



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**KLASIFIKÁCIA PRIAMEHO A ODRAZENÉHO
SIGNÁLU POMOCOU VSTAVANÉHO SYSTÉMU**

CLASSIFICATION OF DIRECT AND REFLECTED SIGNAL USING EMBEDDED SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MIROSLAV CHALKO

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK

BRNO 2022

Zadání bakalářské práce



Student: **Chalko Miroslav**
Program: Informační technologie
Název: **Klasifikace přímého a odraženého signálu pomocí vestavěného systému**
Classification of Direct and Reflected Signal Using Embedded System
Kategorie: Zpracování signálů

Zadání:

1. Prostudujte vývojovou platformu STM32Cube.AI a výstup lokalizačního zařízení na bázi technologie UWB.
2. Seznamte se se základními postupy v oblasti strojového učení a možnostmi jejich využití pro účely klasifikace signálů.
3. Zvolte vhodnou metodu strojového učení schopnou klasifikovat přímý a odražený signál, případně navrhněte pro tento účel vlastní metodu.
4. Zvolenou klasifikační metodu implementujte v jazyce C s využitím platformy STM32 H7.
5. Zhodnoťte dosažené výsledky z pohledu výkonnosti, spolehlivosti a energetické spotřeby, diskutujte možná rozšíření.
6. Prezentujte klíčové vlastnosti řešení formou plakátu a krátkého videa.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr 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: 11. května 2022

Datum schválení: 29. října 2021

Abstrakt

Cielom tejto práce je navrhnúť a implementovať algoritmus klasifikácie signálov pri lokalizácii objektov použitím technológie Ultra-wideband. Klasifikačná metóda by mala rozpoznať prekážku medzi prijímačom a vysielačom, teda klasifikovať signály na tie s priamou viditeľnosťou (LOS – Line Of Sight) a bez priamej viditeľnosti (NLOS – Non Line Of Sight). Tento systém musí byť dostatočne rýchly a jednoduchý, aby detekcia v reálnom čase bola možná priamo vo vstavanom systéme. Pri hľadaní riešenia boli preskúvané viaceré klasifikačné metódy, z ktorých najlepšie výsledky mali rôzne varianty rozhodovacích stromov. Vzhľadom na výkon cieľového zariadenia bola zvolená klasifikačná metóda Random forest ako finálne riešenie. Tento klasifikátor dosahuje úspešnosť až 89% pri vyhodnotení na datasete. Pri reálnom nasadení bol schopný rozpoznať objekt medzi vysielačom a prijímačom. Klasifikácia a výpočet parametrov trvá 6000 inštrukčných cyklov a model zaberá 4kB pamäte. Výsledky tejto práce umožňujú zdokonaľiť existujúce riešenia detekcie NLOS signálov, a tým zvýšiť presnosť lokalizácie pri sledovaní polohy objektov vo vnútorných priestoroch.

Abstract

This thesis aims to design and implement an algorithm for classification of signals that are used for tracking objects using ultra-wideband technology. The classification method should be able to detect an obstruction between receiver and transmitter, which means to classify signals as those with line of sight (LOS) and non-line of sight (NLOS). This system must be quick and lightweight enough, so real-time detection can be achieved directly in the embedded system. While searching for the solution, multiple classification methods were examined. The best-performing ones involved numerous variants of decision tree classifiers. Considering the restricted computing power of embedded devices, random forest classifier was chosen as the final solution. This classification method was able to achieve accuracy of up to 89% while evaluating the dataset. When deployed in real-life environment, it was able to detect an object between transmitter and receiver. Classification and calculation of parameters takes 6000 instruction cycles and the algorithm fits into 4kB of memory. Results of this thesis enable improvement of existing solutions for detection of NLOS signals that degrade tracking performance. This will boost the accuracy of localization while tracking objects in indoor environments.

Klíčové slová

UWB, Signály, klasifikácia, DW1000

Keywords

UWB, Signals, classification, DW1000

Citácia

CHALKO, Miroslav. *Klasifikácia priameho a odrazeného signálu pomocou vstavaného systému*. Brno, 2022. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

Klasifikácia priameho a odrazeného signálu pomocou vstavaného systému

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Václava Šimka. Ďalšie informácie mi poskytol pán Ing. Ľubomír Mráz zo spoločnosti Sewio Networks s.r.o.. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Miroslav Chalko
15. mája 2022

Podakovanie

Rád by som poďakoval vedúcemu práce Ing. Václavovi Šimkovi za rady a pomoc pri vypracovaní. Taktiež by som rád poďakoval pánovi Ing. Ľubomírovi Mrázovi a pánovi Ing. Radimovi Lipkovi zo spoločnosti Sewio Networks s.r.o. za poskytnutie potrebného hardvéru a zaškolenie do problematiky práce.

Obsah

1	Úvod	2
2	Lokalizácia a signály	4
2.1	Priame a odrazené signály	4
2.2	Ultra-wideband	5
2.3	Techniky lokalizácie	6
2.4	DW1000	7
3	Klasifikácia a existujúce riešenia	10
3.1	Klasifikačné metódy	10
3.2	Metodika evaluácie klasifikačných metód	11
3.3	Dataset	13
3.4	Existujúce riešenia	14
4	Návrh riešenia	18
4.1	Cieľ práce	18
4.2	Analýza dát	19
4.3	Výber klasifikačnej metódy	21
4.4	Výber parametrov	22
5	Implementácia	24
5.1	Klasifikačný model	24
5.2	Export do jazyka C	25
5.3	Knižnica pre kotvu	27
5.4	Vyhodnotenie	30
6	Záver	33
	Literatúra	35
A	Obsah priloženého pamäťového média	37

Kapitola 1

Úvod

Rôzne druhy lokalizačných systémov sú používané na dennej báze každým z nás. Medzi tie najpopulárnejšie používané bežnými ľuďmi patria globálne pozičné systémy. V závislosti na počasí a prostredí, pri najlepších podmienkach dosahujú presnosť až niekoľkých desiatok centimetrov. Medzi tieto systémy patrí napríklad GPS, GLONASS, BEIDOU a GALILEO. Na vonkajšiu lokalizáciu sú zároveň použiteľné aj technológie ako mobilné siete alebo radar.

V poslednej dobe je ale čoraz väčší dopyt po sledovaní objektov vo vnútorných priestoroch. Na tento účel však vyššie uvedené systémy nie sú vhodné. Steny budov dokážu utlmiť signál a tým pádom razantne degradovať presnosť. Tieto technológie zároveň podliehajú prísnyim legislatívnym obmedzeniam, ktoré znemožňujú súkromnú prevádzku. Z toho dôvodu medzi doteraz najvyužívanejšie technológie IPS¹ patria Bluetooth a WiFi. Tieto technológie sú využívané napríklad pri manažmente tovaru v rámci skladov a výrobných hál, alebo sledovanie pohybu strojov a ľudí z dôvodu zvýšenia efektivity a bezpečnosti. Medzi potenciálne využitia patrí napríklad aj navigácia v rozsiahlych budovách ako sú letiská a výstaviská.

Nedostatkom vyššie spomínaných IPS technológií je slabá presnosť² a spoľahlivosť. Tento problém je vyriešený relatívne novou technológiou UWB³, ktorá dosahuje presnosti až niekoľkých centimetrov. Medzi ďalšie výhody patrí zvýšená odolnosť voči šumu a interferencii, veľmi nízke oneskorenie, nízka spotreba a vyššia spoľahlivosť. Žiadna technológia ale nie je dokonalá. Pokiaľ neexistuje priama viditeľnosť medzi vysielačom a prijímačom, presnosť merania sa zhorší až na jednotky metrov. Práve preto je nutné detekovať a reportovať takúto situáciu.

Bolo vypracovaných viacero štúdií, ktoré sa pokúšajú o detekciu priamej viditeľnosti, no žiadna z nich nevyhodnocuje detekciu priamo vo vstavanom systéme. Súčasnú industriálne nasadenú riešenie pochádza z dokumentácie integrovaného obvodu, ktorý poskytuje možnosť použitia technológie UWB. Presnosť tohto riešenia je však minimálna. Tento nedostatok a potenciál na zlepšenie je hlavnou motiváciou mojej práce.

Na začiatku mojej práce vysvetlím ako sa správajú signály vo vnútorných priestoroch a aké problémy predstavujú pri použití technológie UWB. Následne stručne predstavím klasifikačné metódy a vyhodnotím existujúce riešenia. Na základe zistených výsledkov navrhmem a implementujem klasifikátor do vstavaného systému, overím jeho funkčnosť a efektívnosť. Nakoniec zhodnotím výhody a nevýhody môjho riešenia a navrhmem potenciálne vylepšenia.

¹IPS – Indoor Positioning System je pozičný systém používaný vo vnútorných priestoroch.

²Presnosť lokalizácie pomocou Bluetooth a WiFi je typicky len niekoľko metrov.

³UWB – Ultra-wideband (ultra široké pásmo) je bezdrôtová technológia pracujúca na širokom pásme.

Poznámka k zadaniu práce

Pri zadaní práce bolo predpokladané využitie platformy STM32Cube.AI⁴, ktorú podporujú novšie procesory typu STM32 H7 používajúce ARM Cortex M7. Táto platforma uľahčuje implementáciu náročných klasifikačných algoritmov pomocou optimalizovanej knižnice, ktorá využíva akceleráciu poskytovanú procesorom. Zároveň dokáže exportovať váhy klasifikačných modelov a skomprimovať ich tak, aby nezaberali veľa miesta v pamäti zariadenia. Práve procesory typu STM32 H7 sú použité v novšej verzii hardvéru (kotiev), ktorý vyvíja firma Sewio Networks s.r.o.⁵.

