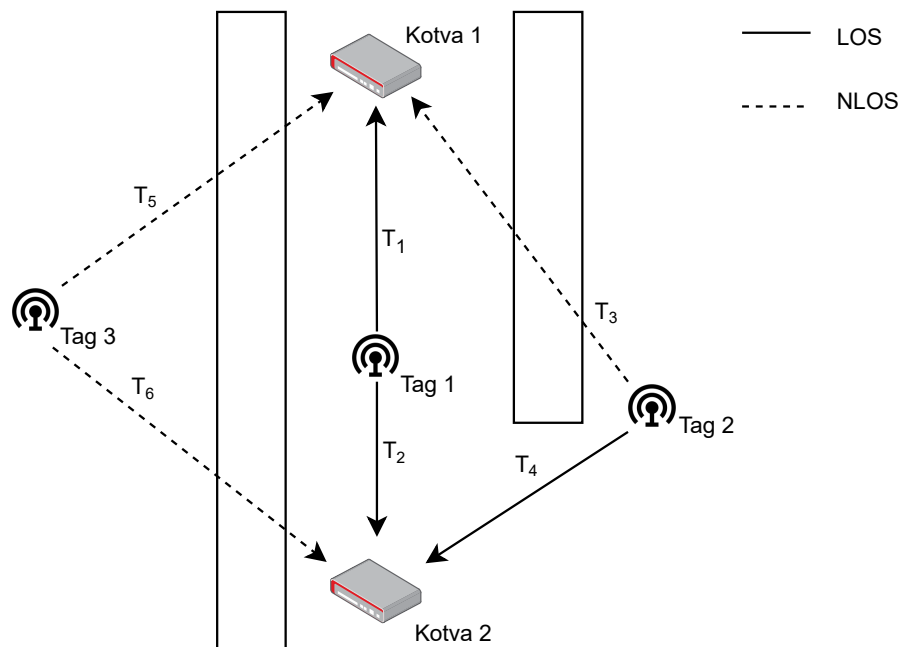
### 2.5.2 Viditelnost

Přímá viditelnost signálu (anglicky line of sight propagation, LOS) je charakteristika elektromagnetického záření, kdy signál prochází vzduchem přímočaře od vysílače k přijímači.

Na překážky v propagaci signálu jsou citlivé především velmi krátké vlny, což je případ technologie UWB. Pojem line of sight vznikl jako analogie pozorování viditelného světla lidským okem. Čím průhlednější jsou materiály na přímé cestě k pozorovanému objektu, tím kvalitnější bude získaný vjem.

Opakem LOS je pojem NLOS (non line of sight), kdy je přímočará cesta signálu částečně nebo úplně blokována. Objekty s vyšší hustotou než vzduch způsobují zpomalení přenášeného signálu, což potenciálně vytváří problém pro lokalizaci založené na TDoA, kde je konzistentní rychlost signálu klíčová. Průchod elektromagnetického záření překážkami také způsobuje značný útlum signálu. Blokující materiály také mohou způsobit lámání (refraction). V závislosti na prostředí se může signál šířit i odrazy od překážek, čímž se dále prodlužuje dráha signálu od předpokládané přímky. NLOS propagace je jedním ze zdrojů nepřesností u UWB lokalizace.

Na obrázku 2.6 je příklad půdorysu prostředí se 2 kotvami a 3 tagy. Signály  $T_1$  a  $T_2$  od vysíláče tag 1 je přijat oběma kotvami přímou cestou. Tag 2 má přímou viditelnost na kotvu 2, ale viditelnost na kotvu 1 je zablokována zdí. Signály  $T_5$  a  $T_6$  vysíláče tagu 3 jsou kompletně blokovány.



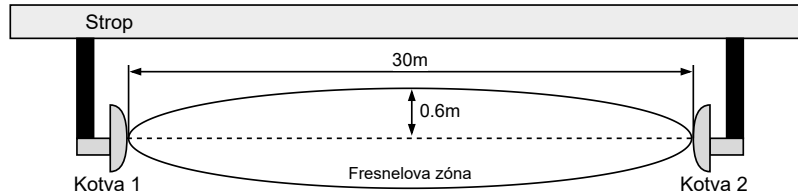
Obrázek 2.6: Příklad propagace signálu LOS a NLOS

### Fresnelova zóna

Fresnelova zóna je prostor ve tvaru elipsoidu mezi vysíláči elektromagnetického vlnění. Primárně se signál šíří přímočaře středem zóny, ale i vedlejším vlněním mimo přímou viditelnost dvou vysíláčů. Vedlejší vlnění se zpravidla šíří delší cestou a k přijímači dojde mírně později, než primární signál. Zóna je nejširší přesně uprostřed mezi vysíláči. Šířka zóny závisí na vzdálenosti vysíláčů a nosné frekvenci bezdrátové technologie.

Intuitivně se může zdát, že přímá viditelnost dvou antén u UWB by měla zaručit kvalitní signál. V porovnání se světlem, které má vlnovou délku ve stovkách nanometrů, je

vlnová délka UWB signálu v jednotkách centimetrů, takže vzniklá zóna se bude pohybovat v těchto rozměrech. Narušení zóny překážkami vede ke zhoršení kvality signálu a destruktivním interferencím. UWB signál lze považovat za neblokovaný pouze pokud se ve Fresnelově zóně nenachází žádná překážka. Platforma Sewio popisuje bezkolizní zónu jako elipsoid s poloměrem 60 centimetrů. Obrázek 2.7 ilustruje Fresnelovu zónu mezi 2 UWB kotvami.



Obrázek 2.7: Fresnelova zóna mezi kotvami

### 2.5.3 Vícecestná propagace

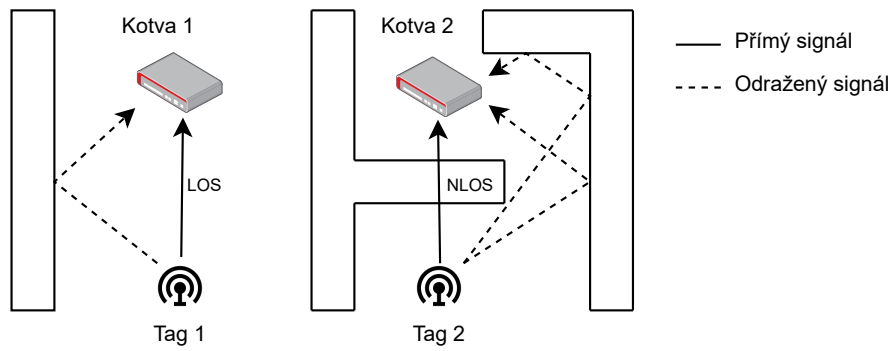
Vícecestná propagace (v anglické literatuře se také používá pojem *multipath*) je jev, kdy je vysílaný radiový signál přijat ze dvou nebo více cest. Příčina tohoto jevu je závislá na použité radiové technologii a prostředí, kterým signál prochází. Nejběžnějším zdrojem vícecestné propagace u UWB je odraz od povrchu překážek a zdí ve vnitřních prostorách. Ve spektru viditelného světla si tento jev lze představit jako místnost se zrcadlem, kde pozorovaný objekt lze vidět přímo a také v odrazu zrcadla.

Síla přijatého signálu se může lišit v závislosti na cestě. V prostředí kde na sebe mají zařízení přímou viditelnost (LOS) má obvykle signál největší sílu na přímé cestě, a signály šířené odrazy a lámáním jsou tlumené a vykazují fázový posun a zpoždění. UWB kotva v drtivé většině případů dokáže správně indikovat sílu přímého signálu jako parametr **FP** (first path). FP je podobný indikátoru RSSI, ale je citlivější na odrazy. Nejlepší hodnotou síly signálu FP je -80 dBm a nejhorší použitelnou je -115 dBm.

Vícecestná propagace znamená pro přijímač problém, protože může docházet k destruktivní interferenci. Přijímač potom není schopný přesně odvodit původní signál, což může vést k chybám v lokalizaci. Tento problém se v praxi daří řešit například pomocí odhadu maximální věrohodnosti (*maximum likelihood estimate*) [14].

Obrázek 2.8 ukazuje příklad dvou situací vícecestné propagace. V prvním případě má signál přímou viditelnost na kotvu. Odražený signál přijímač zaznamená se zpožděním. V druhém případě již signál neprojde přímou cestou a může se dostat s dostatečným výkonem delší cestou odrazem. Tento případ může zkreslit čas příchozu a tím znemožnit lokalizačnímu systému přesně vypočítat TDoA.





Obrázek 2.8: Vícecestná propagace signálu odrazy od zdí interiéru

## Kapitola 3

# Techniky strojového učení

Strojové učení je oblast z komplexní a stále se aktivně vyvíjející oblasti umělé inteligence. Cílem strojového učení založeného na datech je navrhnout a naučit model, který bude dostatečně přesně popisovat chování systému z reálného světa. Takto vytvořený model lze potom nasadit na řešení problému klasifikace nových dat do tříd nebo predikce hodnot systému v blízké budoucnosti. Od tradičních modelů a algoritmů pro řešení těchto úloh se strojové učení liší ve schopnosti naučit se vzory ve zkoumaném datasetu automatizovaně.

Základem moderních technik strojového učení tvoří tzv. *soft computing*. Společnou vlastností algoritmů z oblasti soft computingu je schopnost tolerovat nepřesnosti, nejistoty a šum, který se vyskytuje v datech pocházejících z fyzického světa. Tyto algoritmy často pracují s principy fuzzy logiky a teorie pravděpodobnosti. V kontrastu s klasickým "hard" výpočtem se soft computing nespolehá na zcela přesné mapování vstupů a výstupů systému, ale spíše na nalezení obecných vzorů.

Tato kapitola se zabývá základními principy a problémy strojového učení a moderními knihovnamí a nástroji. Dále se zabývá kategorií technik učení bez učitele (unsupervised learning). Zde jsou nastíněny konkrétní algoritmy z oblasti shlukové analýzy a modelování Gaussovských rozdělení, které jsou relevantní pro tuto práci. Na závěr kapitoly jsou zmíněny techniky hlubokého učení, které jsou perspektivní pro možný budoucí rozvoj této práce. Kapitola čerpá z knihy *Hands-On Machine Learning with Scikit-Learn and TensorFlow* [6].

### 3.1 Typy strojového učení

Aby bylo možné vybrat správný přístup k řešení problému zpřesnění pozic v lokalizačním systému, je potřeba prozkoumat základní dělení strojového učení a zaměřit se na oblast, kterou bude možné nejefektivněji využít pro datovou sadu RTLS systému. Techniky lze kategorizovat podle těchto vlastností:

- Datová sada obsahuje označené - anglicky *labeled* - data. Label je obvykle přidán lidmi<sup>1</sup> ve fázi předzpracování a dává konkrétnímu objektu v datové sadě význam, třídu nebo cílovou (*target*) hodnotu. Taková datová sada je dobře použitelná pro strojové učení s učitelem (*supervised learning*). Opakem je učení bez učitele (*unsupervised learning*), kdy jsou k dispozici pouze čistá data bez označení a snahou je vzory a třídy odvodit automaticky.

---

<sup>1</sup>Labely může generovat i chytrější stroj, který je nasazen speciálně pro vytváření trénovacích dat.

- Učení probíhá neustále na nově příchozích datech (*online learning*), nebo se model naučí na velkém reprezentativním vzorku (*batch learning*).
- Aplikace funguje na základě porovnávání nových dat s existujícími vzorky, nebo podle naučeného matematického modelu.

Tyto kategorie lze libovolně kombinovat. Celý systém strojového učení se může skládat z více komponent s téměř libovolnou topologií. Mějme původní neoznačený dataset, který se pomocí technik učení bez učitele transformuje na dataset očištěný od šumu. Následně se z nalezených vzorů vytvoří označený dataset, na který se aplikuje některá technika učení s učitelem.

Například datovou sadu zkoumanou v této práci lze považovat za neoznačenou - máme informace o pozicích v čase, které zaznamenal poziční systém za určitých podmínek, ale už nemáme explicitní informaci o kvalitě a přesnosti této pozice. Chtěli bychom pozice roztrždit na takové, které dostatečně přesně odpovídají pozici v reálném prostoru, a na pozice, které vznikly odskokem do vedlejší lokalizační buňky. Ideálně bychom chtěli umět odhadnout věrohodnost pozice a na základě toho ohodnotit pozici podle kvality, například od 0 až 1. K tomuto bude potřeba získat o datasetu znalosti za použití technik učení bez učitele, které zvládnou hodnotit data téměř automatizovaně.

## 3.2 Vlastnosti dat pro strojové učení

Jak již bylo v úvodu kapitoly naznačeno, velká část procesu návrhu systému strojového učení spoléhá na pochopení dat, která máme k dispozici. K prozkoumání obecných vlastností datasetu je potřeba odhalit úskalí, se kterými se běžně projekty založené na strojovém učení setkávají.

### Množství trénovacích dat

Obvykle největší výzvou projektů založených na strojovém učení je získat dostatečné množství dat. I relativně jednoduché modely s malým počtem příznaků potřebují až tisíce kvalitních instancí. Úlohy z oblasti počítačového vidění nebo zpracování přirozeného jazyka potom vyžadují miliony vzorků. Pokud máme štěstí a víme o zkoumané doméně něco více (například fyzikální zákony, které data reflektují), můžeme model učení značně přizpůsobit a poradit si tak i s menším množstvím dat. Také existují různé předtrénované modely, které lze často univerzálně použít pro komplexní problémy a ulevit tak potřebě trénovat model od zнова na obrovské kvantitě dat.

O efektivitě učení modelů na velkých datech pojednává článek *The Unreasonable Effectiveness of Data* [7]. Autoři se zaměřují na výsledky z oblasti zpracování lidského jazyka, kde strojové učení za pomoci obrovských slovníků už dávno zvítězilo nad ručně navrženými algoritmy. Hlavní ideou je, že velká neoznačená a zdánlivě zašuměná data obsahují všechny potřebné komplexní vzory chování, které algoritmy dokáží rozpoznat mnohem rychleji a často i efektivněji, než manuální analýza lidských inženýrů.