Po konzultácií sme sa však rozhodli vypustiť použitie týchto procesorov. Po príchode globálnej pandémie bol svet zasiahnutý nedostatkom materiálov, hlavne nedostatkom silikónu. Z toho dôvodu aj dostupnosť, aj vývoj nového hardvéru je problematický. Navyše mnou navrhnutý algoritmus nepočíta s využívaním komplexných výpočtov, takže klasifikáciu je možné implementovať aj do vstavaného systému, ktorý nepodporuje akceleráciu výpočtov. Preto sme sa rozhodli použiť staršiu, ale časom overenú verziu kotvy na báze ARM Cortex M4.

⁴STM32Cube.AI – Vid' [st.com/en/embedded-software/x-cube-ai](https://www.st.com/en/embedded-software/x-cube-ai).

⁵Sewio Networks s.r.o. – Vid' [sewio.net](https://www.sewio.net).

Kapitola 2

Lokalizácia a signály

Základom úspešnej klasifikácie je porozumenie riešenému problému a zoznámenie sa s dostupnými parametrami. Účelom tejto kapitoly je predstaviť hardvérovú a technickú stránku problematiky. Ako prvé stručne popíšem fyzikálne vlastnosti signálov a ich správanie. Následne predstavím technológiu UWB a jej použitie pri lokalizácii. V poslednej časti tejto kapitoly sa budem zaoberať integrovaným obvodom, ktorý je použitý ako rozhranie medzi MCU¹ a fyzickou vrstvou².

2.1 Priame a odrazené signály

V rámci tejto práce pojem signál predstavuje elektromagnetické vlnenie špeciálne modulované tak, aby mohlo niesť určité informácie. V ideálnom prostredí, napríklad vo vákuu, sa elektromagnetické vlny šíria do všetkých strán s konštantnou rýchlosťou a spontánne nemenia svoje vlastnosti. No pri každej zmene prostredia, cez ktoré toto vlnenie prechádza, môže nastať zmena frekvencie, amplitúdy, fázového posuvu, smeru, rýchlosti šírenia, alebo viacerých z týchto efektov naraz. Miera zmeny vlastností je vždy definovaná materiálom, na ktorý vlna narazí. Vlnenie taktiež môže interagovať s inými vlnami a tým zmeniť svoje vlastnosti. Tento jav sa nazýva interferencia, ktorá môže byť pozitívne, alebo negatívne ovplyvniť amplitúdu. Ostatné prípady možnej zmeny vlastností, ako napríklad dopplerov efekt³, sú pre účely tejto práce zanedbateľné.

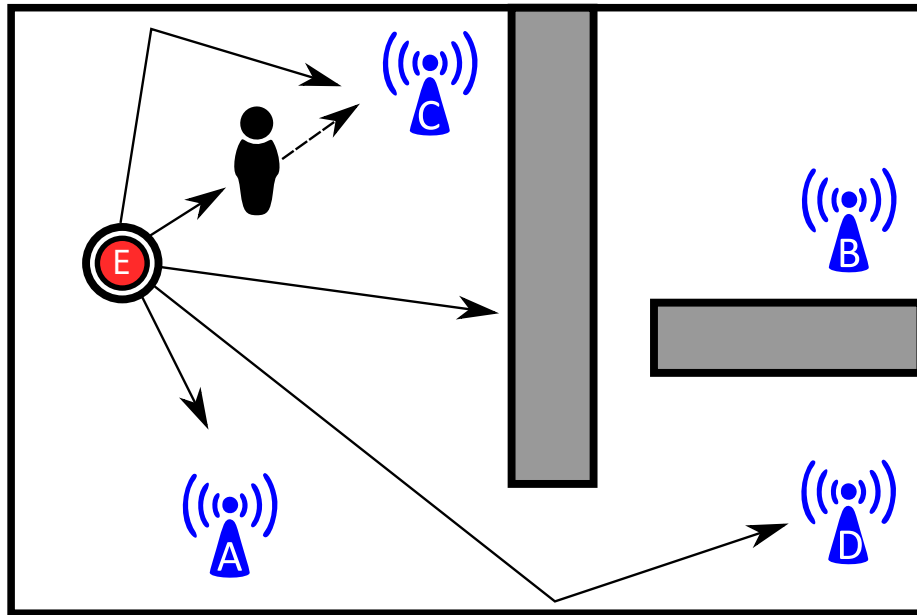
Pri lokalizácii objektov sú tieto efekty nežiadúce a degradujú presnosť lokalizácie. Prejavujú sa najmä vo vnútorných prostrediach, kde signál jednoducho narazí na niekoľko prekážok, ktoré tvoria napríklad steny budovy, skladované objekty a ľudia. Pokiaľ existuje priama viditeľnosť medzi vysielateľom a prijímačom, prvý prijatý signál bude vždy neskreslený, keďže priama cesta je vždy najkratšia a najrýchlejšia. Až ostatné signály budú do určitej miery skreslené nakoľko sa jedná o odrazy, viď obrázok 2.1. Problém však nastáva pokiaľ neexistuje priama viditeľnosť. V takom prípade priamy signál môže byť utlmený až pod úroveň šumu a k prijímaču sa dostanú len odrazené signály. Pri výpočte vzdialenosti na základe doby trvania letu signálu tak vzniká pozitívna chyba merania. Chybu merania spôsobujú nielen prekážky, ale aj prostredie samotné. Pri zmene teploty a vlhkosti vzduchu

¹MCU – Micro Controller Unit (jednočipový počítač, mikrokontrolér) je integrovaný obvod obsahujúci plne funkčný počítač schopný ovládať periférie.

²Fyzická vrstva – Zaisťuje prenos signálov po určitom médiu, napríklad vzduchom alebo káblom. Pracuje priamo s elektrickými impulzmi, definuje napríklad moduláciu, prenosovú rýchlosť, a pod..

³Dopplerov efekt – Elektromagnetické vlnenie mení frekvenciu pokiaľ sa jeho zdroj pohybuje v priestore.

sa mení aj dopad na šírenie elektromagnetického vlnenia. Tento problém je však v kontexte mojej práce ignorovaný, nakoľko použité dáta boli nazbierané v rovnakých prostrediach.

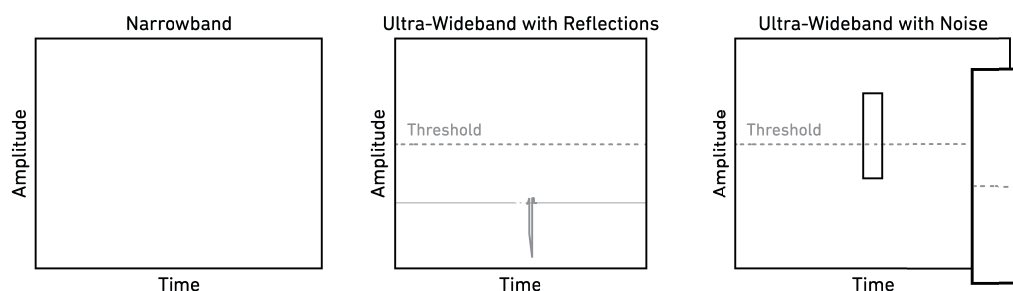


Obr. 2.1: Príklad šírenia signálov vo vnútornom priestore. Objekt E je vysielateľ. V prípade A je signál prijatý v plnej kvalite. Na rozdiel od toho prijímač B znázorňuje úplné zatienie. Prípad C znázorňuje príjem aj priameho, aj odrazeného signálu. V situácii D sa k prijímaču dostane už len odrazený signál.

2.2 Ultra-wideband

Ako napovedá už názov technológie, jedná sa o typ komunikácie na širokom pásme, rádovo v stovkách MHz⁴. V porovnaní šírka pásma WiFi je 20MHz alebo 40MHz, šírka pásma Bluetooth je 1 MHz. Široké pásmo síce umožňuje vysoké rýchlosti prenosu dát, ale pri lokalizácii objektov rýchlosť prenosu nie je prioritou. Forma UWB signálov je definovaná podľa IEEE 802.15.4 [2], upresnená v[4]. Dáta sú vysielané v zhlukoch veľmi úzkych pulzov, širokých len niekoľko nanosekúnd, čo redukuje náchylnosť na nepriaznivé efekty pri lokalizácii spomínané v predchádzajúcej kapitole 2.1, viď obrázok 2.2. Úzke pulzy s rýchlou nábežnou hranou zároveň umožňujú presné meranie času príchodu signálu, na základe ktorého je následne vypočítaná poloha objektu. V porovnaní, lokalizácia pomocou Bluetooth alebo WiFi sa spolieha na silu prijatého signálu, keďže čas prijatia nie je možné určiť s dostatočnou presnosťou. Práve vďaka možnosti presného merania času dosahuje UWB lepšie výsledky ako ostatné lokalizačné techniky. Medzi ďalšie výhody patrí napríklad veľmi nízka odozva (menej ako 1 milisekunda), nízka spotreba energie, alebo možnosť sledovania veľkého počtu objektov naraz (typicky až desiatitisíce objektov). Viac informácií o porovnaní UWB a iných lokalizačných technológií v [19].

⁴FCC (Federal Communications Commission) definuje UWB signál ako signál, ktorého šírka pásma dosahuje viac ako 500MHz, alebo aspoň 20% jeho strednej frekvencie [11].



Obr. 2.2: Príklad UWB signálu a prirodzenej odolnosti voči odrazom a šumu. Prevzaté z [19].

2.3 Techniky lokalizácie

Pri určovaní polohy pomocou UWB RTLS⁵ sa typicky hovorí o 2 druhoch zariadení – kotva a značka. Značka je zariadenie, ktoré je umiestnené na požadovaný objekt k lokalizácii. Kotva je staticky umiestnená vo vyvýšenej polohe, slúži na určovanie polohy značiek. Lokalizácia samotná je definovaná podľa IEEE 802.15.4z [4]. Riešenia lokalizácie sú ilustrované na obrázku 2.3.