### Reprezentativnost

Aby bylo možné naučit model, který dobře generalizuje, je důležité, aby trénovací data reprezentovala všechny možné případy. Pokud máme řešit předpověď chování nějakého objektu v prostoru, ideálně chceme mít dostatek dat ze všech možných situací, které mohou

pro tento objekt přirozeně nastat. Nestačí pouze zaznamenat chování v běžných situacích (takových dat je obvykle dostatek), chceme získat i data z krajních situací. V praxi se můžeme setkat s pojmem *sampling bias*, což je v podstatě zkreslení reality při použití nevhodných metod vzorkování. Ne každý *sampling bias* je ale nutně škodlivý, v některých případech se chceme více zaměřit na extrémní chování systému - sbíráme více dat ze zajímavých a vzácných situací. Podstatné je si při návrhu modelu uvědomit, kde ke zkreslení dochází a podle toho se přizpůsobit.

## Kvalita

Máme-li dostatečné množství reprezentativních dat, stále může naše úsilí narazit na příliš velké množství chybných vzorků nebo na špatně formulované příznaky v datasetu. V závislosti na použité technice strojového učení se model může naučit chybný vzor, který v realitě neexistuje. Například chybný senzor posílá na svůj výstup značně posunuté nebo ořezané hodnoty. Pokud dataset reprezentuje časovou řadu, problematické mohou být i lokálně zpřeházené vzorky (důsledek špatné synchronizace). O technikách čištění dat se dá napsat mnoho stránek, pár stručných postupů vypadá například takto:

- Pokud jsou některé vzorky na první pohled odlehlé (například atribut síly signálu nabývá hodnot, která je fyzicky nereálná), můžeme tento vzorek zahodit.
- Pokud u vzorků chybí hodnoty atributů, můžeme se rozhodnout tento atribut zcela ignorovat, zejména když chybí u velkého množství vzorků. Některé chybějící hodnoty lze doplnit, především u příznaků které částečně korelují s jiným atributem.
- Chybějící hodnoty lze také doplnit průměrnou hodnotou, mediánem a podobně.

## Výběr příznaků

Ve fázi čištění dat se již můžeme začít rozhodovat, které příznaky jsou významné pro řešení a které jsou nerelevantní. Pro úspěšné strojové učení jsou kvalitně vybrané a zkonstruované příznaky kritické. Tomuto kroku zpracování dat se říká *feature engineering* a provádí se v něm následující akce:

- Výběr příznaků: máme data ze senzorů, které tvoří časovou řadu naměřených veličin. Některé veličiny spolu mohou úzce souviset nebo být zcela duplicitní. Takové příznaky můžeme spojit. Naopak informace jako například náhodně generované ID vzorku bude nejspíše zbytečný atribut a je dobré jej vyřadit.
- Extrakce příznaků: pomocí algoritmů redukce dimenzionality je možné data transformovat na menší množinu příznaků, které lépe oddělují vzory a třídy. Tento postup je vhodný především pro komplexní datasety. Nevýhodou je většinou ztráta původního významu atributů - nově extrahované příznaky jsou abstraktní hodnoty. Používaným algoritmem v této oblasti je *Principal Component Analysis* (PCA).
- Tvorba nových příznaků: ve fázi výběru příznaků zjistíme, že bychom chtěli nasbírat dat více a tentokrát s novými atributy, které pomohou lépe modelovat řešený problém.

### 3.3 Učení bez učitele

Mezi nejvýznamnější implementace k řešení těchto úkolů patří metody shlukové analýzy (*clustering*) - K-Means, DBSCAN, OPTICS a další. Fungují za předpokladu, že příznaky instancí stejné třídy nabývají podobných hodnot. Úlohou shlukování je nalézt skupiny sobě podobných instancí, kterým bychom jako lidé přiřadili společný význam, třídu nebo roli. Dále můžeme v datasetu hledat instance, které jsou natolik odlišné od všech ostatních, že bychom je prohlásili za anomálie. Některé shlukovací metody tento jev berou na vědomí a dokáží anomálie detekovat.

Data můžeme modelovat i jako rozložení pravděpodobnosti. Úkolem algoritmů z této oblasti je naučit se funkci hustoty pravděpodobnosti, která nejlépe modeluje proces, ze kterého vzorky pocházejí. Naučený model potom dokáže hodnotit nové instance podle pravděpodobnosti a věrohodnosti. Typickým příkladem těchto algoritmů je směsice Gaussovských rozdělání učených algoritmem *expectation-maximization*.

#### DBSCAN

Algoritmus DBSCAN definuje shluk jako spojitou oblast s vysokou hustotou instancí. DBSCAN je jednoduchý, ale výkonný algoritmus, který je schopen identifikovat libovolný počet shluků jakéhokoliv tvaru, je odolný vůči odlehlým hodnotám a má pouze dva hyperparametry. Učení DBSCAN funguje následovně:

- Pro každou instanci v datasetu spočítá vzorky ve vzdálenosti  $\epsilon$ . Vznikne oblast sousedních instancí.
- Pokud má instance ve svém sousedství alespoň `minPts` jiných instancí, je označen za jádro (*core instance*). Proměnná `minPts` je jedním z hyperparametrů algoritmu.
- Všechny vzorky v blízkosti jádrového vzorku patří do stejného shluku. Může tak vzniknout zřetězení několika jádrových instancí libovolného tvaru.
- Jakákoliv instance, která není jádrem a není blízkosti žádného jádra, je považovaná za anomálii.

DBSCAN funguje výborně pro shluky složitých tvarů, které jsou odděleny oblastmi s nízkou hustotou instancí. Slouží jako dobrý nástroj při prvotní analýze dat nebo detekci anomálií v datasetu. Bohužel ale může selhávat v případech, kdy jsou shluky příliš promíchané. DBSCAN (v kombinaci s klasifikátorem k-nejbližších sousedů) také není snadno použitelný pro měkkou klasifikaci, protože příslušnost do shluků přiřazuje binárně.

#### Směsice Gaussovských rozdělání

Směsice Gaussovských rozdělání (GMM) je pravděpodobnostní generativní model. GMM se řadí do kategorie strojového učení bez učitele a k metodám shlukové analýzy. GMM předpokládá, že modelovaná datová sada byla vygenerována konečnou množinou Gaussovských rozdělání, přičemž nejsou známy parametry těchto rozložení. Typickým ideálním rysem dat generovaných Gaussovským rozdělením je tvar elipsoidy s velkou hustotou instancí uprostřed. Kombinací těchto rozložení je možné modelovat rozložení jednotlivých podtříd analyzovaného problému. Výhodou oproti shlukovacím metodám jako k-means nebo DBSCAN je možnost tvorby poměrně robustních měkkých klasifikátorů založených na vyhodnocení vzorků podle jejich pravděpodobnosti, že spadají do naučeného rozložení.

Algoritmus expectation-maximization (EM) je běžně používaný k učení GMM. Jeho cílem je naučit se na trénovacích datech odhadnout takové parametry směsice rozložení, které maximalizují věrohodnost pravděpodobnostního modelu. Jak již název napovídá, algoritmus pracuje ve dvou krocích, které se iterativně provádějí až do zastavení ukončující podmínkou. Z počátečního náhodně zvoleného nastavení parametrů rozložení algoritmus vždy konverguje k některému lokálnímu optimu funkce věrohodnosti. Toto lokální optimum může být někdy nedostatečné, proto je potřeba algoritmus restartovat několikrát a vybrat model z nejlepší věrohodnosti.

Při učení i aplikaci model pracuje s parametry vícerozměrného Gaussovského rozložení:

- Střed rozložení (anglicky *mean*, v literatuře značeno jako symbol  $\mu$ ) je vektor, jehož velikost odpovídá počtu dimenzí modelovaného prostoru.
- Kovarianční matice (anglicky *covariance*,  $\Sigma$ ) definuje velikost, tvar a sklon elipsoidy.
- Relativní váha (anglicky *weight*,  $\phi$ ) udává relativní váhu komponenty směsice, obvykle v rozsahu 0 až 1.

Techniky modelování a učení GMM jsou vhodné pro modelování relativně jednoduchých procesů, které produkují nízkodimenzionální data. Proto se GMM řadí spíše ke klasickým technikám strojového učení, přesto ale tyto techniky zvládají uspokojivě řešit mnoho problémů z reálného světa.

### 3.4 Lineární regresní model

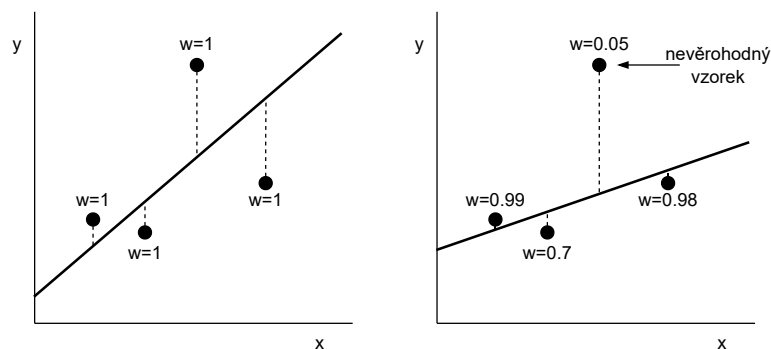
Lineární regrese je klasická statistická metoda pro modelování lineárního vztahu mezi skalární výstupní hodnotou a 1 nebo více hodnotami, které výstupní hodnotu popisují. Model lineární regrese lze použít pro predikci hodnoty podle vztahu:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (3.1)$$

- $\hat{y}$  je odhadovaná hodnota.
- $\hat{n}$  je počet příznaků nebo atributů, které popisují  $\hat{y}$
- $x_i$  je  $i$ -tý příznak nebo atribut a  $\theta_i$  je  $i$ -tý parametr modelu ( $\theta_0$  je speciální parametr tzv. bias)

Existuje několik způsobů učení lineárního regresního modelu - metoda nejmenších čtverců a její vážená varianta je nejčastěji implementovaná v knihovnách pro strojové učení. Cílem učení je nalézt takové parametry, které na daných trénovacích datech minimalizují střední kvadratickou chybu (MSE). Obvykle jsou k dispozici trénovací data, která obsahují šum, ale jako celek vykazují určitý trend, který chceme pochopit. Tento šum může být způsobený přirozeným chováním modelovaného systému, chybou měření a podobně. Pro velmi výrazné anomálie v datech se potom výrazně projeví kvadratická chyba a model se může naučit lineární přímkou nevhodně. Pokud máme o datech nějakou informaci navíc, například jejich věrohodnost, můžeme tuto věrohodnost použít jako váhy při učení metody nejmenších čtverců. Příklad takového modelu ilustruje obrázek 3.1. Datová anomálie zde má nízkou váhu  $w = 0.05$  v porovnání s jinými vzorky, které jsou blízko hodnoty 1.

Lineární regrese je díky své jednoduchosti zajímavým nástrojem pro řešení drobných úloh predikce, kdy je často potřeba generovat rychlý odhad trendu a vyhlazení šumu z časové řady.



Obrázek 3.1: Na levém grafu je ilustrační lineární model naučený metodou nejmenších čtverců, kde všechny vzorky mají stejnou váhu. Na pravém grafu jsou stejné vzorky nyní s váhami, které mají značný vliv na výsledný model.

### 3.5 Hluboké učení

Je-li řeč o moderních technikách strojového učení, nepochybně je důležité zmínit hluboké učení a umělé neuronové sítě. Techniky hlubokého učení (*deep learning*) získávají v posledních letech na popularitě především díky mnoha úspěšným příkladům praktického využití na velmi komplexních problémech z oblasti zpracování obrazu, zvuku, hraní her a dalších. Přirozeně je tedy zajímavé analyzovat, kdy je využití technik z oblasti hlubokého učení opodstatněný přístup k problému, a kdy by pokusy o aplikaci těchto technik vedly spíše do slepé uličky. Hluboké učení má obvykle smysl použít, když jsme schopni zpracovat tyto kritéria:

- Pro trénování hlubokých sítí potřebujeme mít k dispozici velké množství dat. Pokud chceme modelovat i ty nejmenší detaily a nenápadné vzory složitěho systému z reálného světa, je potřeba mít skutečně reprezentativní a vysoce dimenzionální data. Klasickým příkladem je dataset MNIST [5], který tvoří obrázky ručně psaných číslic. Černobílé obrázky jsou čtvercového formátu o hraně 28 pixelů, tedy celkem 784 pixelů. Pro rozpoznávání číslic je toto dostatečně velké rozlišení, ale pro rozpoznávání složitějších objektů je většinou i 784 pixelů málo.
- Lze pro problém použít a odladit předtrénovaný model hlubokého učení. Namísto učení složitěho hlubokého modelu na zcela novém problému, použijeme již naučený model na podobném problému jako odrazový můstek. Pokud je úkolem sestavit klasifikátor objektů na obrázku, pravděpodobně již bude možné přizpůsobit jeden z mnoha předtrénovaných modelů. Předtrénované modely lze použít pro extrakci příznaků, ladění jednotlivých vrstev na potřeby cílového problému a další.
- K trénování a někdy i aplikaci modelů je potřeba velký výpočetní výkon. Samotná inference modelu může být natolik výpočetně drahá, že nebude model možné prakticky použít. To je obvykle velký problém při snaze o využití strojového učení u aplikací, které zpracovávají data v reálném čase. U těchto aplikací je nutnost provést klasifikaci nebo predikci ve zlomku sekundy.

## 3.6 Nástroje a frameworky

Moderním trendem v oblasti strojového učení je řešený problém studovat a experimentálně implementovat v jazyce Python za použití populárních frameworků, které pomáhají urychlit a sjednotit výzkumnou práci.

Pro průzkum, vizualizaci a analýzu dat je oblíbená knihovna pandas [1]. Pandas pracuje s daty jako s tabulkami *DataFrame*. Tabulky lze načítat z běžných souborových formátů jako csv nebo json. S tabulkami lze potom provádět operace jako selekce řádků, filtrování podle sloupců, agregace a podobně. Pandas podporuje vizualizaci dat pomocí knihovny matplotlib.

Pro klasické strojové učení je vhodná knihovna Scikit-learn [12]. Je to open-source framework, který se zaměřuje na metody z oblasti klasifikace, predikce, regrese, shlukování, redukce počtu dimenzí a další. Framework je ideální pro použití na jednoduchých datových sadách. Pro hluboké učení existují moderní frameworky TensorFlow a Keras. Jsou vhodné především na problémy dobře řešitelné hlubokými neuronovými sítěmi a dalšími hlubokými modely. Jak bude zmíněno v dalších kapitolách, povaha řešeného problému v této práci přiklání k použití jednodušších modelů, na které stačí framework Scikit-learn. Použití hlubokého učení bude směr pro možný budoucí rozvoj práce.



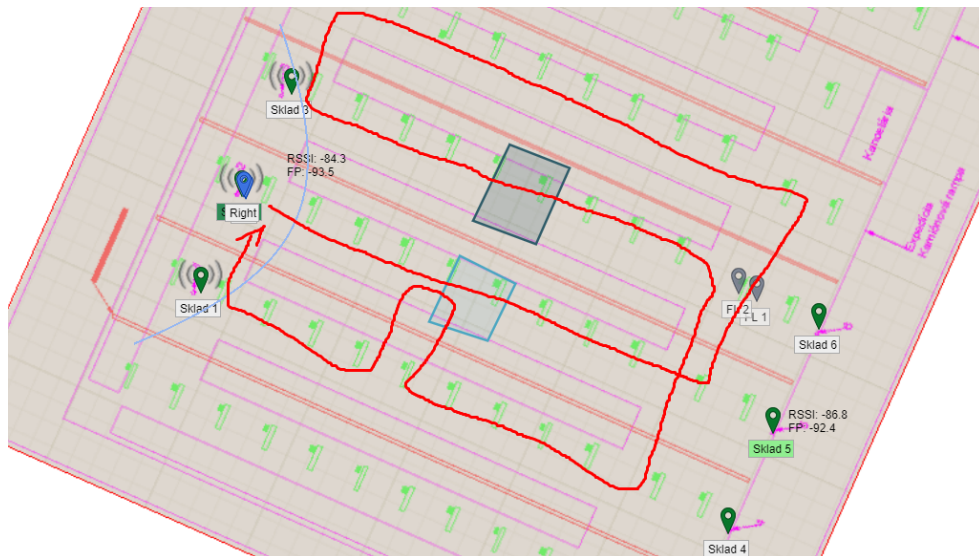
# Kapitola 4

## Analýza lokalizačních dat

Tato kapitola se zabývá daty získanými z reálné RTLS Sewio instalace a analýzou chování signálových parametrů. V prvních sekci kapitoly je popsána struktura datasetu se vzorky z průběhu lokalizace objektů. V další sekci jsou data statisticky analyzované pro potřeby metod strojového učení.

### 4.1 Analýza datasetu

Poziční data pro zpracování pocházejí z reálného skladového prostoru. Plán prostoru a instalace kotev je na obrázku 4.1. Je zde 6 UWB kotev umístěných na koncích skladových uliček. Lokalizační buňky jsou nastaveny jako 1D buňky (popis v kapitole 2.4). Červenou barvou je načrtnut pohyb monitorovaného objektu, ke kterému jsou připevněny 2 tagy. Lokalizační systém počítá pozice každého tagu nezávisle – tato vlastnost datasetu bude užitečná později ve fázi přípravy příznaků pro strojové učení. Data z redundantního tagu umožní lépe označkovat správné a špatné datové rámce.



Obrázek 4.1: Plán interiéru a náčrt pohybu lokalizovaného objektu

## Struktura datové sady

Datová sada se skládá z množiny záznamů vygenerovaných RTLS platformou. Tabulka 4.1 popisuje strukturu jednoho záznamu, což je v podstatě záznam jednoho výsledku pozicování systému. Záznamy jsou v datové sadě seřazeny chronologicky podle časového razítka vytvoření záznamu.

Každý záznam patří právě jednomu tagu a je výsledkem pozicování právě jedné sady kotev – v případě tohoto datasetu jsou to vždy dvojce kotev master-slave v instalaci 1D lokalizační buňky. Identifikátor jednoznačně určuje UWB zařízení, které se v systému nachází, takže je možné datasety spojovat bez větších obtíží s čištěním duplicitních a konfliktních rámců.

K dispozici jsou signálové parametry RSSI, FP, FPI a CIR a jsou vždy vztažené ke každé kotvě zvlášť. Například `master-RSSI` je naměřená hodnota RSSI master kotvy a podobně. Charakteristiky synchronizace jsou označeny prefixem `sync` a udávají stav poslední proběhlé synchronizace kotev s master kotvou. Atribut `Time` je Unixový čas v nanosekundách a reprezentuje časový okamžik vytvoření záznamu – u tohoto atributu je potřeba myslet na velikost datového typu a při implementaci použít alespoň 64 bitový integer. Prostorové souřadnice jsou dány odpovídajícími atributy X, Y a Z. V každé instalaci je platformou definovaný počáteční bod a souřadnice pozicovaného objektu jsou posunutí od tohoto středu.

Atribut	Datový typ	Rozsah	Popis
Tag	String	12 znaků	Identifikátor tagu
Index	Integer	0 až N-1	Index záznamu v rámci (N = počet buňek)
Master	String	12 znaků	Identifikátor master kotvy
Slave	String	12 znaků	Identifikátor slave kotvy
RSSI	Float	-115 až -80	Indikátor síly signálu přijaté blink zprávy
FP	Float	-115 až -80	Indikátor síly první cesty signálu
FPI	Float	0 až 1023	First path index, indikuje kolize impulsů tagu
CIR	Float	–	Poměr užitečného signálu a rušení
Sync-RSSI	Float	-100 až -80	RSSI poslední synchronizace kotev
Sync-FP	Float	-100 až -80	FP poslední synchronizace kotev
Aisle-part	Float	0 a více	Vzdálenost k nejbližší kotvě v metrech
X	Float	–	Relativní pozice X od středu prostoru
Y	Float	–	Relativní pozice Y od středu prostoru
Z	Float	–	Vzdálenost od země
Time	Integer	–	Unixový čas v nanosekundách

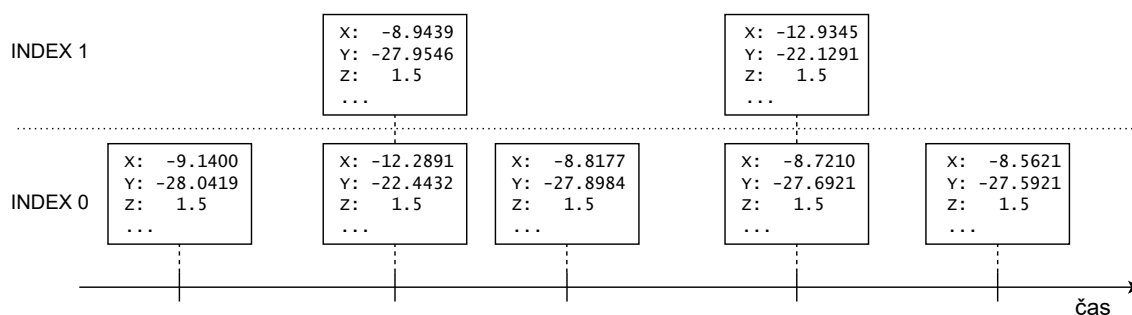
Tabulka 4.1: Struktura záznamu datasetu

## Posloupnost rámců

Záznamy jsou sice v datasetu seřazeny chronologicky, ale časové mezery mezi nimi jsou různé. Tyto výkyvy způsobují síťové latence a různé proměnlivé jevy při výpočtu pozice. V průměru jsou časové rozdíly v řádech 100 milisekund, tedy každou sekundu lokalizační buňka vygeneruje přibližně 10 rámců. Delší prodlevy začnou nastávat v momentě, kdy je lokalizace na hranici schopností získávat dostatečně kvalitní pozici nebo ztratí signál od tagu úplně. V prvním případě kotvy možná blink zprávu tagu obdrží, ale charakteristiky signálu nesplní některý minimální práh na straně lokalizačního serveru (například utlumený impulz kvůli NLOS propagaci, nebo byl zjištěn FPI pod minimální přípustnou hodnotu).

V nejhorším případě je tag i mimo dosah samotných kotev. Při získávání tohoto datasetu byl monitorovaný objekt poslán do uličky, která není zmapovaná žádnou lokalizační buňkou. Podle očekávání tedy z tohoto časového úseku nejsou žádná poziční data.

Pokud záznamy o pozici jednoho tagu promítneme na časovou osu, kde časem je časové razítko vytvoření rámce, tak uvidíme, že v některých časových okamžicích máme více záznamů o pozici z více lokalizačních buněk 4.2. Na obrázku je vzorek z datasetu, kdy lokalizace vedla v některých časových rámcích na 2 různé pozice. Z uvedených 5 časových rámců jsou dva takové, kde lokalizační systém vypočítal pozici pomocí dvou různých lokalizačních buněk. Index záznamu potom řadí záznamy podle kvality signálu. Je tedy pravděpodobnější, že data s indexem 0 budou obsahovat korektnější pozici než data s vyšším indexem. Každopádně na tomto vzorku je vidět, že současný systém může chybovat – indexy druhého časového rámce jsou prohozené. Naučit se vybrat věrohodnější pozici v těchto situacích bude jedním z úkolů klasifikátoru.



Obrázek 4.2: Vzorek dat na časové ose

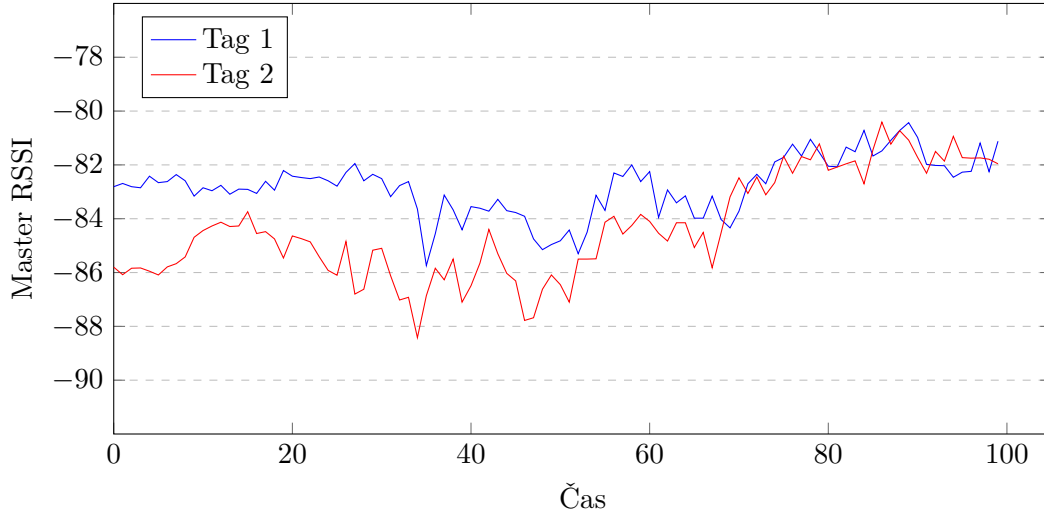
Dále může nastat situace, kdy máme v určitém okamžiku pouze 1 záznam a tento jediný záznam obsahuje nepřesná data.

- Reálný monitorovaný objekt je zaznamenán ve správné uličce nebo lokalizované buňce, ale vypočtená pozice neodpovídá realitě. V tomto případě dojde k odskoku směrem dopředu nebo dozadu v 1D lokalizaci. Data z takového záznamu mohou být stále užitečná pro prediktor.
- Objekt je zaznamenán pouze ve špatné uličce, obvykle v jedné ze sousedních paralelních uličkách. V tomto případě pozice odskočí mimo správnou lokalizační buňku. Záznam potom obsahuje zcela špatnou pozici a signálové charakteristiky od jiné dvojice kotev.

#### 4.1.1 Analýza signálových parametrů

Data obsahují naměřené signálové parametry od 2 různých tagů připevněných na 1 pohybujícím se objektu. Hypoteticky by tedy měly v přibližně stejných časových úsecích vykazovat podobné charakteristiky. Na obrázku 4.3 je graf průběhu atributu RSSI při průchodu objektu prostřední uličkou. Je zde vidět podobnost ve změnách hodnot. Podobně se chovají i jiné atributy, například indikátor síly signálu FP. Sledování sekvence signálových atributů bude významný příznak při optimalizaci klasifikátoru správných pozičních dat. Velké odchylky v časovém kontextu od předpokládaného chování mohou být indikátorem chybující lokalizace.

K neurčitosti pozice často dochází, když se tag nachází uprostřed uličky. V této části prostoru je velká pravděpodobnost, že vznikne přímá viditelnost na kotvy ve vedlejší uličce mezerami v regálech. RTLS má v těchto situacích zjevně problém odlišit správnou pozici pouze na základě aktuálních indikátorů, protože atributy nabývají podobných hodnot.



Obrázek 4.3: Chování parametru master RSSI dvou sousedních tagů při průchodu uličkou

Parametr CIR bude z dalšího zkoumání vyřazen, protože u drtivé většiny záznamů nabývá prázdné nebo nulové hodnoty. Stejně tak parametry synchronizace kotvy nejsou příliš zajímavé, protože UWB zařízení ze studovaného prostředí jsou vždy dobře synchronizovaná. Vliv synchronizace tedy nebude předmětem další analýzy.

### Korelace parametrů

Tabulka 4.2 shrnuje vzájemnou míru korelace parametrů formou korelační matice. Největší míry korelace kolem hodnoty 0.8 dosahují parametry RSSI a FP u master i slave kotvy. Tato korelace není příliš překvapivá, protože oba parametry popisují sílu přijatého signálu. Liší se jen v definici způsobu přijetí.

	Master			Slave		
	RSSI	FP	FPI	RSSI	FP	FPI
Master-RSSI	1.0000	0.8110	0.2995	0.3763	0.4422	0.1010
Master-FP	0.8110	1.0000	0.5329	0.3680	0.4502	0.0904
Master-FPI	0.2995	0.5329	1.0000	0.0750	0.1600	0.0186
Slave-RSSI	0.3763	0.3680	0.0750	1.0000	0.8709	0.1767
Slave-FP	0.4422	0.4502	0.1600	0.8709	1.0000	0.2769
Slave-FPI	0.1010	0.0904	0.0186	0.1767	0.2769	1.0000

Tabulka 4.2: Korelační matice signálových parametrů v neoznačeném datasetu

#### 4.1.2 Analýza označeného vzorku

Z původního neoznačeného datasetu bylo náhodně vzorkováno 100 pozic. Tyto pozice byly následně ručně klasifikovány do dvou tříd podle správnosti pozice vzhledem k realitě. Pozice,

které nepochybně padly do správné uličky, byly označeny za správnou a mají atribut *label* s hodnotou 0. Pochybné a zjevně odskočené pozice jsou jinak zařazeny do třídy 1. Označený vzorek bude sloužit pro vizualizaci, asistenci u datové analýzy a při experimentech s klasifikátorem.