Time difference of arrival

Tento prístup vyžaduje použitie viacerých kotiev, ale medzi kotvou a značkou prebieha minimálna komunikácia, čo drasticky zníži spotrebu energie. Značka raz za prednastavenú periódu vyšle signál, ktorý zachytia všetky kotvy. Tie následne reportujú prijatie signálu centrálnemu systému, ktorý vypočíta presnú pozíciu značky na základe času prijatia na každej kotve. Realizácia tohoto prístupu vyžaduje synchronizáciu času na všetkých kotvách s nanosekundovou presnosťou. Keďže oscilátor⁶ generujúci hodinový signál pre kotvu si nevie dlhodobo udržať presnosť, synchronizácia musí prebiehať každú sekundu.

Two Way Ranging

Pri použití tejto metódy prebieha krátka výmena správ medzi kotvou a značkou, na základe ktorej je možné vypočítať čas letu signálu a následne aj vzdialenosť. Pri meraní súradníc v priestore je nutné zopakovať celý proces niekoľkokrát a pomocou triangulácie vypočítať finálnu pozíciu. Kvôli zvýšenému počtu vymenených správ táto metóda nie je vhodná pri implementáciách v prostrediach, v ktorých sa nachádza veľké množstvo značiek. Použitia zahŕňajú prípady, pri ktorých kotva nemusí byť stacionárny objekt. Napríklad pri detekcii priblíženia, alebo počas merania vzdialenosti pomocou mobilného zariadenia.

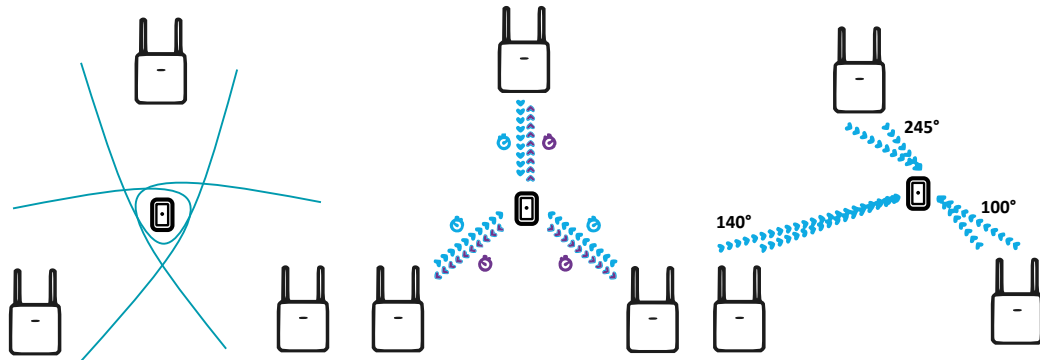
Phase Difference of Arrival

Nejedná sa priamo o lokalizačnú metódu, keďže nedokáže určiť vzdialenosť objektu. Kotva môže mať hardvérovo implementovaných až niekoľko antén a na základe rozdielu fáz zachytených signálov je možné vypočítať uhol dopadu. Táto metóda sa väčšinou používa v kombinácii s vyššie spomínanými metódami. Pri použití TWR (Two Way Ranging) je tak

⁵RTLS – Real-time locating system (lokalizačný systém pracujúci v reálnom čase) je systém, ktorý meria pozíciu zariadení s veľmi nízkym oneskorením.

⁶Oscilátor – Elektronická súčiastka alebo obvod, ktorý generuje impulzy s určitou frekvenciou.

možné už pomocou jednej kotvy zistiť približnú pozíciu značky. V prípade použitia TDoA (Time Difference of Arrival) táto metóda slúži ako ďalší mechanizmus zvýšenia presnosti a odolnosti voči odrazeným signálom.



Obr. 2.3: Metódy merania zľava doprava: TDoA (Time Difference of Arrival), TWR (Two Way Ranging), PDoA (Phase Difference of Arrival). Prevzaté z kinexon.com.

2.4 DW1000

DW1000[3] od firmy Decawave⁷ je prvý komerčne dostupný vysielač-prijímač v integrovanom obvode pracujúci s technológiou UWB. Používa sa ako rozhranie medzi MCU a fyzickou vrstvou. Tento obvod je ovládaný po zbernici SPI⁸, čo vytvára určité časové oneskorenie pri čítaní dát. Medzi hlavné použitia patrí komunikácia s minimálnou spotrebou energie a možnosť lokalizácie. Zároveň poskytuje množstvo diagnostických údajov pre účely následnej klasifikácie.

Počet prijatých symbolov

Každá správa obsahuje preamble, SFD⁹, PHR¹⁰ a samotné dáta. Preamble sa skladá z jednotlivých symbolov, má nastaviteľnú dĺžku a slúži k detekcii začiatku vysielania. Symbol je zhuk UWB pulzov, definovaný podľa IEEE 802.15.4a [2]. Jeden symbol nesie 1 až 2 bity informácie (závisle na implementácii). Šírka symbolu definuje prenosovú rýchlosť, ale v rámci preamble je konštanta, menovite 992ns pri 16MHz PRF¹¹ a 1016ns pri 64MHz PRF. Symboly v rámci preamble sú taktiež definované podľa [2] a majú vlastnosť perfektnej periodickej autokorelácie¹², čo umožňuje prijímaču určiť presnú impulznú odozvu.

Tento parameter následne vyjadruje, koľko symbolov z preamble bolo správne prijatých. Komunikujúce zariadenia by mali mať prednastavenú rovnakú dĺžku preamble na

⁷Decawave – Vid www.decawave.com.

⁸SPI – Serial Peripheral Interface (sériové periférne rozhranie) je rozhranie používané pre komunikáciu s perifériami (inými integrovanými obvodmi) na krátke vzdialenosti.

⁹SFD – Start of Frame Delimiter (oddelovač rámca) je sekvencia symbolov oddeľujúca preamble a rámec.

¹⁰PHR – PHY Header (hlavička pridaná fyzickým rozhraním) je sekcia obsahujúca charakteristiky prenášaných dát.

¹¹PRF – Pulse Repetition Frequency (frekvencia opakovania pulzov).

¹²Autokorelácia – Miera značiaca nakoľko časová postupnosť koreluje sama so sebou s určitým časovým oneskorením.

základe typu použitia komunikácie. Nižšie hodnoty znižujú spotrebu, vyššie zasa zvyšujú spoľahlivosť a dosah.

Impulzná odozva

Reprezentuje odozvu kanálu na krátky impulz. Jedná sa o pole vzoriek, kde každá obsahuje imaginárnu a reálnu časť signálu. Počet vzoriek závisí na prednastavenej šírke symbolu, každá vzorka reprezentuje 1 ns. Dáta v rámci poľa sú akumulované, každý prijatý symbol mení hodnotu jednotlivých vzoriek. V rámci krivky tvorenej týmito dátovými bodmi je následne možné pozorovať napríklad odrazené komponenty signálu.

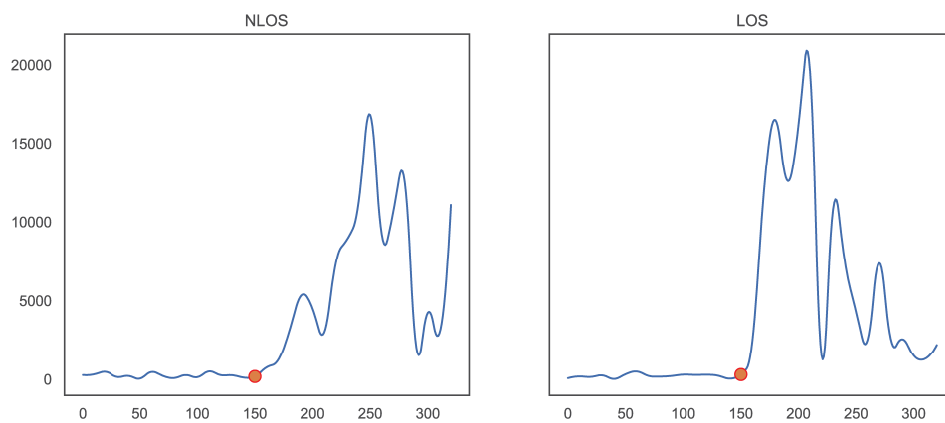
Úroveň hrany

DW1000 obsahuje proprietárny LDE¹³ algoritmus na detekciu indexu prvej vzostupnej hrany prijatého signálu na základe impulznej odozvy kanálu, viď obrázok 2.4. Vďaka tomu je možné presne určiť a dopočítať čas, kedy bol signál prijatý. Tento parameter teda predstavuje hodnotu úrovne impulznej odozvy, kde algoritmus detegoval nástupnú hranu prichádzajúceho signálu.

Odvožené parametre

Keďže celá impulzná odozva pozostáva zhruba z 32kB dát a je možné ju stiahnuť len celú naraz, jej prenos po SPI zbernici je časovo náročný, čo je pri reálnom nasadení problematické. Z toho dôvodu časti impulznej odozvy alebo odvožené parametre sa nachádzajú priamo v rýchlo dostupných registroch. Sú to napríklad prvé tri vzorky impulznej odozvy kanálu po detekcii vzostupnej hrany, energia prijatého signálu, index maximálnej amplitúdy, index prvej vzostupnej hrany signálu a štandardná odchýlka šumu.

¹³LDE – Leading Edge Detection (detekcia vzostupnej hrany).



Obr. 2.4: Impulzná odozva kanálu s vyznačenou úrovňou vzostupnej hrany prvého prijatého signálu, ktorá bola detegovaná pomocou LDE algoritmu. Signál je interpolovaný pre lepšiu vizualizáciu, vzorkovanie časovej osi je 100 ps. Signál vľavo bol utlmený prekážkou (NLOS – Non Line of Sight), signál vpravo je priamy (LOS – Line of Sight). Vyobrazené impulzné odozvy pochádzajú z použitého datasetu a predstavujú ideálne vzorky, ktoré vhodne reprezentujú ideálny prípad.

Kapitola 3

Klasifikácia a existujúce riešenia

Na začiatku tejto kapitoly stručne vysvetlím princíp funkcie klasifikačných metód, ktoré budú neskôr používané. Následne popíšem použitý dataset a jeho vlastnosti. V rámci hľadania najvhodnejšieho riešenia overím výsledky štúdií, ktoré sa zaoberali klasifikáciou LOS a NLOS signálov pri použití technológie UWB.

3.1 Klasifikačné metódy

Klasifikácia je proces triedenia dát do vopred definovaných tried. Cieľom tejto práce je odlíšiť LOS a NLOS signály, teda existenciu prekážky medzi vysielačom a prijímačom. Tým pádom sa budem zaoberať binárnou klasifikáciou, t.j. klasifikáciou do dvoch tried. V tejto kapitole popíšem klasifikačné techniky založené na strojovom učení, ktoré budú použité v nasledujúcich častiach práce.

Multi Layer Perceptron

MLP (Multi layer perceptron) je plne prepojená dopredná neurónová sieť. Skladá sa zo vstupnej, výstupnej a aspoň jednej skrytej vrstvy. Každá vrstva pozostáva z niekoľkých uzlov – neurónov, ktoré mapujú svoje vstupy na jeden výstup pomocou aktivačnej funkcie. Toto mapovanie je spropagované až na výstup celej neurónovej siete. Vstup neurónu pozostáva z vektoru výstupov predchádzajúcej vrstvy, ktorý je vynásobený váhou a ovplyvnený cez bias¹. Medzi aktivačné funkcie patrí napríklad ReLU², logistický sigmoid alebo hyperbolický tangens. ReLU je typicky používaná v skrytých vrstvách a logistický sigmoid na výstupe. Podrobnejšie v [13] a [5].

Convolutional Neural Network

CNN (konvolučná neurónová sieť) je druhom MLP. Jej zmyslom je vylepšenie úspešnosti pri klasifikácii série dát, ktorej dáta sú na seba naviazané v určitom priestore. Jedná sa napríklad o časovú postupnosť (1D), alebo obrázky (2D). CNN obsahuje rovnako ako MLP niekoľko vrstiev. Prvé vrstvy sú zpravidla konvolyčné. Sekvencia je prechádzaná oknom, ktoré na základe konvolyčného jadra³ vypočíta výsledok. Novo vzniknutá séria je odovzdaná

¹Bias – Hodnota pričítaná k vektoru vstupov neurónu.

²ReLU – Rectified Linear Unit je aktivačná funkcia neurónu $f(x) = \max(0, x)$.

³Konvolyčné jadro – Vektor alebo matica hodnôt, ktorá je určitou matematickou operáciou aplikovaná na časť datovej sekvencie.

ďalšej vrstve. Na konci konvolučných vrstiev sa nachádza MLP klasifikátor, ktorého vstup je jednorozmerná séria dát a klasifikácia prebieha rovnako ako v predchádzajúcej sekcii. Podrobnejšie v [6].

Decision Tree

Rozhodovacie stromy sa snažia hierarchicky rozdeliť dáta do formy binárnych stromov. Každý uzol stromu obsahuje podmienku pre jeden alebo viac parametrov a každá vetva predstavuje pravdepodobnosť, s ktorou daný parameter spadá do vopred daných tried. Listové uzly následne obsahujú finálnu pravdepodobnosť klasifikácie. Podrobnejšie v [5].

Medzi varianty klasifikačných stromov patrí napríklad Random Forest alebo stromy používajúce „gradient boosting“ – postupné vylepšovanie. Random Forest je zoskupenie stromov, teda les, kde každý strom vyhodnocuje klasifikované dáta. Výsledná pravdepodobnosť je daná hlasovaním (priemerom) predikcií individuálnych stromov. Pri vytváraní stromov je použitá náhodná časť datasetu, čo čiastočne zamedzuje pretrénovaniu klasifikačnej metódy. Všetky stromy sú však vytvárané rovnocenne.

Gradient Boosted Classifier (GB tree) je metóda, ktorej stromy sú trénované iteratívne. Každú iteráciu je vytvorený nový strom, ktorý dopĺňa ten predchádzajúci. Ich úspešnosť je vyhodnocovaná pomocou predom zvoleného kritéria.

Naive bayes

Rýchla propabilistická metóda založená na použití bayesoveho teorému, predikuje aposteriornu pravdepodobnosť, ktorá vyjadruje či vzorka bude patriť do určitej triedy. Aby sa dosiahlo najlepších výsledkov, všetky vstupné parametre by mali byť na sebe štatisticky nezávislé. Zároveň táto metóda očakáva len diskrétna dáta, spojité musia byť predom upravené. Existuje viacero variant bayesovských klasifikátorov, na základe predpokladaného rozloženia hodnôt v rámci datasetu. Menovite je to napríklad gaussovský, multinomiálny alebo bernouliho naive bayes. Podrobnejšie v [8] a [5].

Support vector machine

Metóda, ktorá sa snaží lineárne separovať jednotlivé vzorky tak, aby bol maximalizovaný „margin“, t.j. vzdialenosť medzi vektorom separujúcim dáta a jednotlivými dátovými bodmi. Pomocou transformácie dát použitím jadrovej funkcie je možné separovať aj také dátové body, ktoré nie sú na prvý pohľad lineárne oddeliteľné. Tento typ klasifikátoru má typicky zvýšenú odolnosť voči pretrénovaniu, ale jeho komplexita prudko rastie s počtom vzoriek. Podrobnejšie v [8] a [5].

3.2 Metodika evaluácie klasifikačných metód

Jednoduché vyhodnotenie úspešnosti klasifikačnej metódy na základe pomeru počtu úspešne vyhodnotených vzoriek a všetkých vzoriek nemusí presne popisovať binárny klasifikátor. Z toho dôvodu sú zavedené nasledovné metriky a pojmy, ktoré budú využívané v tejto práci:

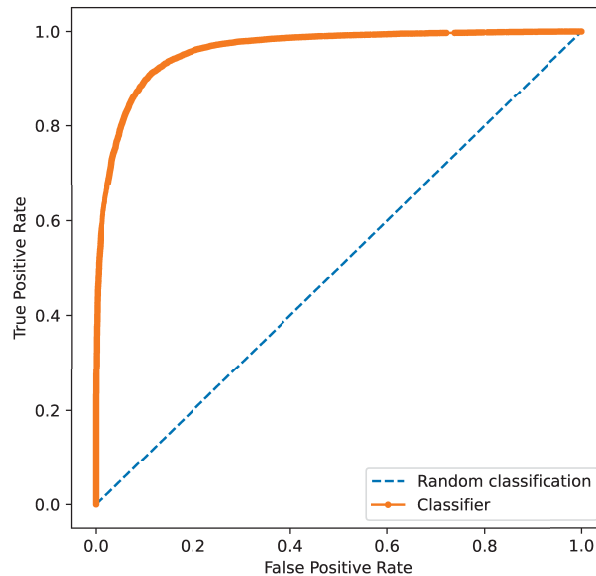
- **Úspešnosť** – Pomer počtu správne vyhodnotených vzoriek a počtu všetkých vzoriek.
- **FP, FN** – False Positive, False Negative (falošné pozitívum, falošné negatívum), označujú vzorky, ktoré boli zaradené do opačnej triedy.

- **TP, TN** – True Positive, True Negative (skutočné pozitívum, skutočné negatívum), označujú správne vyhodnotenú triedu jednotlivých vzoriek.
- **Úplnosť** (TPR – True positive rate) určuje pomer správne vyhodnotených LOS signálov a všetkých LOS signálov v datasete, viď rovnica 3.1. Opakom TPR je FPR – False Positive Rate, viď rovnica 3.2.
- **Selektívnosť** (TNR – True negative rate) určuje pomer správne vyhodnotených NLOS signálov a všetkých NLOS signálov v datasete. Opakom TNR je FNR – False Negative Rate.
- **ROC** – Receiver Operating Characteristic (Operačná krivka) je krivka, ktorá ukazuje pomer medzi TPR a FPR pri rôznych nastaveniach rozhodovacej úrovne, viď obrázok 3.1. Táto krivka zároveň umožňuje určenie najideálnejšej rozhodovacej hranice, ktorá umožňuje najpresnejšie triedenie vzoriek pomocou daného klasifikátoru.
- **ROC AUC** – area under the ROC curve (plocha pod krivkou ROC) je metrika vyjadrujúca kvalitu klasifikačnej metódy. Jej použitie je ideálne pokiaľ klasifikačná metóda udáva pravdepodobnosť jednotlivých tried. V takom prípade je nutná voľba rozhodovacej úrovne, čo môže ovplyvniť úspešnosť klasifikácie. Týmto je napríklad najviac ovplyvnené $F1^4$ skóre, keďže pracuje len s počtom úspešne a neúspešne klasifikovaných tried.

$$TPR = \frac{TP}{TP + FN} \quad (3.1)$$

$$FPR = 1 - TNR \quad (3.2)$$

⁴F1 skóre – Harmonický priemer medzi presnosťou (pomer počtu korektne vyhodnotených LOS signálov a všetkých signálov, ktoré boli vyhodnotené ako LOS) a úplnosťou (TPR).



Obr. 3.1: Príklad ROC krivky pre klasifikátor XGBoost⁶, ktorý je natrénovaný na použitom datasete, viď nasledujúca sekcia. Jeho úspešnosť dosahuje 89%, TNR 87%, TPR 90% a ROC AUC skóre 0.95. Pri výpočte tejto krivky bolo zistené, že ideálna rozhodovacia úroveň pre klasifikáciu je 0.57. Prerušovaná čiara znázorňuje prípad, v ktorom sú triedy náhodne typované a klasifikačný algoritmus nemá žiadnu schopnosť predpovedať triedy.