### Korelace parametrů

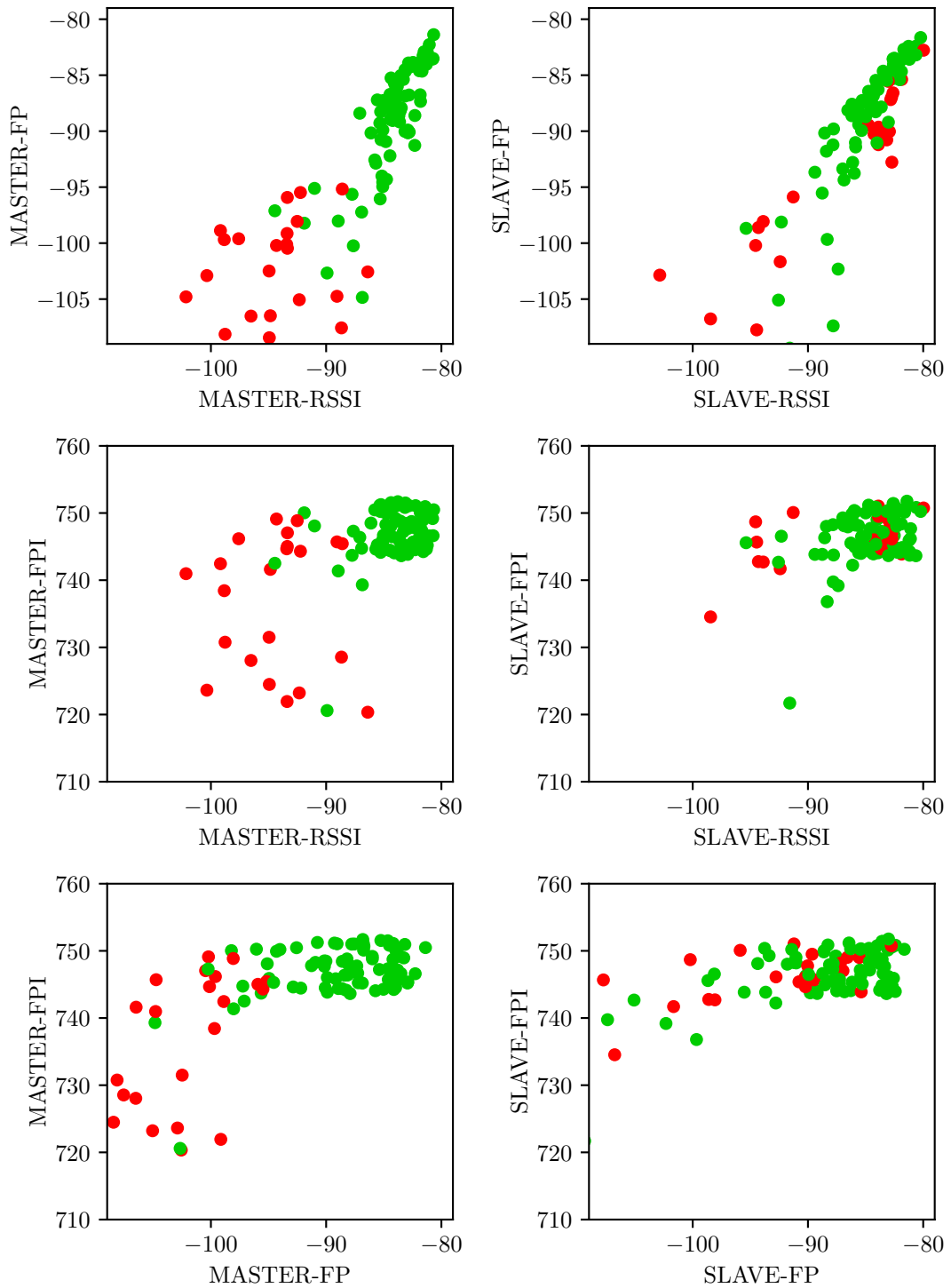
Podobně jako u neoznačeného datasetu je přínosné se podívat na korelaci, nyní především na shodu ruční klasifikace s jednotlivými parametry. Tabulka 4.3 obsahuje výňatek z korelační matice. Label nejlépe koreluje s parametry master kotvy. U slave parametrů je korelace o dost horší, ale stále není nulová. Tento rozdíl v hodnotách korelací master a slave parametrů může být spíše důsledkem pohybu lokalizovaného objektu blízko slave kotev. Viditelnost master kotev na opačnou stranu uliček byla značně horší, a proto i špatná síla signálu dobře souvisí s přesností pozice. Je slušná šance, že především parametry RSSI a FP obou typů kotev budou dobře sloužit při modelování a učení klasifikátoru.

	Master			Slave			Label
	RSSI	FP	FPI	RSSI	FP	FPI	
Label	-0.8239	-0.7547	-0.4932	-0.2908	-0.3244	0.1479	1.0000

Tabulka 4.3: Korelace ruční klasifikace se signálovými parametry

Obrázek 4.4 ilustruje různé pohledy na dimenze signálových parametrů. Třída správných pozic je označena zeleně, jinak červeně. První dva grafy zachycují silnou korelaci parametrů RSSI a FP, především pro třídu správných pozic blízko hodnot  $-80$ . Správné pozice mají celkově nízký rozptyl hodnot všech parametrů. Špatné pozice jsou podstatně více rozptýlené a naneštěstí pro data ze slave kotev se prolínají s množinou správných pozic.

Vzhledem k těmto zjištěním bude vhodné klasifikaci provádět na kombinaci master a slave RSSI parametrů, protože nejlépe oddělují třídy, které bude klasifikátor rozpoznávat.



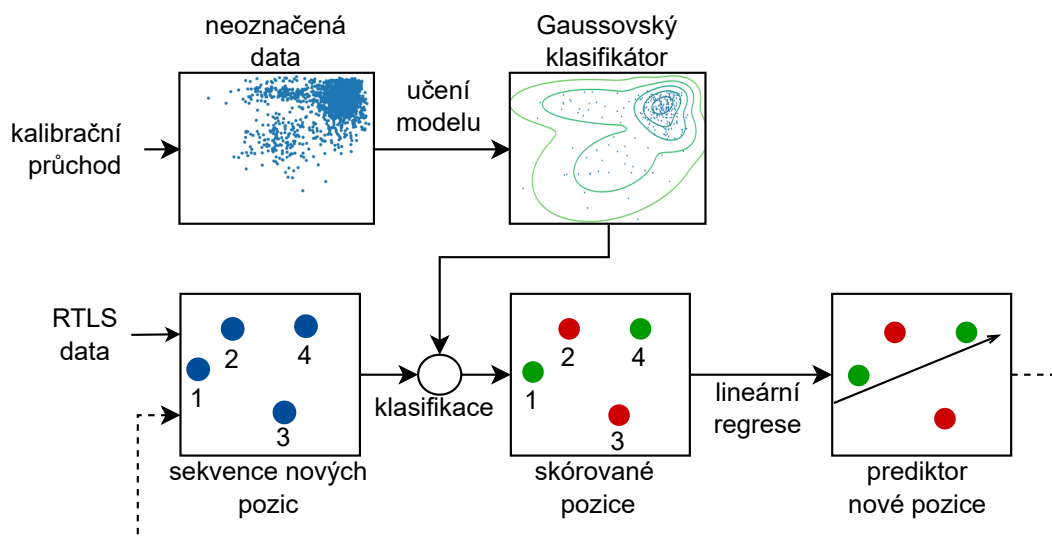
Obrázek 4.4: Kombinace parametrů RSSI, FP, FPI pro master a slave

# Kapitola 5

## Implementace řešení

Navržený systém strojového učení se skládá ze 2 hlavních komponent. První komponentou je klasifikátor datových rámců, které jsou popsány v kapitole 4. Cílem tohoto klasifikátoru je naučit se distribuci dat podle signálových parametrů a zvládnout ohodnotit nové pozice měkkým skóre, které reprezentuje kvalitu či signálovou stabilitu vypočtené pozice. Druhou komponentou je prediktor pozice pomocí lineární regrese. Úlohou prediktoru je v reálném čase odhadovat pravděpodobnou trajektorii na základě klasifikátorem ohodnocených pozic. Systém je navržen jako tzv. *downstream* proces, úlohou je přebírat a zpracovávat datový tok stávající lokalizace a z těchto dat odhadovat nové pozice.

Obrázek 5.1 konceptuálně ilustruje navržený systém. V první fázi se při tvorbě nové lokalizační instalace provede kalibrační průchod prostředím. Vznikne tak vzorek dat, který by měl dostatečně dobře reprezentovat signálové charakteristiky v jednotlivých lokalizačních buňkách. Na těchto datech se naučí a odladí model směsice Gaussovských rozdělání. Tento model potom bude použitý jako klasifikátor správných pozic. Nové pozice vypočtené stávajícím RTLS systémem jsou dále klasifikátorem ohodnocené měkkým skóre, které se použije jako váha u vážené lineární regrese. Regrese následně na krátké sekvenci historických pozic odhadne trajektorii a současnou pozici.

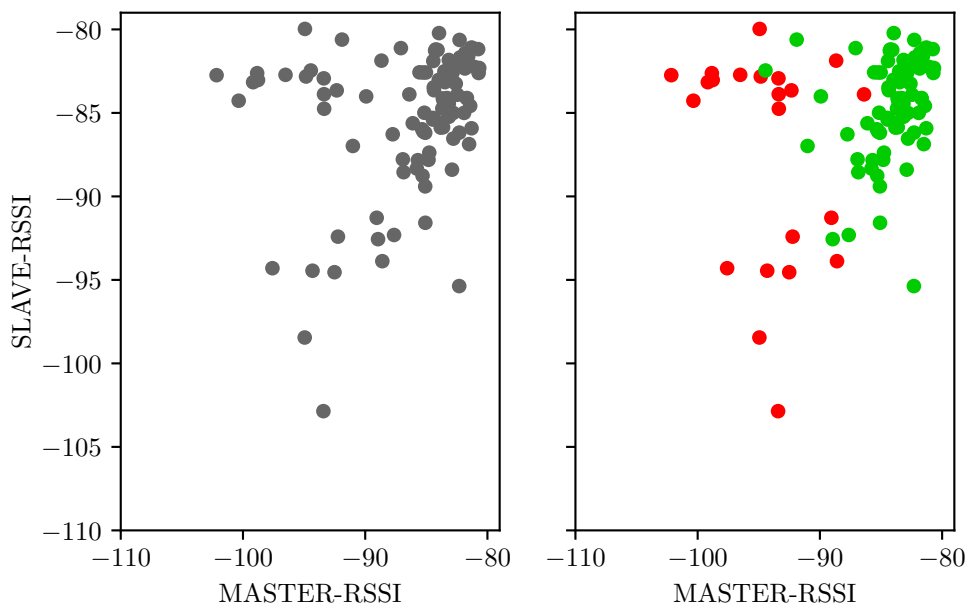


Obrázek 5.1: Koncept systému pro zvýšení robustnosti lokalizace

## 5.1 Klasifikace Gaussovským modelem

Při volbě klasifikátoru bylo potřeba vzít v úvahu poměrně malé množství neoznačených dat. Klasifikátor musí mít schopnost data ohodnotit měkkým skóre, které značí míru stability vypočtené pozice. Data mají naštěstí dobrou vlastnost, kdy relativně přesně vypočtené pozice tvoří shluky. Méně kvalitní a odskočené pozice jsou potom obvykle vzdálenější v prostoru signálových parametrů. Pro jednoduchost vizualizace budou modely prezentované na dvourozměrných datech (signálové parametry master a slave RSSI).

Na obrázku 5.2 je ukázka malého vzorku dat ručně roztříděných do 2 tříd. Vzorky označené zeleně jsou pozice, které byly lidským úsudkem označeny za přesné. Červeně jsou naopak pozice, které odskočily do vedlejší uličky a jsou to tedy chybné pozice. Intuitivně tedy platí, že pozice s příznaky dobrých signálových charakteristik RSSI blízko hodnoty -80 mají velmi vysokou pravděpodobnost úspěšné lokalizace.



Obrázek 5.2: Porovnání vzorku neoznačených a označených dat podle kvality

Tato zjištění lákají data modelovat pomocí technik založených na odhadu pravděpodobnostních rozložení, například směsí Gaussovských rozdělání (GMM). Framework Scikit-learn obsahuje implementaci GaussianMixture v modulu mixtures. GaussianMixture zapouzdřuje metody pro trénování a evaluaci těchto modelů pomocí metod `fit` a `predict`. Výstupem trénování je střed rozložení a kovarianční matice, která definuje velikost a sklon rozložení v dimenzích signálových parametrů.

Druhým algoritmem z rodiny Gaussovských rozdělání je rozšířená varianta Bayesian Gaussian Mixture. V porovnání s klasickým GMM se zvládne lépe přizpůsobit shlukům v trénovacích datech, které nemají podobu běžného normálního rozložení. Scikit-learn obsahuje implementaci s názvem BayesianGaussianMixture s podobným rozhraním jako GMM implementace.



### 5.1.1 Hyperparametry

Metody při trénování vyžadují nastavení několika hyperparametrů, které je potřeba experimentálně přizpůsobit:

- `n_components` je počet Gaussovských rozložení mapovaných na data,
- `covariance_type` je typ kovariančních matic, které definují tvar a velikost rozložení,
- `weight_concentration_prior` je váhový parametr koncentrace středu rozdělení. Vyšší hodnoty způsobí, že těžiště modelu má větší váhu a tím aktivuje více komponent, nízké hodnoty vedou k rovnoměrnějšímu rozprostření. Dobrou výchozí hodnotou je  $1 / n\_components$ . S tímto parametrem pracuje pouze Bayesovská varianta GMM.

#### Výběr počtu komponent rozložení

Výzvou při tvorbě modelů typu GMM je správná volba počtu komponent Gaussovského rozdělení. Pokud je zvolena jedna komponenta pro celý trénovací dataset, algoritmus při učení zahrne i zjevně chybné pozice, zejména pokud je v datech chybných vzorků poměrně velké množství. Rozložení tak nebude správně mapovat menší shluky vzorků, které mohou být pro klasifikaci nových vzorků zásadní. Při volbě 2 komponent se rozložení podstatně lépe rozdělí na shluk kvalitních a špatných vzorků. Po natrénování modelu se 2 komponentami tak stačí vybrat tu komponentu, která reprezentuje třídu kvalitních vzorků. Tímto postupem lze experimentálně přidávat více složek rozdělení, dokud algoritmus učení konverguje.