3.3 Dataset

Dáta použité na tréning a klasifikáciu boli získané z dvoch nezávislých zdrojov. Prvá časť datasetu bola dodaná firmou Sewio Networks s.r.o.. Tento dataset bol získaný meraním v industriálnom prostredí, obsahuje vzorky ako pohybujúcich sa objektov, tak aj stacionárnych. Zber dát prebiehal metódou TDoA, viď 2.3, čiže neprebíhala žiadna spätná komunikácia z kotvy na značku a dataset neobsahuje meranie vzdialenosti. Konfigurácia kotvy je nasledovná:

- **PRF** 64 MHz
- **Rýchlosť prenosu** 6.8 Mbps
- **Dĺžka preamble** 1024 symbolov

Druhá časť datasetu⁷ bola prevzatá zo štúdie [9]. Merania tohto datasetu boli vykonané v neindustriálnych priestoroch, ale bolo využitých celkovo 7 odlišných miestností. V tomto datasete sa nachádzajú len statické dáta, značka nebola pri meraní v pohybe. Zber dát prebiehal metódou TWR, tým pádom je dostupná aj zmeraná vzdialenosť medzi kotvou a značkou. Konfigurácia kotvy je nasledovná:

⁶XGBoost – Optimalizovaná knižnica, ktorá obsahuje klasifikačnú metódu založenú na postupnom zlepšovaní rozhodovacích stromov (Gradient boosting). Viď xgboost.readthedocs.io/en/stable/.

⁷Dataset dostupný z github.com/ewine-project/UWB-LOS-NLOS-Data-Set.

- **PRF** 64 MHz
- **Rýchlosť prenosu** 110 kbps
- **Dĺžka preamble** 1024 alebo 1536 symbolov

Všetky záznamy boli ručne označené ako LOS signály alebo NLOS signály na základe viditeľnej existencie priamej cesty medzi kotvou a značkou. Celkovo je použitých viac ako 100 000 záznamov, čo by malo zaručiť dostatočnú rozmanitosť dát. Pred použitím sú odstránené všetky záznamy s chybnými hodnotami, napríklad záznamy s nulovým počtom prijatých symbolov, keďže by mohli komplikovať tréning modelu a výpočet odvodených parametrov.

3.4 Existujúce riešenia

V posledných rokoch bolo publikovaných veľa štúdií, ktoré sa zaoberali klasifikáciou UWB signálov. Zároveň sa o rôznych technikách rozlíšenia LOS a NLOS signálov píše v dokumentáciách integrovaného obvodu DW1000 alebo v knihách o UWB technológii. V tejto časti ich popíšem a pokúsím sa replikovať výsledky na vlastnom datasete.

Technická dokumentácia DW1000

Dokumentácia integrovaného obvodu DW1000 navrhuje použitie jednoduchého pravidla na klasifikáciu signálov, viď [3]. Používa dva parametre: odhadovanú silu prijatého signálu a odhadovanú silu prvého prijatého signálu.

$$RXpower = 10 \log_{10} \left(\frac{C \cdot 2^{17}}{N^2} \right) - A \quad [\text{dBm}] \quad (3.3)$$

$$FPpower = 10 \log_{10} \left(\frac{F_1 + F_2 + F_3}{N^2} \right) - A \quad [\text{dBm}] \quad (3.4)$$

Kde:

- C je energia prijatého signálu.
- N je počet prijatých symbolov.
- A je konštanta závislá na PRF.
- F_1, F_2, F_3 , sú prvé tri hodnoty impulznej odozvy kanálu po detekcii vzostupnej hrany.

Klasifikácia ďalej pozostáva z odčítania $FPpower$ od $RXpower$. Pokiaľ je rozdiel menší ako 6 dB, jedná sa o LOS signál. Pokiaľ je väčší ako 10 dB, signál je NLOS. Pri mojom testovaní som zistil, že táto metóda je síce najrýchlejšia, ale dosahuje úspešnosť len 64% na celom datasete.

Úspešnosť	TPR	TNR	ROC AUC
0.64	0.85	0.34	0.58

Tabuľka 3.1: Vyhodnotenie metódy klasifikácie podľa [3] na vlastnom datasete.

Identification and Mitigation of NLOS based on Channel Information Rules for Indoor UWB Localization

Podobný prístup založený na pravidlách bol navrhnutý v práci [20]. Pri klasifikácii boli použité nasledovné parametre:

- $RXpower$ – hodnota, ktorá bola vypočítaná pomocou rovnice (3.3).
- Čas vzostupu signálu – doba, za ktorú signál dosiahol svoju maximálnu úroveň.

Pre tieto parametre boli zvolené hraničné hodnoty, ktoré rozdeľovali signály na priame a utlmené. Pri mojom testovaní táto metóda mala rovnakú úspešnosť ako predchádzajúce riešenie z technickej dokumentácie, bola dosiahnutá úspešnosť 64%.

Úspešnosť	TPR	TNR	ROC AUC
0.64	0.73	0.51	0.62

Tabuľka 3.2: Vyhodnotenie metódy klasifikácie podľa [20] na vlastnom datasete.

UWB Theory and Applications

Na konci tejto knihy autori navrhujú rozlíšenie signálov na základe jednotlivých amplitúd impulznej odozvy [17]. Následne pokiaľ je signál klasifikovaný ako NLOS, uvádzajú možnosť výpočtu dôveryhodnosti daného signálu, ktorá môže byť použitá pri konečnej lokalizácii.

Podstatou riešenia je vyhľadanie jednotlivých vrcholov impulznej odozvy. Pokiaľ je prvý vrchol najväčší, tak signál je klasifikovaný ako LOS. V opačnom prípade priama cesta bola zatienená a signál je klasifikovaný ako NLOS.

Pri mojom testovaní táto metóda dosiahla podobnú úspešnosť ako predchádzajúce dve metódy, a síce 63%. Pri šírení signálu môže nastať pozitívna interferencia, ktorá zosilní amplitúdu odrazeného signálu až nad úroveň amplitúdy priameho signálu. Z toho dôvodu mnoho signálov nie je korektne klasifikovaných.

Úspešnosť	TPR	TNR	ROC AUC
0.64	0.74	0.49	0.63

Tabuľka 3.3: Vyhodnotenie metódy klasifikácie podľa [17] na vlastnom datasete.

NLOS Channel Detection with Multilayer Perceptron in Low-Rate Personal Area Networks for Indoor Localization Accuracy Improvement

Klasifikácia v rámci tejto práce bola realizovaná pomocou MLP, viď [10]. Použité parametre zahŕňajú $RXpower$, $FPpower$ a štandardnú odchýlku šumu. Všetky použité parametre je možné získať bez zdĺhavého prístupu k impulznej odozve kanálu. Dataset bol meraný TWR metódou, pozostáva z 36 lokalít. Keďže autori ho zverejnili, bude používaný aj v mojej práci. Použitý MLP sa skladá z 3 vrstiev s relatívne malým počtom neurónov (5, 12 a 3), čo je jednoducho realizovateľné aj pri klasifikácii priamo vo vstavanom systéme. Autori práce dosiahli pri klasifikácii úspešnosť až 90%.

Pri vlastnom testovaní sa mi však podarilo dosiahnuť úspešnosť 85% na nevidených dátach. Keďže na overenie funkčnosti použitého klasifikátora nebola použitá krížová validá-

cia⁸, tento pokles môže byť spôsobený buď rozdielnymi tréningovými a testovacími dátami, alebo bol použitý rozdielny random seed⁹ pri vytváraní MLP.

Úspešnosť	TPR	TNR	ROC AUC
0.85	0.84	0.86	0.92

Tabuľka 3.4: Vyhodnotenie metódy klasifikácie podľa [10] na vlastnom datasete.

NLOS Detection and Mitigation in Differential Localization Topologies Based on UWB Devices

Štúdia [21] sa zameriava na detekciu LOS a NLOS signálov pri použití TDoA techniky lokalizácie objektu. Hovorí o tom, že pri prijatí nepriameho signálu môže nastať ako pozitívna, tak aj negatívna chyba (TDoA meria pozíciu v priestore na rozdiel od TWR), čo nebolo zvažované pri ostatných štúdiách založených na TWR technike lokalizácie. Použitý dataset pozostáva zo 100 000 vzoriek a bol meraný na viacerých miestach. Vzorky, ktoré vykazovali chybu merania väčšiu ako 30 cm boli klasifikované ako NLOS signály, v opačnom prípade LOS. Parametre použité v štúdiu boli nasledovné:

- $RXpower$ – vid' (3.3)
- $FPpower$ – vid' (3.4)
- Najväčšia energia signálu v bode
- Čas vzostupu signálu
- Rozdiel medzi $RXpower$ a $FPpower$
- Úroveň šumu
- Koeficient asymetrie¹⁰ vypočítaný z prvých troch amplitúd prvého prijatého signálu
- Pomer maximálnej hodnoty signálu a priemeru prvých troch amplitúd prvého prijatého signálu
- Pomer $RXpower$ a šumu

Klasifikácia je realizovaná probabilistickými metódami, dosahuje úspešnosť až 86%. Sú testované 2 prístupy, ktoré dosahujú podobné výsledky. V prvom prípade je použitý Naive Bayes na výpočet aposteriórnej pravdepodobnosti LOS a NLOS signálu na základe každého parametra individuálne, následne sú pravdepodobnosti skombinované. Druhý prístup využíva „thresholding“ – klasifikáciu na základe hraničných hodnôt.

⁸Křížová validácia (cross-validation) je proces, pri ktorom je klasifikátor iteratívne trénovaný a testovaný na rôznych častiach datasetu.