Pro nalezení vhodného počtu složek směsice lze použít analytické techniky, které ohodnotí přesnost statistického modelu. Implementace GMM v knihovně Scikit-learn obsahuje techniky založené na kritériích AIC (Akaike information criterion) a BIC (Bayesian information criterion). Na obrázku 5.3 jsou vypočtené obě metriky pro variabilní počet komponent rozložení. Optimální počet komponent je globální minimum těchto metrik. Zatímco křivka pro AIC se zplošťuje, BIC doporučuje počet složek kolem sedmi. Přesná optimální hodnota bude vždy záležet na datech, které vyprodukuje konkrétní instalace, lokalizační buňka nebo lokalizované zařízení.

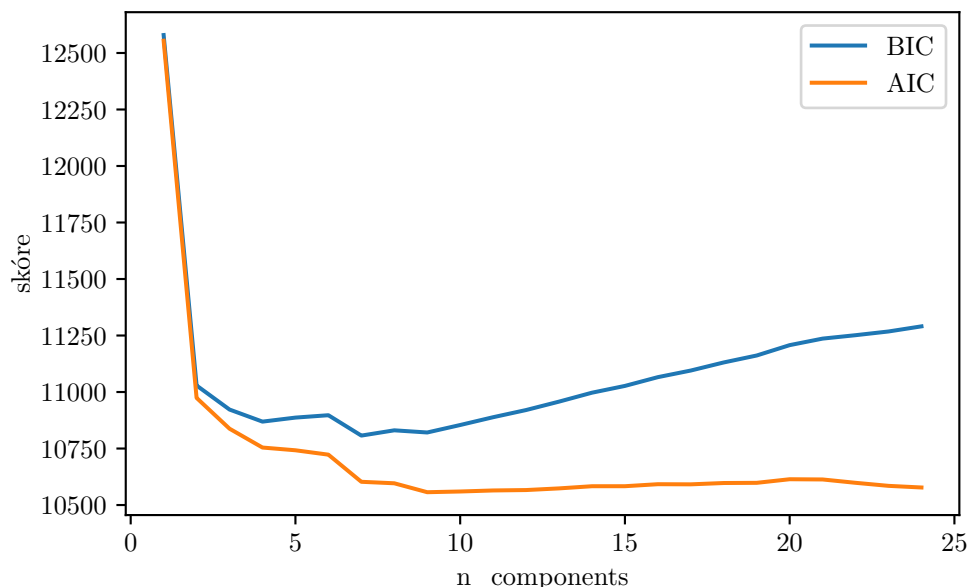
#### Výběr typu kovarianční matice

Typ kovarianční matice dává algoritmu větší či menší volnost přizpůsobit se lépe tvaru trénovací sady. Matice typu `spherical` je nejstriktnější a dovoluje rozložení pouze kruhového tvaru pro všechny komponenty. Typ `diag` je diagonální matice, která umožňuje tvar rozložení roztáhnout v dimenzích různě podle potřeby. Typ `full` potom umožňuje popsat každou složku svou unikátní maticí, s možností rozložení definovat jako libovolně natočenou elipsu. S mírou volnosti obecně roste výpočetní náročnost algoritmu, každopádně u tohoto jednoduchého datasetu je možné data modelovat typem `full`.

### 5.1.2 Analýza modelů

V této sekci budou zkoumány pohledy na natrénované směsice rozložení s různým nastavením hyperparametrů.

Obrázek 5.4 ilustruje vrstevnice dvourozměrné směsice, která byla natrénovaná pomocí pouhých dvou komponent. Největší hustotu má komponenta v oblasti dobrých signálových



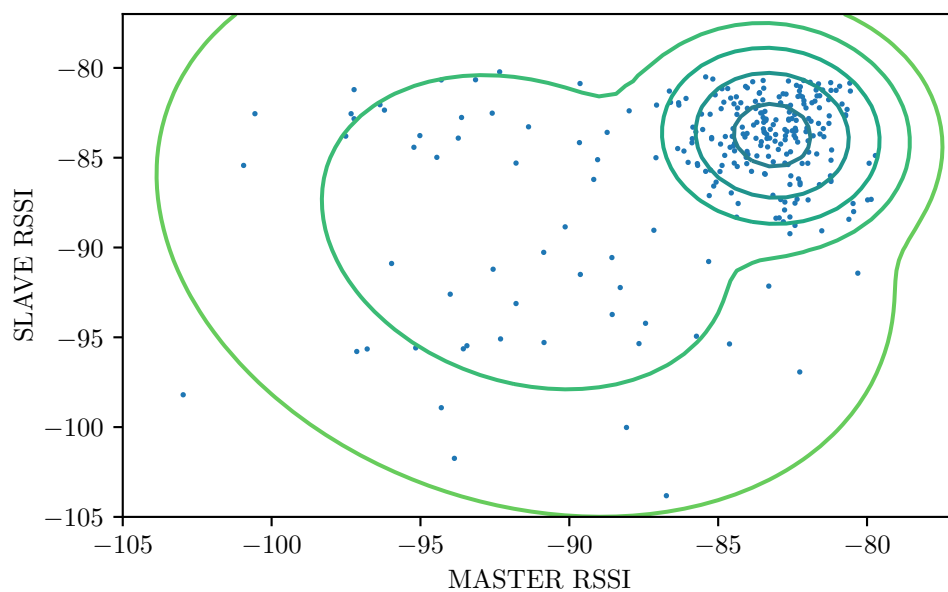
Obrázek 5.3: Metriky BIC a AIC pro různé počty komponent rozložení

parametrů se středem  $(-83.1, -83.7)$  – tuto komponentu rozložení lze pravděpodobně použít pro dobře generalizující klasifikaci správných pozic, pokud nepotřebujeme detailnější a více specifický model. Druhá komponenta se středem  $(-91.4, -89.1)$  je tvarově podstatně širší a řidší. Snaží se zachytit množinu vzorků s horšími signálovými parametry, které bychom klasifikovali jako nepřesné pozice.

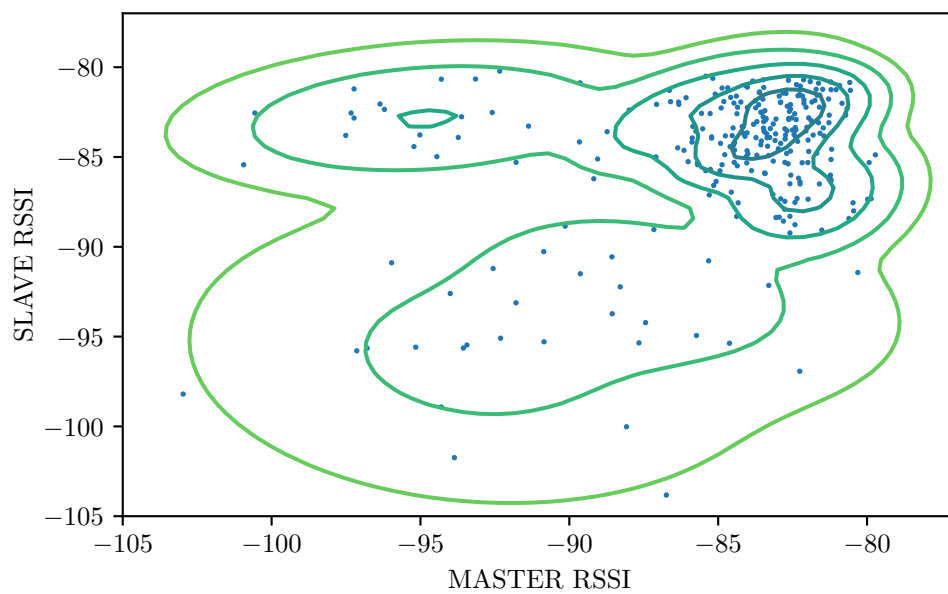
Obrázek 5.5 ilustruje podstatně detailnější rozložení. Směsice obsahuje 7 komponent, které určilo kritérium BIC jako optimální. U tohoto modelu si můžeme zvolit klasifikaci správných pozic pomocí více komponent, které tvoří shluk v pravém horním rohu. Model dokáže i lépe odlišit různé podtřídy chybných pozic. Komponenta se středem  $(-94.8, -82.8)$  popisuje chyby v lokalizaci, které často vznikají na koncích uliček. Komponenta se středem  $(-92.9, -96.0)$  je třída chybných vzorků, kterými jsou obvykle odskoky uprostřed uličky.

Na obrázku 5.6 je nyní model Bayesovské varianty směsice rozložení (BGMM). BGMM se dokáže vyrovnat i s nepřesně určeným počtem komponent a do jisté míry zabránit možnému přeučení. Model byl naučen s inicializací `n_components=12`, učící algoritmus ale nakonec přiřadil sedmi komponentám nulovou váhu. BGMM tak odhalil dva řídké shluky chybných vzorků, podobných jako u předchozího modelu GMM se 7 složkami. Shluk blízký kvalitním vzorkům BGMM modeluje pomocí tří komponent s podobnými středy a kovariančními maticemi.

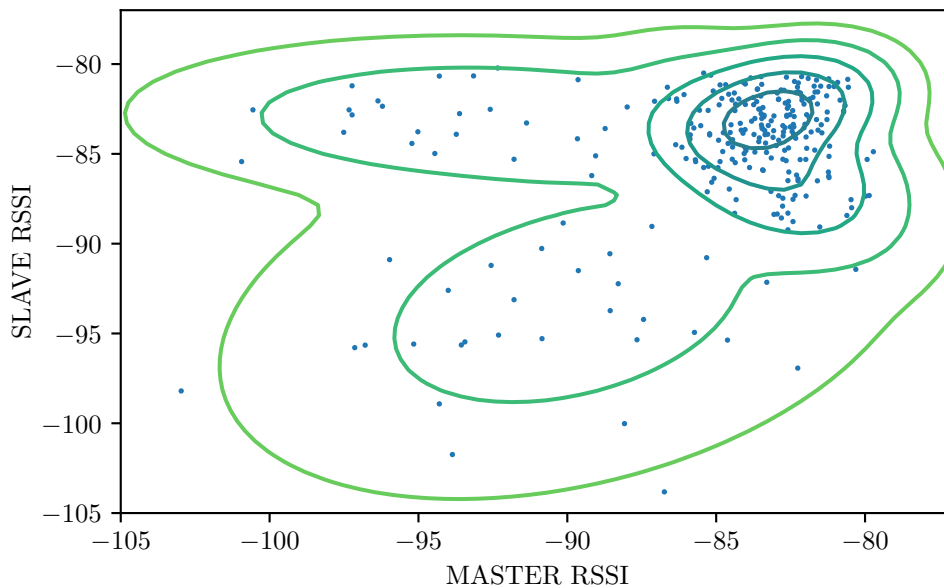
Pro vytvoření klasifikátoru nyní stačí vybrat podmnožinu Gaussovských komponent, které náležejí shlukům kvalitních pozic. Z prezentovaného BGMM modelu byly vybrány 2 komponenty se středy  $(-83.0, -82.8)$  a  $(-83.3, -85.4)$  a váhami 0.488, 0.292. Testovací data jsou vyhodnocena dávkově metodou `predict_proba`. Výstupem je matice, kde sloupce jsou pravděpodobnosti náležitosti vzorků k jednotlivým komponentám směsice. Jako finální skóre je vybráno maximum z těchto pravděpodobností.



Obrázek 5.4: Kontury směsice Gaussovských rozdění se 2 složkami



Obrázek 5.5: Kontury směsice Gaussovských rozdění se 7 složkami



Obrázek 5.6: Kontury směsice Bayesovské varianty

Pro vyhodnocení byl použit testovací ručně označený vzorek z obrázku 5.2. Nyní jsou vzorky klasifikovány na obrázku 5.7, a to tvrdým (levý graf) a měkkým (pravý graf) skóre. Práh tvrdého skóre je nastaven na  $p > 0.5$ , pokud se jedná o správnou pozici (zeleně), jinak špatná pozice (červeně). Měkké skóre je ilustrováno jako sytě zelená pro vzorky s vysokou pravděpodobností, že patří do třídy kvalitních vzorků.

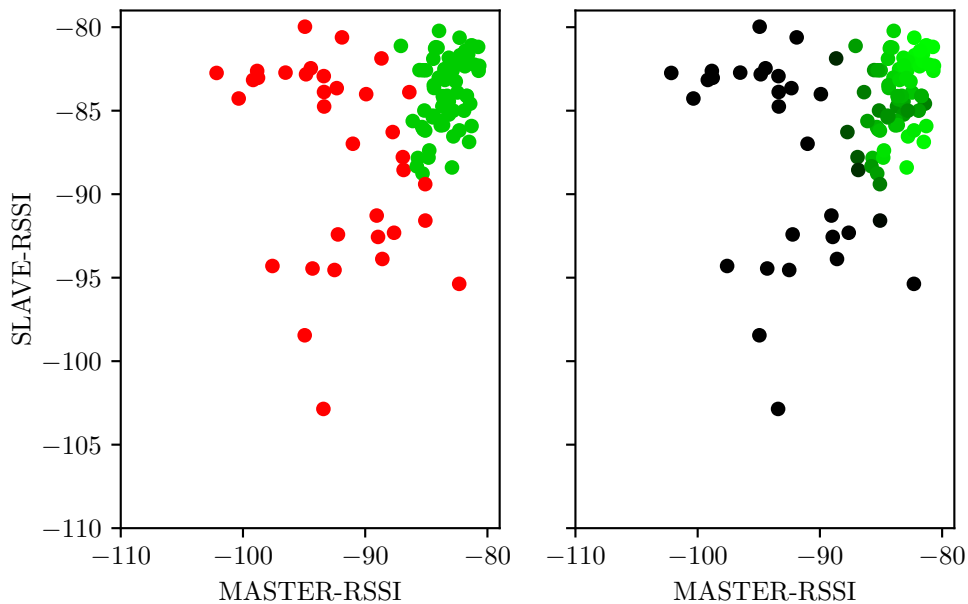
Na testovacím vzorku o velikosti 100 bylo

- 66 vzorků klasifikováno správně jako kvalitní pozice (skutečně pozitivní),
- 22 vzorků klasifikováno správně jako špatná pozice (skutečně negativní),
- 0 vzorků klasifikováno nesprávně jako kvalitní pozice (falešně pozitivní),
- 12 vzorků klasifikováno nesprávně jako špatná pozice (falešně negativní).

Klasifikátor je tedy v režimu tvrdého skórování spíše přísnější, protože zahazuje i správně vypočtené pozice. Pro další část systému se bude klasifikátor používat v režimu měkkého skóre. Toto skóre bude sloužit jako váha u regresního modelu, takže přísnost vzhledem k třídě správných vzorků není nutně na škodu.

### Zdrojové kódy

Grafy a modely v této části kapitoly vznikly v jazyce Python za použití knihoven numpy, pandas, Scikit-learn a matplotlib. Zdrojové kódy k obrázkům jsou k nalezení ve složce `src/plots/` v souborech `plot_example.py` a `plot_gaussian_*.py`. Skript pro tvorbu a uložení Gaussovského modelu je k nalezení v `src/models/cluster.py`. Tento skript také provádí učení Gaussovského modelu pomocí GMM. Prvním vstupním argumentem skriptu je cesta k JSON souboru, který obsahuje trénovací data. Druhým argumentem je



Obrázek 5.7: Tvrdá a měkká klasifikace testovacího vzorku

cesta výstupního souboru, do kterého se uloží GMM objekt, aby mohl být dále použit při vyhodnocení.

V adresáři `data` jsou připraveny příklady 2 naučených modelů BGMM, které budou použité při vyhodnocení:

- `model_1_2f.sav` - Gaussovský model využívající parametry slave a master RSSI,
- `model_1_4f.sav` - Gaussovský model využívající navíc parametry FP.

## 5.2 Predikce trajektorie

Prediktor trajektorie lokalizovaného objektu je postavený na váhované lineární regresi pozic  $(X, Y)$  z časového intervalu  $I$  sekund. Tento interval je implementovaný jako klouzavé okno. Idea tohoto přístupu je taková, že se lokalizovaný objekt pohybuje v prostoru podle zákonů fyziky. Pokud se objekt pohybuje nějakou rychlostí v čase  $T - 1$  sekund, bude se pravděpodobně pohybovat podobnou rychlostí v čase  $T$  sekund. Z nasbíraných dat o pohybu objektu za posledních několik sekund lze tedy provést slušný odhad opravdové pozice. Nepřesné a odskočené pozice budou mít díky předchozí klasifikaci poměrně nízkou váhu, takže se regrese bude spíše přiklánět vzorkům s vysokou vahou. Finální odhad pozice tak lépe odfiltruje odskoky a konfliktně vypočtené pozice.

### 5.2.1 Hyperparametry

Při ladění prediktoru je možné experimentovat s těmito hyperparametry:

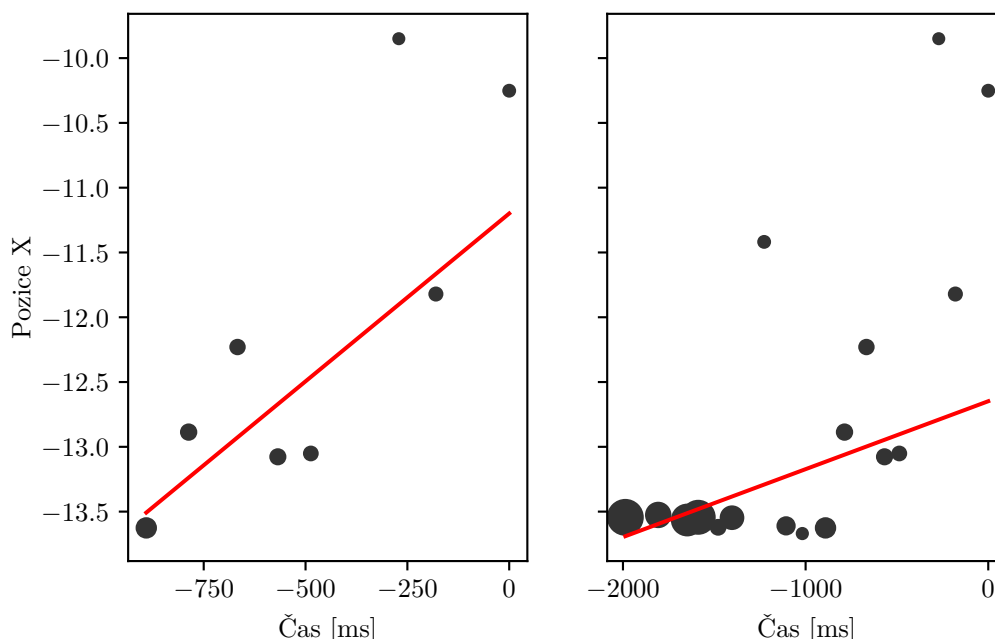
- Velikost klouzavého okna v sekundách: regrese přizpůsobí trajektorii vzorkům, které se vlezou do tohoto intervalu.

- Minimální počet vzorků v okně: umožňuje vypnout predikci, pokud není k dispozici dostatek dat z blízké minulosti.
- Minimální průměrné skóre pozic v okně: pokud má regrese dostatek dat, ale jejich průměrná váha je nízká, predikce se taktéž neprovede.

### Velikost okna

Klouzavé okno (anglicky *sliding window*) je technika běžně používaná u zpracování časových řad. Okno reprezentuje historii několika pozic, na kterých se regresní algoritmus naučí trajektorii. Velikost okna má vliv nejen na výsledné predikce, ale i na výpočetní čas. Tyto dvě kritéria často jdou proti sobě. Cílem je najít vhodný kompromis, který bude server stíhat zpracovávat data v reálném čase a zároveň bude predikce uspokojivě kvalitní.

Bodové grafy na obrázku 5.8 porovnávají regresi pro okna s velikostmi 1 a 2 sekundy. Velikost bodu reprezentuje váhu pozice. Odhadovaná souřadnice je na grafu pouze X, ale analogicky funguje i pro souřadnici Y. Je zde zachycený moment, kdy lokalizovaný objekt začal procházet ze spodní uličky do prostřední. Původní vypočtené pozice jsou v tuto chvíli velmi nestabilní, přesto ale správně naznačují hrubý směr pohybu. U krátkého okna začne regrese rychleji reagovat na odskoky a nemusí tedy dostatečně dobře dráhu pohybu vyhladit. Pro dvousekundové okno je predikce opatrnější, protože stále vidí úsek pozic ze spodní uličky s vysokými váhami.



Obrázek 5.8: Porovnání regrese nad krátkým a dlouhým oknem při přechodu do vedlejší uličky

## Minimální počet vzorků

Kromě velikosti okna je možné ladit i počet vzorků, pro které se v daném časovém intervalu regrese provede. Toto nastavení má velký vliv na chování v situacích, kdy objekt opustí lokalizovaný areál a server přestane počítat pozice a generovat nové vzorky.

## Minimální skóre

Určení prahu minimálního skóre dokáže řešit situace, kdy se objekt dlouhodoběji pohybuje těsně mimo uličku nebo ve vedlejší nemonitorované uličce, do které zasahuje signál z monitorované uličky. V praxi vniká posloupnost vzorků, které mají špatné signálové parametry a tím i relativně špatné měkké skóre od klasifikátoru. Nad oknem se vypočte průměrné skóre vzorků. Pokud průměr nedosahuje určeného prahu, regrese se neprovede a na výstup se nevygeneruje žádná pozice. V kombinaci s prahem minimálního počtu vzorků se prediktor deaktivuje pro různé okrajové situace.

## Zdrojové kódy

Zdrojové kódy jsou k nalezení na paměťovém médiu:

- `src/plots/plot_regression_window.py` - zdroj pro obrázky a grafy týkající se regresního modelu.
- `src/models/predict.py` - skript pro transformaci skórovaného datasetu na dataset s predikcí pozice.

Implementovaný systém shrnuje pseudokód 1. Tento algoritmus pracuje vždy s nejnovější množinou vypočtených pozic z RTLS platformy a po jejich zpracování je již nepotřebuje. Algoritmus si musí udržovat akorát částečně zpracované pozice v klouzavém okně, tedy historii pozic za posledních pár sekund.

---

### Algorithm 1 Klasifikace a regrese nad klouzavým oknem

---

Klasifikátor: GMM se složkami rozložení  $\mathcal{N}(\mu, \Sigma)$

Klouzavé okno s velikostí  $k$  milisekund.

Vstup: množina RTLS pozic  $F$  v čase  $T$

Výstup: nová pozice  $f_n$  se skóre stability v čase  $T$

1. Každou pozici v  $F$  klasifikuj pomocí GMM a získej skórované pozice  $F_s$ .
  2. Z množiny  $F_s$  vytvoř jednu pozici  $f_s$  jako vážený průměr, kde váhy jsou skóre.
  3. Pozici  $f_s$  přidej do klouzavého okna a vyřaď pozice starší než  $T - k$  milisekund.
  4. Vypočti průměrné skóre okna  $W_s$  a počet prvků okna  $W_l$ .
  5. Pokud  $W_s$  a  $W_l$  splňují minimum, proved' lineární regresi nad oknem.
  6. Regresním modelem odhadni novou pozici  $f_n$  v čase  $T$  se skóre stability  $W_s$ .
  7. Čekaj na nové pozice a vrať se na krok 1.
-

## 5.3 Vizualizační nástroje

Pozice v datasetu tvoří časovou řadu, kterou bychom chtěli animovat. Vizualizace dat nám poskytne lepší představu o tom, co data znamenají v kontextu 2D mapy pozic zařízení v monitorovaném prostředí. Data z platformy jsou také z principu real-time, takže je vhodné sekvenci pozic sledovat při změnách v čase.

Nástroj Tagio byl vytvořen v rámci této práce pro usnadnění vizualizace a transformace dat a pro emulaci dat získaných z poziční platformy. Momentálně je software ve fázi prototypu, který dokáže data přehrávat a zobrazovat ve webovém prohlížeči. Nástroje se implementačně skládají ze tří částí – jednou z částí je skript v jazyce Python 3 pro předzpracování datasetu. Další dvě části jsou moduly aplikace v jazyce Javascript.

### Parser

Parser je skript v Pythonu zpracovávající původní dataset z platformy. Dataset je ve formátu csv, tedy jednoduchá tabulka s hlavičkou a množinou záznamů. Parser překládá data do strukturovaného formátu JSON.

Na nejvyšší úrovni jsou data roztržena podle jednotlivých identifikátorů zařízení. Do záznamů jsou doplněny časové rozdíly mezi rámci vypočtené z absolutních časových razítek. Záznamy jsou potom organizované do časových rámců (podobně jako na obrázku 4.2).

V původním datasetu explicitně není obsažena informace o velikosti monitorovaného prostředí. Parser velikost prostředí odhadne pomocí výpočtu *bounding box* (BB) [17]. Berou se na vědomí všechny prostorové souřadnice datasetu a souřadnice všech instalovaných kotev. BB může být užitečný atribut pro vizualizaci a pro normalizaci dat.

### Replay

Modul replay umožňuje načíst a přehrát poziční data ve webovém prohlížeči. Modul implementuje logiku načítání dat ze souboru JSON a ovládání uživatelského rozhraní. Podobně jako jiné přehrávací aplikace nabízí možnosti přehrávat libovolnou rychlostí nebo krokovat po jednotlivých rámcích.

- Klávesa mezerník: pauza
- Klávesy C a V: zpomalení a zrychlení
- Klávesy , a .: posun rámce dozadu a dopředu

Datové soubory pro přehrávání se nachází ve složce `data`. Příložené poziční data ve formátu JSON lze prohlížet přetáhnutím souborů do okna prohlížeče, například:

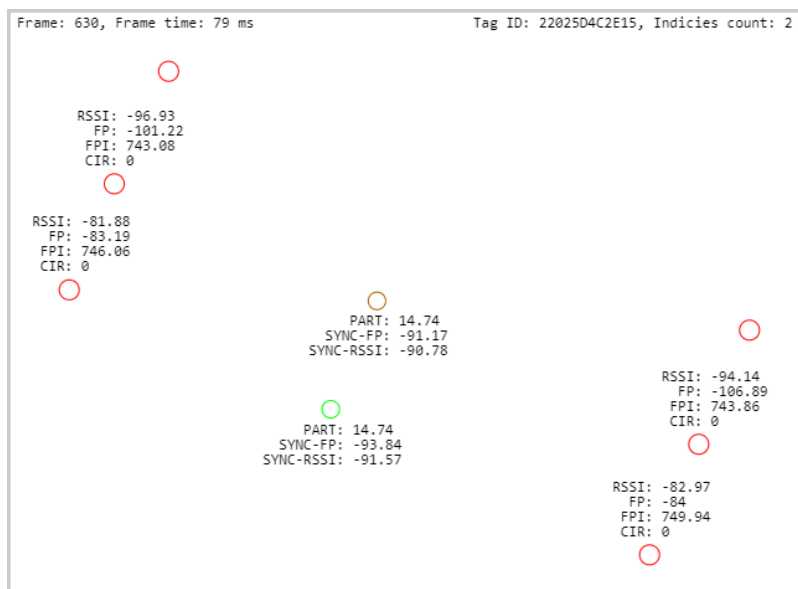
- `data_1.json` s původními pozicemi,
- `data_1_scored.json` s klasifikátorem skórovanými pozicemi,
- `data_1_predicted.json` s finálními pozicemi.

### Renderer

Vizualizace v úlohách získávání znalostí z dat je důležitá pro prvotní pochopení datasetu. Renderer je modul pro vizualizaci datasetu v jazyce Javascript. Pro zobrazení využívá canvas API v kontextu 2D grafiky [10]. Cílem je zobrazovat právě přehrávaná data replay



modulu a poskytovat tak lepší náhled na dataset. Na obrázku 5.9 je snímek obrazovky aplikace. Prostředí a dataset odpovídá půdorysu z obrázku 4.1. Červené značky jsou aktivní instalované UWB kotvy. Zelený kroužek reprezentuje správnou pozici lokalizovaného objektu a oranžový kroužek je alternativní záznam vypočtený jinou lokalizační buňkou. V levém horním rohu je číslo právě zobrazeného rámce (frame) a času do dalšího rámce (frame time).



Obrázek 5.9: Snímek obrazovky vizualizačního nástroje

# Kapitola 6

## Evaluace řešení

Tato kapitola se zabývá vyhodnocením implementovaného řešení na datech ze dvou reálných instalací. První z těchto instalací je stejná testovací instalace z kapitoly 4. Druhá instalace se nachází v jiném prostředí a poslouží tak jako dobrý zdroj evaluačních dat. Evaluace je provedena transformací původních dat z lokalizačního RTLS serveru na dataset nově odhadnutých pozic. Cílem kapitoly je vyhodnotit chování modelu pro různá zajímavá nastavení hyperparametrů, ukázat eliminaci odskoků a změřit výpočetní čas, který je v jednotlivých krocích potřebný k aplikaci. Pozice vypočtené původním lokalizačním systémem jsou porovnány s novými pozicemi - k této kapitole byly vytvořeny demonstrační videa, která se nachází na přiloženém paměťovém médiu.