⁹Random seed – Číslo, ktoré sa používa pri nastavení počiatočného stavu náhodného generátora. Náhodné generátory, ktoré majú rovnaký random seed budú generovať rovnaké čísla.

¹⁰Koeficient asymetrie – Hodnota vyjadrujúca šikmosť pravdepodobnostného rozdelenia.

UWB NLOS/LOS Classification Using Deep Learning Method

V tejto štúdii autori testujú klasifikáciu na základe impulznej odozvy kanálu [14]. Zvolený prístup síce nie je vhodný pri implementácii do vstavaného systému, ale reprezentuje odlišné riešenie problému. Na detekciu LOS a NLOS signálov je otestovaná kombinácia CNN¹¹ a RNN¹². Úspešnosť dosiahla až 82%, použitie RNN podľa priložených výsledkov je však zanedbateľné. Túto úspešnosť sa mi podarilo overiť aj na vlastnom datasete.

Úspešnosť	TPR	TNR	ROC AUC
0.83	0.84	0.82	0.89

Tabuľka 3.5: Vyhodnotenie metódy klasifikácie podľa [14] na vlastnom datasete.

A decision tree-based NLOS detection method for the UWB indoor location tracking accuracy improvement

Štúdia [16] skúma použitie rozhodovacieho stromu na klasifikáciu NLOS signálov. Použitý dataset obsahuje vzorky z viacerých lokalít, je použitá TWR metóda merania vzdialenosti. Použité parametre zahŕňajú:

- Štandardná odchýlka šumu
- Energia prijatého signálu
- Index prvého prijatého signálu v rámci impulznej odozvy
- Prvé tri amplitúdy prvého prijatého signálu
- Počet prijatých symbolov
- Čas letu signálu

Pri klasifikácii boli pridané aj niektoré statické parametre konfigurácie kotvy, napríklad PRF, číslo kanálu, maximálna dĺžka preamble a podobne. Úspešnosť klasifikácie dosahuje až 90% a výsledky boli overené krížovou validáciou. Použitý klasifikátor bol dokonca úspešne nasadený pri použití existujúceho lokalizačného systému.

¹¹CNN – Convolutional Neural Network (konvolučná neurónová sieť).

¹²RNN – Recurrent Neural Network (rekurentná neurónová sieť) je používaná na klasifikáciu časových postupností, pri klasifikácii berie do úvahy aj predchádzajúci stav.

Kapitola 4

Návrh riešenia

Na začiatku kapitoly podrobnejšie popíšem cieľ tejto práce. Na základe získaných znalostí z predchádzajúcej kapitoly sa ďalej budem venovať analýze a návrhu. Analyzujem parametre získané z fyzického rozhrania, ich použiteľnosť a dôležitosť. Predchádzajúce štúdie používali veľa parametrov, no ich výsledky naznačujú, že nie všetky sú potrebné pre úspešnú klasifikáciu. Na koniec tejto kapitoly si vyberiem finálny klasifikačný model a odôvodním moje rozhodnutie.

4.1 Cieľ práce

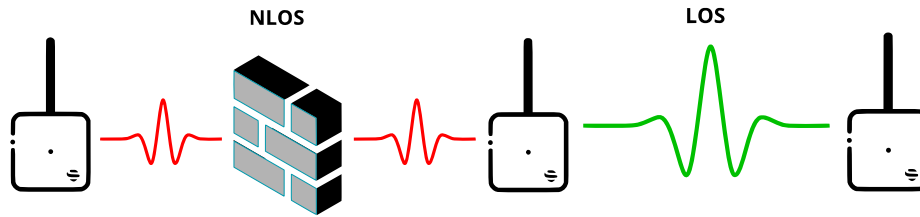
Samotný názov práce napovedá, že sa jedná o klasifikáciu priameho a odrazeného signálu. Táto definícia však môže byť trochu zavádzajúca. Ak vysielač vyšle jeden signál, prijímač ich môže detegovať až niekoľko kvôli odrazom v prostredí. V rámci klasifikácie je však podstatný len ten prvý prijatý signál, ostatné sú implicitne odrazené a tým pádom ignorované. Následne pri zbere dát je komplikované presne určiť, či prvý prijatý signál je elektromagnetická vlna, ktorá sa naozaj odrazila. Z toho dôvodu sa v technickej literatúre používajú termíny LOS¹ a NLOS², viď obrázok 4.1. Pri klasifikácii sa teda bude jednať o zisťovanie, či existuje priama viditeľnosť medzi vysielačom a prijímačom, alebo sa v ceste nachádza prekážka schopná dostatočne utlmiť signál.

Klasifikácia bude prebiehať priamo vo vstavanom systéme. Natrénovaný a validovaný klasifikačný model bude nutné exportovať v takej forme, aby bol ľahko presunutelný do vstavaného systému – kotvy. Kotva neobsahuje žiadnu externú pamäť, model bude umiestnený do flash pamäti³ MCU, takže bude musieť byť dostatočne kompaktný. V poslednom kroku bude potrebné vytvoriť knižnicu pre jazyk C, ktorá dokáže interpretovať klasifikačný model a klasifikovať dáta v reálnom čase.

¹LOS – Line Of Sight (priama viditeľnosť).

²NLOS – Non Line Of Sight (bez priamej viditeľnosti).

³Flash pamäť – Polovodičová pamäť s možnosťou opakovaného zápisu. Informácie sú uložené permanentne, teda nie je nutné napájanie na ich uchovanie.



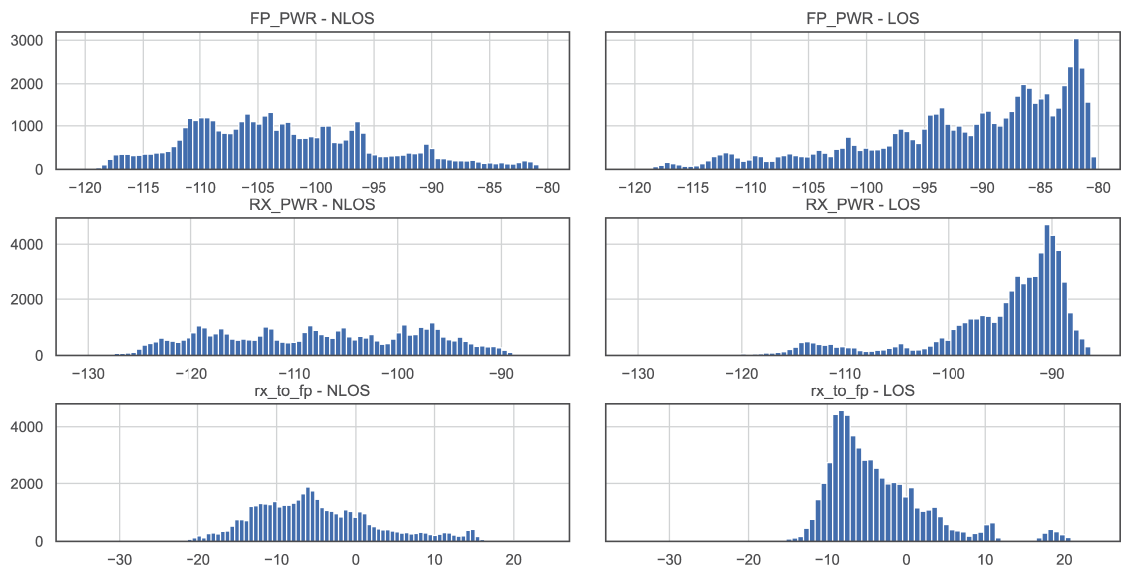
Obr. 4.1: Príklad LOS a NLOS podľa technickej dokumentácie firmy Sewio Networks s.r.o..
 Prevzaté z docs.sewio.net/docs/los-vs-nlos-25593229.html.

4.2 Analýza dát

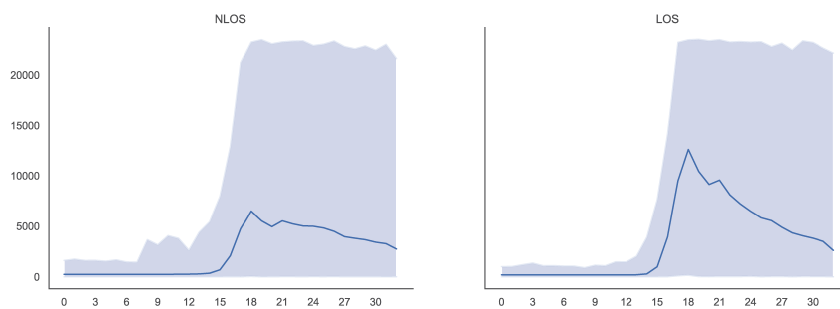
Analýza dát môže uľahčiť výber modelu. Niektoré typy klasifikačných metód fungujú lepšie na lineárne oddeliteľných dátach, ako napríklad SVM. Klasifikačné metódy založené na pravdepodobnosti zasa očakávajú nezávislosť parametrov. V ideálnom prípade by dáta mali byť aspoň čiastočne separovateľné.

Rozloženie parametrov som preskúmal pomocou histogramov. Skúmané parametre boli získané aj z registrov DW1000, aj vypočítané na základe existujúcich štúdií, viď 3.4. Ani jeden parameter však nedokázal perfektne oddeliť dáta, oba typy signálov nadobúdali rovnaké hodnoty. Podľa [21] parametre $RXpower$, $FPpower$ a ich rozdiel vykazujú najlepšiu úspešnosť pri klasifikácii použitím modelu Naive Bayes. Tieto parametre ukazujú zároveň najväčší rozdiel rozloženia hodnôt na histograme, viď obrázok 4.2.