### 6.1 Parametry modelů

Pro porovnání budou použité 2 podobné modely klasifikátoru, které se liší v počtu využitých signálových parametrů. První model BGMM je naučený na dvou parametrech Master-RSSI a Slave-RSSI, kde Gaussovská komponenta reprezentující třídu správných pozic má střed

$$\mu_{(1)} = [-83.12 \quad -83.76] \quad (6.1)$$

a kovarianční matici

$$\Sigma_{(1)} = \begin{bmatrix} 2.32 & -0.08 \\ -0.08 & 4.21 \end{bmatrix} \quad (6.2)$$

Druhý model je naučený na čtyřech parametrech Master-RSSI, Slave-RSSI, Master-FP a Slave-FP a modeluje třídu správných pozic se středem

$$\mu_{(2)} = [-83.05 \quad -83.63 \quad -86.71 \quad -85.90] \quad (6.3)$$

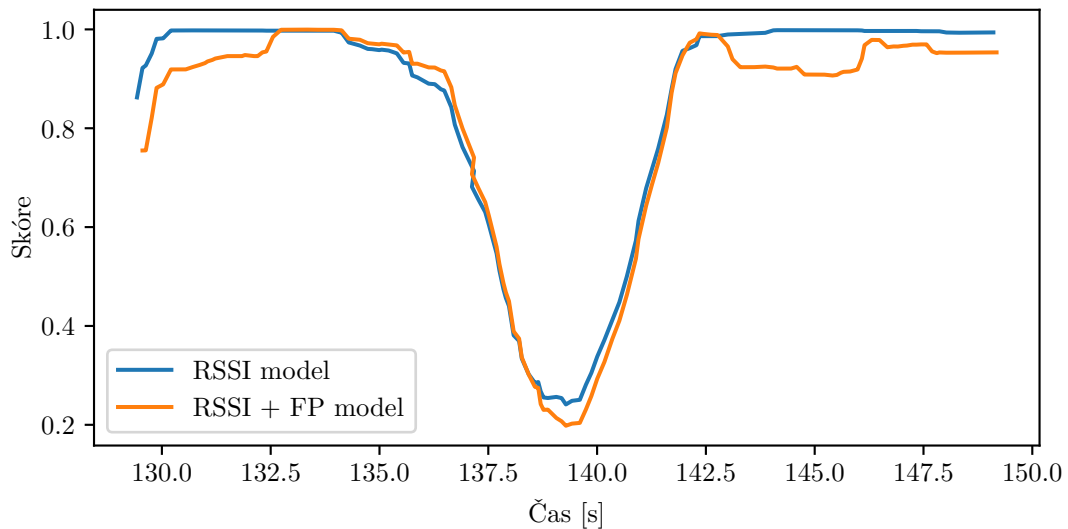
a kovarianční maticí

$$\Sigma_{(2)} = \begin{bmatrix} 2.12 & -0.05 & 3.09 & -0.19 \\ -0.05 & 4.35 & 1.57 & 5.17 \\ 3.09 & 1.57 & 8.4 & 1.75 \\ -0.19 & 5.17 & 1.75 & 6.86 \end{bmatrix} \quad (6.4)$$

Hyperparametry regrese a klouzavého okna jsou dále nastavené individuálně pro jednotlivé instalace. Výchozí velikost klouzavého okna byla experimentálně nastavena na 3000 milisekund, jelikož se při evaluaci osvědčila jako vhodná velikost pro odstranění náhodných odskoků. Predikce se potom provede pro minimálně 3 vzorky v tomto okně. Pro evaluaci první instalace je nastaven práh minimálního skóre na hodnotu 0.1 a pro druhou instalaci na 0.2. Práh musel být u druhé instalace zpřísněn kvůli poměrně dlouhým sekvencím odskoků, které klasifikátor ohodnotil vyšším skóre.

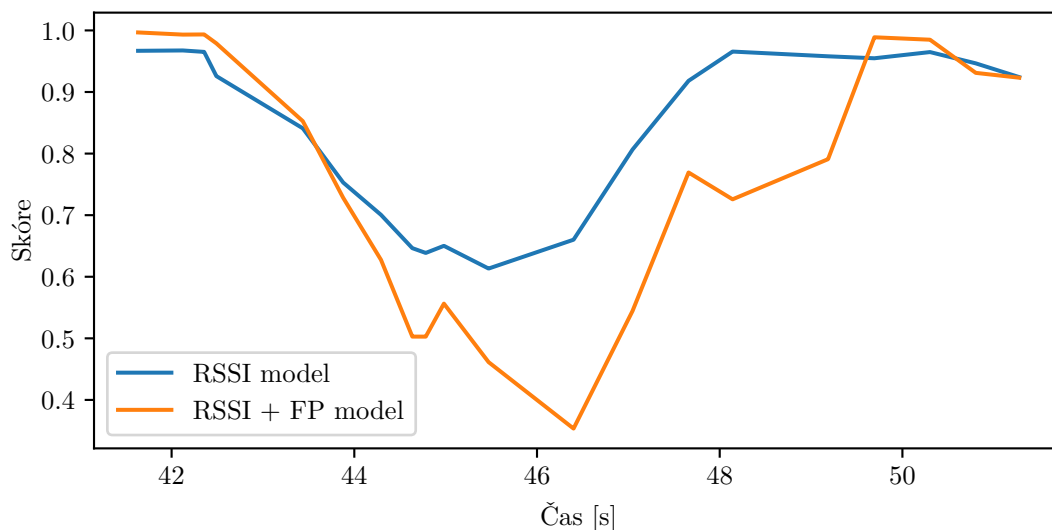
## 6.2 Skóre stability

Obrázek 6.1 ilustruje průběh vypočteného skóre, které reprezentuje stabilitu pozice. Skóre je získáno z průměrného skóre okna, nad kterým se počítá predikce. Je zde ukázaná situace přechodu mezi uličkami v první instalaci, kdy jsou signálové parametry slabé. Modrá křivka je Gaussovský model naučený pouze na parametrech RSSI. Oranžovou barvou jsou výsledky Gaussovského modelu, který je navíc naučen na parametrech FP. Oba modely obecně hodnotí pozice podobně. Model RSSI je jednodušší a ve stabilních situacích dává pozicím skóre 1.0. Model RSSI + FP je často přísnější i ve stabilních situacích a může tak částečně znehodnotit i relativně kvalitní pozice. Pro účely obecného zrobustnění stability pozice se tedy zdá být jednodušší RSSI model dostačující.



Obrázek 6.1: Skóre pozic klasifikátorů podle RSSI a RSSI+FP parametrů v první instalaci

V druhé evaluační instalaci se vyskytují místa, kde také dochází ke konfliktnímu vyhodnocení pozice (obrázek 6.2). V tomto prostředí jsou uličky rozčleněny do více částí. V některých místech je tak částečná viditelnost na více kotev různých lokalizačních buněk. U těchto situací jsou parametry RSSI stále relativně dobré, ale parametr FP slábně. Model naučený pouze na parametrech RSSI v těchto situacích hodnotí pozice lépe než model s dodatečnými parametry FP. Je tedy možné, že složitější varianta modelu bude v tomto typu instalací efektivněji odhalovat nestabilní pozice.

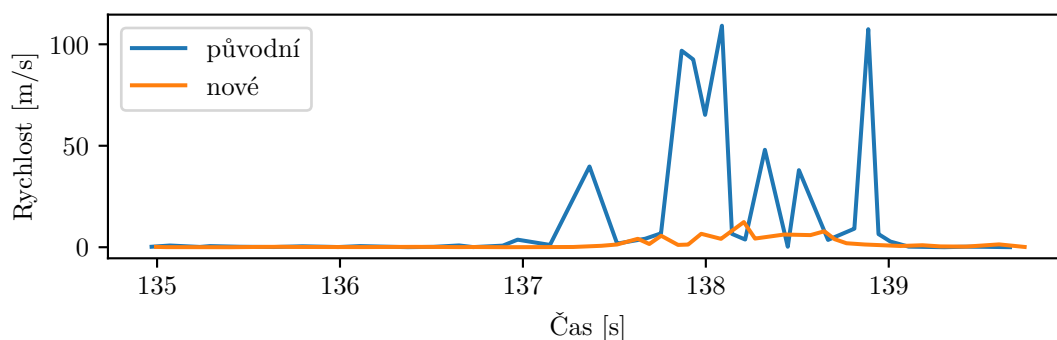


Obrázek 6.2: Skóre pozic klasifikátorů podle RSSI a RSSI+FP parametrů v evaluační instalaci

Pro ukázkou vyhodnocení průchodu instalacemi je doporučeno shlédnout demonstrační videa, která se nacházejí v adresáři `videos`. Poziční data jsou zde přehrávána ve vizualizačním nástroji, který emuluje průchody objektů v reálném čase. Video porovnává pozice z původního datového toku z RTLS serveru a pozice vypočtené algoritmem.

### 6.3 Eliminace odskoků

Odskočené či nestabilní pozice lze graficky zobrazit jako náhlé zrychlení pohybujícího se objektu. V nestabilních situacích původně vypočtená pozice kmitá mezi uličkami. Rychlost objektu lze odhadnout podle rozdílu pozic za čas. Na obrázku 6.3 je porovnání původních pozic a nových odhadnutých pozic v situaci přechodu mezi uličkami. Kvůli velkým odskokům tak původní pozice dosahují nerealistický rychlostí, zatímco křivka rychlostí nových pozic je zřetelně hladší. Model tedy dokáže uspokojivě eliminovat náhodné odskoky a zároveň zajistit relativně plynulý přechod mezi uličkami.



Obrázek 6.3: Odhadnuté rychlosti monitorovaného objektu původních a nových pozic

Odskoky pozic uvnitř lokalizovaného prostředí mezi jednotlivými uličkami lze tedy úspěšně řešit. Na řešení těchto situací se nejvíce podílí vlastnost regrese hledat lokální průměrnou pozici.

Existuje ale i jiný typ odskoků, ke kterým dochází při pohybu lokalizovaného objektu v nepokryté uličce. V tomto případě může pozici chybně vypočítat okrajová lokalizační buňka, a jelikož je tato pozice jediná dostupná v daném časovém intervalu, regresní algoritmus nebude mít jiná data pro lepší odhad. Normálně by tak výsledná pozice připadla právě na tuto jedinou chybnou pozici. Navržený systém proto ověřuje, zda se v klouzavém okně nachází dostatečné množství relevantních vzorků pozic. Tato situace je nejlépe demonstrována v druhém evaluačním průchodu, kdy objekt procházel mimo lokalizační prostor. Od časového rámce číslo 345 následuje přibližně 25 chybných odskoků, ke kterým náhodně docházelo po dobu 90 sekund (prvních 10 vyhodnocení je v tabulce 6.1). Systém s regresním oknem v tomto případě eliminoval všechny odskoky. V 9 případech se predikce neprovedla z důvodu nedostatečného počtu pozic v okně - mezi pozicemi je velká časová mezera. V ostatních případech by byla splněna podmínka počtu pozic v okně, ale nebyl dosažen práh minimálního skóre okna.

Číslo rámce	Časové razítko	Skóre	Počet pozic	Dosaženo skóre	Dosažen počet
345	1650350102489	0	1	Ne	Ne
346	1650350102824	0.0002	2	Ne	Ne
347	1650350103405	0.0006	3	Ne	Ano
348	1650350103986	0.0005	4	Ne	Ano
349	1650350104333	0.0004	5	Ne	Ano
350	1650350104587	0.0003	6	Ne	Ano
351	1650350108769	0.0010	1	Ne	Ne
352	1650350110372	0.0006	2	Ne	Ne
353	1650350120906	0	1	Ne	Ne
354	1650350125563	0	1	Ne	Ne

Tabulka 6.1: Příklad eliminace náhodných odskoků prediktorem při průchodu objektu vedlejší uličkou. Prediktor eliminuje odskoky na základě nedosažení minimálních prahů.

Podobná situace dále nastala od rámce číslo 463. V tento moment se objekt začal pohybovat směrem k vedlejší nemonitorované uličce. Signálové parametry se postupně zhoršují,

což se evidentně projeví i na průměrném skóre okna. V tuto chvíli je okno stále naplněné dostatečným počtem pozic, ale při propadu skóre pod hranici 0.2 se predikce neprovede a tím jsou eliminovány následující pozice, které již neodpovídají reálné pozici objektu. Situaci a vyhodnocení 10 časových momentů zaznamenává tabulka 6.2.

Číslo rámce	Časové razítko	Skóre	Počet pozic	Dosaženo skóre	Dosažen počet
463	1650350195916	0.4240	7	Ano	Ano
464	1650350195950	0.3710	8	Ano	Ano
465	1650350196170	0.3709	8	Ano	Ano
466	1650350196585	0.3706	8	Ano	Ano
467	1650350197051	0.2490	8	Ano	Ano
468	1650350197651	0.1247	8	Ne	Ano
469	1650350197753	0.1108	9	Ne	Ano
470	1650350198126	0	9	Ne	Ano
471	1650350198711	0	9	Ne	Ano
472	1650350198811	0	10	Ne	Ano

Tabulka 6.2: Příklad plynulého přechodu objektu do vedlejší nemonitorované uličky. Predikce se po zhoršení skóre pod hranici 0.2 vypne.

Na výše uvedených situacích je tedy dobře vidět, že navržený systém zvládne robustně řešit nestabilní situace uvnitř uliček a eliminovat náhodné odskoky při pohybu mimo areál.

## 6.4 Měření výkonu

Model musí být schopný vyhodnotit nová data dostatečně rychle, aby byl použitelný v reálném čase. Pro začátek lze za dostatečně dobrý výsledek považovat výpočetní čas v jednotkách milisekund.

Tabulka 6.3 obsahuje statistiku měření výkonnosti různě nastavených modelů. Výsledek by se dal shrnout jako uspokojivý vzhledem k tomu, že implementace běží v interpretovaném skriptovacím jazyce a jádrem výpočetních operací jsou metody knihovny Scikit-learn. Celkový čas výpočtu nové pozice lze získat součtem výpočetního času klasifikátoru a prediktoru. Ve většině případů je tedy celkový čas výpočtu kolem hodnoty 0.8 milisekund a neměl by přesáhnout 2 milisekundy.

Naměřené časy se příliš neliší mezi jednotlivými modely, pravděpodobně z důvodu velké režie spojené s Scikit-learn implementací, která je spíše stavěná na dávkové vyhodnocení modelů a experimentální offline použití. V oblasti výpočetní optimalizace výkonu je možné předpokládat značné zrychlení, pokud bude řešení přepsáno do produkční nízkoúrovňové implementace.

Model	Klasifikátor			Prediktor		
	min	průměr	max	min	průměr	max
RSSI, okno 1000	0.18 ms	0.20 ms	0.46 ms	0.52 ms	0.57 ms	1.03 ms
RSSI, okno 3000	0.18 ms	0.20 ms	0.34 ms	0.53 ms	0.58 ms	1.00 ms
RSSI+FP, okno 1000	0.18 ms	0.20 ms	0.63 ms	0.52 ms	0.57 ms	0.86 ms
RSSI+FP, okno 3000	0.18 ms	0.20 ms	0.66 ms	0.53 ms	0.60 ms	1.10 ms

Tabulka 6.3: Výpočetní časy modelů s různým nastavením parametrů

## 6.5 Perspektivy budoucí práce

System strojového učení vytvořený v rámci této práce není nutně konečným řešením. Jednou z největších výzev při zpracování tohoto řešení bylo přizpůsobení se vlastnostem poměrně limitované datové sady, která pochází čistě ze stávající implementace lokalizace v režimu 1D lokalizačních buněk. Vypočtené pozice touto metodou jsou pouze nejlepší dostupné odhady na 1D úsečce mezi UWB kotvami. Z principu tak nejsou k dispozici přesnější pozice, které by byli žádoucí pro tvorbu robustnějšího modelu. Pro další výzkum může být zajímavé vytvořit podmínky pro důkladnější a komplexnější lokalizační techniky, které by vedly k tvorbě pokročilejších modelů strojového učení.

První navrhovanou etapou je prozkoumat možnosti využití přesnějších pozičních technik pro tvorbu kalibrační datové sady, na které bude probíhat strojové učení modelu. Taková datová sada by kromě pozic odhadnutých současným UWB systémem obsahovala i mnohem přesněji a stabilněji určené pozice pocházející z kalibračního zařízení jakým je například LiDAR.

Druhou etapou by mohla být implementace validní simulace prostředí lokalizačního systému. Úlohou této etapy je vymodelovat simulované prostředí, model zvalidovat vůči reálnému prostředí a následně simulaci použít ke sběru pozičních dat a evaluaci nových metod lokalizace a experimentů. Po nasbírání dostatečně velkého množství označených a reprezentativních dat napříč několika instalacemi by mohlo být realistické nasadit techniky hlubokého učení nad sekvenčními daty z oblasti rekurentních neuronových sítí.

### 6.5.1 Kooperace UWB a lidarů

Jednou z hlavních nevýhod systému strojového učení implementovaného v této práci je nutnost užití neoznačené kalibrační datové sady pro trénování modelu. Jak již bylo ukázáno v kapitole analýzy dat, je možné pozice ručně označit a roztrždit, ovšem tento postup by byl v praxi velmi pracný a neefektivní. Využití technik strojového učení bez učitele a automatické shlukové analýzy je o něco praktičtější – výběr jednotlivých komponent Gaussovské směsice a ladění několika málo hyperparametrů se zdá být jako v praxi lépe použitelný postup. Zde ale narážíme na předpoklad, že kalibrační dataset tvoří Gaussovské shluky, které oddělují třídy pozic. Tento předpoklad ale nemusí nutně platit v libovolné instalaci.

V oblasti robotiky a autonomního řízení se často využívají kombinované kooperativní lokalizační techniky, které vedou k velmi robustní a přesné lokalizaci. Jednou z atraktivních kombinací je využití klasické UWB lokalizace a optické odometrie nad lidarovými daty [3] [18]. Cílem této kooperativní lokalizace by bylo přesnější zmapování instalace. K běžným pozičním datům z UWB platformy by také byly k dispozici přesnější pozice z lidarové

odometrie. Nad takovou datovou sadou by bylo možné efektivně využít mocnější techniky učení s učitelem. Vzdálenost pozice vypočtené UWB lokalizací od referenční lidarové pozice lze potom použít například jako hodnocení kvality pozice. Model strojového učení by se potom relativně snadno naučil, které prostory signálových UWB parametrů odpovídají přesným pozicím.

Hlavní výzvou tohoto přístupu bude návrh kalibračního postupu, který by vedl k tvorbě správné a reprezentativní datové sady. Bylo by nutné zajistit korektní použití lidarového zařízení a synchronizaci s UWB systémem. Při implementaci by mohl asistovat software pro automatickou odometrii v reálné čase, například LeGO-LOAM [16].

### 6.5.2 Simulace prostředí

Signálové charakteristiky v prostředí s UWB zařízeními je možné simulovat například pomocí technik *ray tracing* (RT) [9]. Využití simulovaného prostředí by vedlo k snadnějšímu generování pozičních dat a experimentování s pokročilejšími technikami strojového učení, které zpravidla vyžadují velké množství dat. Teoreticky by bylo možné vymodelovat množinu typických lokalizačních prostředí, které odpovídají instalacím v reálném prostředí. Například článek *UWB Indoor Localization Using Deep Learning LSTM Networks* [13] se zabývá kompletním řešením lokalizace pomocí hlubokého učení. Autoři článku implementovali síť s vrstvami typu long short-term memory (LSTM). Síť zde pracuje se surovými signály, které pocházejí ze simulovaného prostředí. Řešení popsané v článku uvažuje klasickou 2D lokalizační instalaci, každopádně lze předpokládat úspěšné využití těchto technik i pro 1D lokalizaci.

Jako příklad slibného RT simulačního nástroje lze uvést například open-source software PyLayers [2]. Nástroj je navržen pro simulaci rádiového šíření UWB signálu ve vnitřním uzavřeném prostředí. Použití nástroje by vyžadovalo:

1. Definici a modelování simulovaného prostředí. V tomto kroku by bylo nutné importovat a transformovat půdorys instalací na formát modelu prostředí simulátoru, a to včetně správných nastavení materiálů objektů a překážek v tomto prostředí.
2. Specifikaci komunikujících UWB zařízení, která mají odpovídající vlastnosti RTLS platformy firmy Sewio.
3. Provedení samotné simulace propagace signálů a zhodnocení výsledků.

Iterativním vylepšováním modelovaného prostředí by bylo možné dosáhnout akceptovatelné úrovně validity modelu. Ze získaných výsledků simulace lze potom sestavit datovou sadu podobnou kalibrační datové sadě z fyzického prostředí. Největším přínosem simulace je především množství dat, které lze v relativně krátkém čase vygenerovat. V porovnání s metodou sběru dat za pomoci lidarů dále odpadá potřeba interagovat s fyzickým prostředím. Experimenty lze tak provádět téměř úplně ve virtuálním prostředí, což může značně zvýšit efektivitu výzkumu.



# Kapitola 7

## Závěr

V této práci byly prozkoumány možnosti využití strojového učení na datech pocházejících z RTLS instalace firmy Sewio. Lokalizační datová sada byla analyzována z pohledu přesnosti pozic ve vztahu k signálovým parametrům. Poznatky z analýzy vedly k návrhu systému strojového učení založeném na Gaussovském odhadu rozložení pro klasifikaci a hodnocení kvality pozic pomocí měkkého skóre. Ohodnocené pozice jsou dále zpracovány prediktorem, který dokáže z dostupné krátkodobé historie ohodnocených pozic odhadnout pravděpodobnou trajektorii pohybu objektu a vypočítat tak novou pozici. Systém předpokládá dostupnost krátké kalibrační datové sady, kterou je možné použít jako trénovací množinu pro vytvoření Gaussovského modelu a odladění hyperparametrů pro potřeby konkrétní lokalizační instalace. Navrhovaný systém byl implementován za pomoci frameworku Scikit-learn v jazyce Python 3.

Výsledky experimentálního použití systému na emulovaném datovém toku RTLS platformy ukazují zvýšení stability pohybu objektu v nestabilních situacích na okrajích lokalizačních buněk, kde v původním RTLS řešení často dochází ke konfliktnímu vyhodnocení pozice. Implementovaný systém dokáže řešit situace odskoku pozic uvnitř i mimo lokalizovaný prostor. K demonstraci funkce systému byla vytvořena ukázková videa, která porovnávají pozice původní a vylepšené lokalizace v prostředí reálné instalace. Video je dostupné na přiloženém paměťovém médiu. Dodatečné informace jsou k nalezení v příloze A.

Práci je možné dále zdokonalit výkonnější implementací systému a případně vhodným začleněním do produkční implementace platformy. Další podobné směry výzkumné práce mohou směřovat k možnostem tvorby komplexnější a přesnější datové sady, která by umožnila využít pokročilejší techniky strojového učení.

# Literatura

- [1] *Pandas-dev/pandas: Pandas*. Zenodo, únor 2020. DOI: 10.5281/zenodo.3509134. Dostupné z: <https://doi.org/10.5281/zenodo.3509134>.
- [2] BERNARD UGUEN. *PyLayers: Propagation and Localisation Simulator* [online]. [cit. 2022-04-20]. Dostupné z: <https://github.com/pylayers/pylayers>.
- [3] CHUANWEI, Z., XIAOWEN, M. a PEILIN, Q. LiDAR-IMU-UWB-Based Collaborative Localization. *World Electric Vehicle Journal*. 2022, sv. 13, č. 2. DOI: 10.3390/wevj13020032. ISSN 2032-6653. Dostupné z: <https://www.mdpi.com/2032-6653/13/2/32>.
- [4] DECAWAVE LTD.. *Dokumentace produktu Decawave* [online]. [cit. 2021-12-11]. Dostupné z: <https://www.decawave.com/product-documentation/>.
- [5] DENG, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*. IEEE. 2012, sv. 29, č. 6, s. 141–142.
- [6] GERON, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 1st. O'Reilly Media, Inc., 2017. ISBN 1491962291.
- [7] HALEVY, A., NORVIG, P. a PEREIRA, F. The Unreasonable Effectiveness of Data. USA: IEEE Educational Activities Department. mar 2009, sv. 24, č. 2, s. 8–12. DOI: 10.1109/MIS.2009.36. ISSN 1541-1672. Dostupné z: <https://doi.org/10.1109/MIS.2009.36>.
- [8] IEEE 802 LAN/MAN STANDARDS COMMITTEE. *IEEE 802.15.4-2020 - IEEE Standard for Low-Rate Wireless Networks* [online]. [cit. 2021-12-27]. Dostupné z: [https://standards.ieee.org/standard/802\\_15\\_4-2020.html](https://standards.ieee.org/standard/802_15_4-2020.html).
- [9] LAARAIEDH, M., AMIOT, N. a UGUEN, B. Efficient Ray Tracing Tool for UWB Propagation and Localization Modeling. In: Duben 2013, s. 2307–2311. ISBN 978-1-4673-2187-7.
- [10] MOZILLA. *Dokumentace Canvas API* [online]. [cit. 2022-01-08]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API).
- [11] OPPERMANN, I., HÄMÄLÄINEN, M. a IINATTI, J. *UWB Theory and Applications*. John Wiley & Sons, Ltd, 2004. ISBN 0-470-86917-8.
- [12] PEDREGOSA, F. a VAROQUAUX, G. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, sv. 12, s. 2825–2830.

- [13] POULOSE, A. a HAN, D. UWB indoor localization using deep learning LSTM networks. *Applied Sciences*. Zář 2020, sv. 10, s. 6290. DOI: 10.3390/app10186290.
- [14] RASHID, S., SABIRA, K., BORHANUDDIN, A. a M.A., K. Ultra-Wideband (UWB) Geolocation in NLOS Multipath Fading Environments. In: Prosinec 2005, sv. 2, s. 6 pp. DOI: 10.1109/ICON.2005.1635673. ISBN 1-4244-0000-7.
- [15] SEWIO NETWORKS, S.R.O.. *Veřejná dokumentace platformy RTLS Sewio* [online]. [cit. 2021-12-25]. Dostupné z: <https://docs.sewio.net/docs>.
- [16] TIXIAO SHAN AND BRENDAN ENGLT. *LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain* [online]. [cit. 2022-04-20]. Dostupné z: <https://github.com/RobustFieldAutonomyLab/LeGO-LOAM>.
- [17] WIKIPEDIA. *Minimum bounding box* [online]. [cit. 2022-01-05]. Dostupné z: [https://en.wikipedia.org/wiki/Minimum\\_bounding\\_box](https://en.wikipedia.org/wiki/Minimum_bounding_box).
- [18] ZHANG, J. a SINGH, S. LOAM : Lidar Odometry and Mapping in real-time. *Robotics: Science and Systems Conference (RSS)*. Leden 2014, s. 109–111.

# Příloha A

## Obsah paměťového média

Na přiloženém CD se nachází text práce, skripty na generování grafů, zpracování dat, učení modelu a evaluaci.

- doc/ - text práce ve formátu pdf a Latex zdrojové soubory k jeho generování
- data/ - lokalizační data z reálných instalací. Původní datové sady jsou zde ve formátu csv. Zpracované data pro účely vizualizace v nástroji jsou ve formátu json. V podadresáři results se nacházejí příklady dat výsledných lokalizačních průchodů.
- src/plots - zdrojové kódy pro generování matplotlib grafů
- src/models - zdrojové kódy pro modelování a testování jednotlivých kroků algoritmu. cluster.py slouží pro tvorbu Gaussovského modelu. Další skripty average.py a predict.py vznikly pro snadnější testování podkroků algoritmu. evaluate.py provádí algoritmus od začátku do konce.
- src/tools - skripty pro transformaci datové sady a vizualizační nástroj

V adresáři videos se nachází demonstrační ukázky 2 průchodů instalacemi. Videá porovnávají původní pozice a algoritmem nově odhadnuté. Video data\_1\_demo.mp4 navíc ukazuje živý průchod natočený na kameru.

## Příloha B

# Použití vizualizačního nástroje

Vizualizační nástroj je možné použít k prohlédnutí datasetu průchodu lokalizační instalací. Příklad použití je následující:

1. Otevřete `src/tools/index.html` ve webovém prohlížeči.
2. Měla by se zobrazit ikona pro drag and drop JSON souboru. Zde přetáhněte některou datovou sadu ve formátu JSON, například `data_2.json`.
3. Nyní by měl být vidět průchod tagu včetně signálových parametrů, pokud se jedná o původní data, jinak pouze confidence score, pokud je dataset již upravený.
4. Kdyby se pozice přehrávaly rychle (v levém horním rohu se napíše `undefined ms`), je potřeba dataset zpracovat skriptem `src/tools/_add_frametimes.py`.
5. Pro nové přehrávání stačí refresh okna (F5).