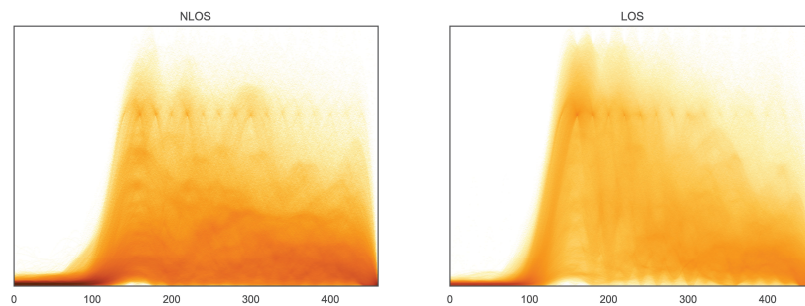
Aj napriek tomu, že by bolo ideálne vyhnúť sa použitiu impulznej odozvy kanálu, preskúmal som aj jej rozdiely. Ideálny príklad impulznej odozvy kanálu, viď obrázok 2.4, nie je typický. Vzorok datasetu častokrát obsahujú impulzné odozvy, z ktorých nie je ani na prvý pohľad možné rozpoznať jej typ. Z toho dôvodu som sa pozrel ako na minimálne a maximálne hodnoty jednotlivých impulzných odoziev, viď obrázok 4.3, tak aj na priemernú formu impulznej odozvy, viď obrázok 4.4. Pri malom množstve NLOS signálov je vidieť reálny utlmený prvý prijatý signál, ktorý bol zle detegovaný LDE algoritmom. Zároveň stredná hodnota LOS a NLOS signálov je na prvý pohľad odlišná. Signály však nie sú jednoznačne oddeliteľné, rovnako ako pri parametrovom prístupe. Z toho vyplýva, že použitie impulznej odozvy kanálu je nutné len pri detekcii prvej vzostupnej hrany prijatého signálu v prípade, že LDE algoritmus zlyhá.



Obr. 4.2: Príklad histogramov, ktoré popisujú rozloženie jednotlivých parametrov.



Obr. 4.3: Minimálna, maximálna a priemerná hodnota impulznej odozvy kanálu. Vzorkovanie časovej osi je 1 ns.



Obr. 4.4: Priemerný odtlačok všetkých impulzných odozvy v datasete. Jednotlivé impulzné odozvy boli interpolované pre lepšiu vizualizáciu. vzorkovanie časovej osi je 50 ps.

4.3 Výber klasifikačnej metódy

Analýzou, ktorá bol vykonaná v predchádzajúcich kapitolách, som ukázal, že klasifikácia na základe jednoduchých pravidiel má najnižšiu úspešnosť. Klasifikácia pomocou impulznej odozvy kanálu taktiež nevykazovala uspokojivé výsledky, z toho dôvodu sa zameriam na parametrický prístup.

Okrem klasifikačných metód, ktoré boli spomínané v štúdiách, vid 3.4., som otestoval aj metódy z balíčku scikit-learn⁴. Moje výsledky korelujú s výsledkami skúmaných štúdií a najlepšie z nich boli dosiahnuté klasifikačnými metódami, ktoré sú založené na rozhodovacích stromoch. Taktiež som otestoval MLP pomocou knižnice Keras⁵, ktorá umožňuje presnejší návrh vrstiev ako scikit-learn, napríklad pridaním dropout vrstiev⁶. Tento MLP dosahoval podobnú úspešnosť ako stromové metódy.

Klasifikačná metóda	Úspešnosť	TPR	TNR	ROC AUC	Čas
Decision Tree	0.84	0.85	0.81	0.83	0.01
Random Forest	0.89	0.92	0.86	0.95	0.18
Gradient Boosted tree	0.86	0.87	0.85	0.93	0.36
Histogram based GB tree	0.88	0.89	0.87	0.95	0.04
XGBoost	0.89	0.91	0.87	0.95	0.01
MLP (Keras)	0.90	0.91	0.88	0.95	3.1
MLP (scikit-learn)	0.85	0.84	0.86	0.92	1.2
SVM (RBF kernel)	0.82	0.83	0.81	0.88	86
SVM (linear kernel)	0.79	0.77	0.82	0.86	16.4
Gaussian Naive Bayes	0.78	0.77	0.78	0.83	0.01

Tabuľka 4.1: Výsledky jednotlivých klasifikačných metód. Čas značí dobu klasifikácie tretiny datasetu. Táto hodnota je len orientačná, slúži na porovnanie rýchlostí jednotlivých metód. Ostatné parametre sú popísané v kapitole 3.2.

Na finálnu implementáciu som si zvolil metódu Random Forest. Táto metóda spolu s metódou XGBoost dosahovala najlepšie výsledky z vyhodnocovaných algoritmov. Zároveň je vhodnejšia na implementáciu a nasadenie ako MLP z nasledujúcich dôvodov:

- MLP klasifikátor má lepšiu úspešnosť pokiaľ sú vstupné dáta upravené do určitého konštantného rozsahu, typicky 0 až 1. Rozhodovacie stromy ale nepožadujú žiadnu úpravu dát, čo umožňuje rýchlejšie spracovanie.
- Pri komerčnom nasadení musí byť cieľové zariadenie riadne otestované a certifikované. Tento proces je však problematický pri niektorých klasifikačných algoritmoch, vrátane MLP, nakoľko sa správajú ako blackbox⁷. Váha neurónov MLP ovplyvňuje výsledok klasifikácie, ale nie je transparentné prečo má neurón určitú aktivačnú úroveň a prečo pri kombinácii určitých neurónov dostaneme špecifický výsledok. V prípade rozhodovacích stromov je však možné presne vypísať cestu, ktorou bol dosiahnutý výsledok na základe rozhodovacej úrovne každého z uzlov.

⁴scikit-learn – Knižnica pre programovací jazyk Python, poskytuje veľké množstvo klasifikačných algoritmov, funkcií na úpravu dát, diagnostiku a podobne. Vid scikit-learn.org.

⁵Keras - Knižnica pre programovací jazyk Python, poskytuje veľké množstvo modelov strojového učenia.

⁶Dropout vrstva – Vrstva MLP klasifikátoru, ktorá náhodne pozmení hodnoty. Touto technikou je možné zamedziť pretrénovaniu a dosiahnuť lepších výsledkov.

⁷Blackbox – Zariadenie alebo systém, ktorý mapuje svoje vstupy na výstupy, presný postup mapovania však nie je známy.

- Posledným dôvodom je rýchlosť a univerzálnosť. MLP vyžaduje vykonanie množstva matematických operácií k tomu, aby bol dosiahnutý výsledok. Pokiaľ nie je použitý procesor optimalizovaný pre tieto operácie, výpočet môže byť relatívne zdĺhavý. Klasifikácia rozhodovacími stromami vyžaduje len jednoduché porovnanie a aj ten najjednoduchší počítač zvládne takúto klasifikáciu pomerne rýchlo.

4.4 Výber parametrov

Analýza dát dokázala, že väčšina parametrov využitých na klasifikáciu je nejednoznačná, t.j. nevedia presne rozdeliť výsledok na 2 triedy. Tento fakt je možné vidieť aj na dôležitosti jednotlivých parametrov pri natrénovanom modeli, viď obrázok 4.5. Veľa zbytočných parametrov má negatívny vplyv na rýchlosť klasifikácie, či dokonca aj úspešnosť. Z toho dôvodu redukcia parametrov môže pomôcť celému klasifikačnému procesu. Zároveň je nutné eliminovať parametre, ktoré sú závislé na konfigurácii kotvy. Parametre ako napríklad energia impulznej odozvy, jej amplitúda, alebo počet prijatých symbolov sú závislé na dĺžke preambuly. Každý prijatý symbol zvyšuje hodnotu v akumulátore, ktorý reprezentuje impulznú odozvu kanálu. Pokiaľ je preambula kratšia, tieto hodnoty budú implicitne menšie. To by malo negatívny dopad na výsledok klasifikácie, keďže vybrané parametre by mali byť nezávislé na nastavení kotvy.

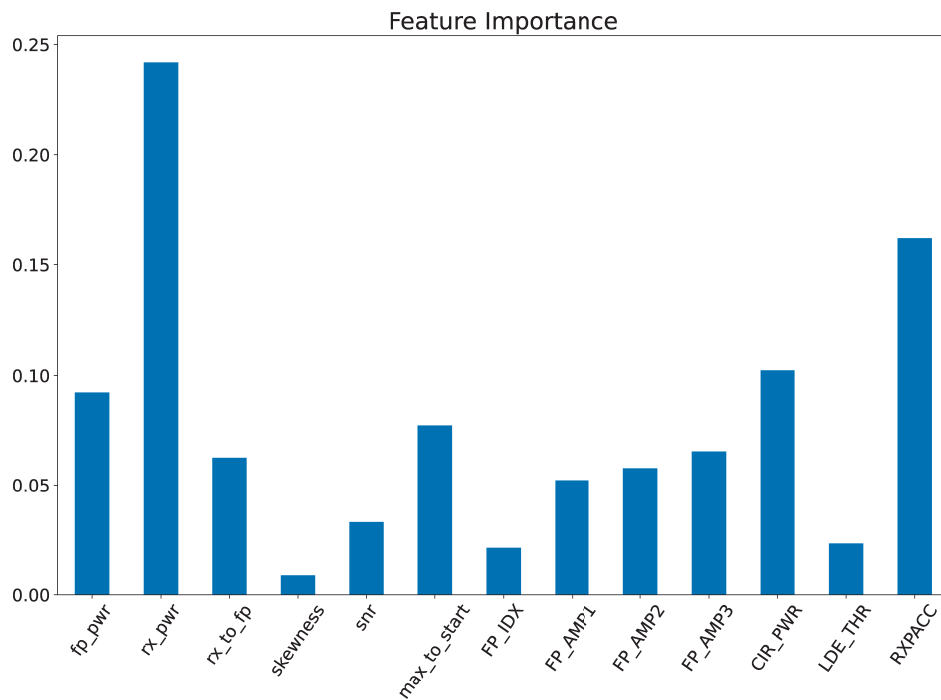
Redukcia parametrov bola vykonaná iteratívnym tréňovaním rozhodovacieho stromu. V každej iterácii boli získané váhy jednotlivých parametrov a ich vplyv na výsledok rozhodnutia⁸. Následne parameter s najnižším dopadom bol eliminovaný. Tento proces prebiehal dovtedy, až úspešnosť klasifikácie nezačala radikálne klesať. Pokiaľ parameter s nízkou dôležitosťou mal viditeľný vplyv na úspešnosť klasifikácie, teda bola znížená úspešnosť, bol ponechaný a iný parameter eliminovaný.

Postupnou elimináciou som zistil, že najväčší dopad na výsledok klasifikácie majú parametre *FPpower* a *RXpower*. Bez týchto parametrov klasifikácia dosahovala minimálnu úspešnosť. Pri ich použití však môže nastať potenciálny problém v prípade, že vysielač a prijímač budú ďaleko od seba, keďže ich hodnota nelineárne klesá so vzdialenosťou.

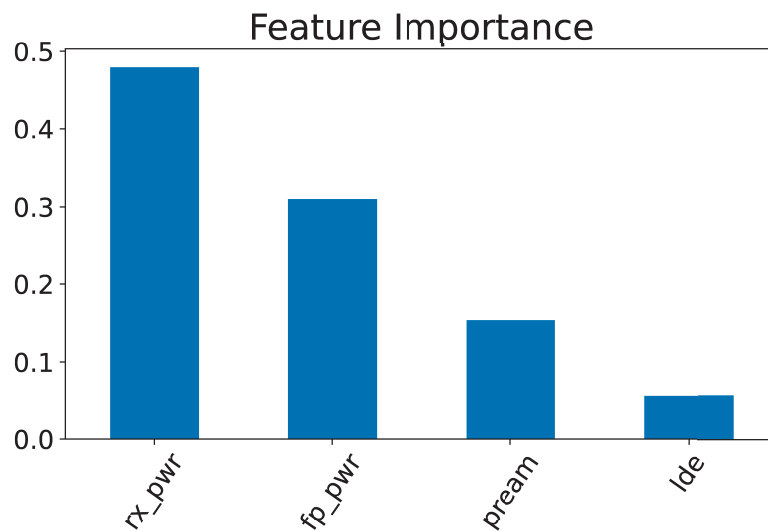
Na finálnu implementáciu som si vybral také parametre, aby boli nezávislé na konfigurácii kotvy. Ich dôležitosť je znázornená na obrázku 4.6. Zvolené parametre sú nasledujúce:

- *FPpower*, viď 3.4.
- *RXpower*, viď 3.3.
- Pomer počtu prijatých symbolov a dĺžky preambuly.
- Pomer detegovanej úrovne začiatku pulzu LDE algoritmom a dĺžky preambuly.

⁸Dôležitosť jednotlivých parametrov je hodnota, ktorá vyjadruje mieru zvýšenia úspešnosti klasifikácie v rámci jednotlivých uzlov všetkých stromov. Viď scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.



Obr. 4.5: Všetky použiteľné parametre a ich dôležitosť v rámci natrénovanej klasifikačnej metódy Random Forest. Dosažená úspešnosť tohto modelu je 89.8%.



Obr. 4.6: Vybrané parametre a ich dôležitosť. Klasifikačná metóda Random Forest bola natrénovaná s rovnakými nastaveniami ako pri obrázku 4.5. Úspešnosť tohto modelu je 88.7%.

Kapitola 5

Implementácia

V tejto kapitole popíšem finálny postup klasifikácie. Implementácia sa skladá z dvoch častí. Klasifikácia a vytvorenie modelu je implementovaná v jazyku Python3, nakoľko poskytuje najlepšie knižnice použiteľné pre klasifikáciu a prácu s dátami. Táto časť zároveň obsahuje načítanie datasetov, vypočítanie parametrov a export natrénovaného modelu. Druhú časť tvorí implementácia do vstavaného systému pomocou jazyka C. V tejto etape vytvorím knižnicu, ktorá dokáže pracovať s exportovaným modelom. Táto knižnica bude obsahovať funkcie pre klasifikáciu, meranie času klasifikácie a prácu s DW1000.

5.1 Klasifikačný model

Ako som spomínal vyššie, model bude implementovaný v jazyku Python3 a skladá sa z modulov `data_loader` a `train_model`.

Načítanie dát

Oba použité datasety majú odlišnú štruktúru a obsahujú iné parametre. Účelom modulu `data_loader` je zjednotiť všetky vzorky daných datasetov tak, aby sa s nimi dobre pracovalo. Datasety zároveň obsahujú vzorky, ktoré majú zlé hodnoty, napr. nulový počet prijatých symbolov. Takéto parametre by viedli k deleniu nulou pri výpočte odvodených parametrov, a preto musia byť odstránené.

Každý z datasetov bol získaný iným spôsobom a obsahuje iné parametre. Keďže tento modul je použiteľný aj na analýzu, vytvára štruktúru, ktorá obsahuje nasledujúce položky:

- Data – Parametre, ktoré sú rovnaké vo všetkých datasetoch.
- CIR – Impulzná odozva kanálu, ktorá obsahuje 15 vzoriek pred indexom prvej vzostupnej hrany prijatého signálu a 15 vzoriek po ňom.
- Others – Ďalšie parametre špecifické na danom datasete.

Vytvorenie modelu

Klasifikačný model Random Forest bol implementovaný pomocou knižnice `scikit-learn`. Pri tréňovaní bola použitá validácia `ShuffleSplit`. `ShuffleSplit` vytvorí niekoľko variant tréňovacích a validačných dát, na ktorých je následne testovaný model. Model je možné považovať

za stabilný, pokiaľ sa jeho úspešnosť razantne nezmení v žiadnom z delení. Stabilnosť je zároveň možné vylepšiť obmedzením počtu stromov, upravením minimálneho požadovaného počtu vzoriek pre vytvorenie nového uzlu a hĺbkou stromov.

Klasifikačné modely obsahujú množstvo parametrov, ktorými je možné vylepšiť úspešnosť klasifikácie. Pri klasifikácii metódou Random Forest platí nasledovné tvrdenie: Čím viac stromov a väčšia maximálna hĺbka, tým väčšia úspešnosť. Pri veľkom množstve stromov ale hrozí pretrénovanie, to znamená že model dokáže perfektne klasifikovať len dáta, ktoré videl alebo ich obdobu. Zároveň platí čím viac stromov a uzlov, tým viac miesta je potrebného na uloženie celého modelu. Z toho dôvodu na hľadanie najlepších parametrov som použil triedu `GridSearchCV`¹, taktiež z knižnice `scikit-learn`. Jedná sa o triedu, ktorá hľadá najlepšiu kombináciu parametrov pre zadanú klasifikačnú metódu. Na určenie skóre každej kombinácie som použil vlastnú funkciu, ktorá vyhodnotí úspešnosť klasifikácie na validačnom datasete len za predpokladu, že model nepresiahne určitú veľkosť. Vďaka tomu sa zamedzí vytváraniu veľkých modelov.

5.2 Export do jazyka C

Natrénovaný klasifikátor je následne nutné previesť do jazyka C, aby mohol byť implementovaný do MCU, ktorý ovláda kotvu. Existuje zopár voľne dostupných programov na export ako napríklad `sklearn-porter`² alebo `emlearn`³. Pri testovaní týchto nástrojov som však zistil, že ich funkcionálnosť nie je dostatočná. Nedokážu pracovať s pravdepodobnosťou v rámci listových uzlov stromu a celý strom exportujú priamo do kódu v podobe zrefazovaných podmienok. Tento prístup je síce najefektívnejší čo sa týka rýchlosti, ale zaberá zbytočne veľa miesta v pamäti vstavaného systému.

Preto som implementoval vlastný spôsob exportu do modulu `export_to_c`. Celý model je exportovaný do zdrojového súboru v jazyku C vo forme poľa štruktúr. Každá štruktúra predstavuje jeden uzol binárneho stromu. Jednotlivé položky by mali byť optimalizované tak, nech zaberajú čo najmenej miesta. Aby bolo možné navigovať binárny strom, každý uzol musí obsahovať minimálne nasledujúce parametre:

- Hraničná hodnota – Hodnota použitá pri porovnávaní hodnoty parametru. Určuje, ktorý synovský uzol bude následne vyhodnotený
- Označenie použitého parametru – Predstavuje parameter, napríklad pomocou indexu, pre ktorý platí hraničná hodnota.
- Pravdepodobnosť jednotlivých tried
- Odkaz na pravý a ľavý synovský uzol

Úprava parametrov

Vybrané parametre na klasifikáciu sú desatinné čísla, ktoré bývajú typicky reprezentované ako `float` o veľkosti 4B, alebo `double` o veľkosti 8B, podľa požadovanej presnosti. Nielenže pri veľkom množstve uzlov by parametre vo forme desatinných čísel zaberali veľa miesta, ale práca s desatinnými číslami je sama o sebe pomalá, hlavne pri použití vo vstavaných

¹`GridSearchCV` – Vid scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

²`sklearn-porter` – Vid github.com/nok/sklearn-porter.

³`emlearn` – Vid github.com/emlearn/emlearn.

systémoch. Z toho dôvodu na skomprimovanie všetkých dát som vytvoril niekoľko pravidiel a kompromisov. Desatinné čísla je možné nahradiť celými číslami pomocou mapovania rozsahu, alebo pomocou použitia fixnej desatinnej čiarky. Parametre boli teda upravené nasledovne:

$$RXpower_{adj} = 1000 \cdot \left(10 + \log_2 \left(\frac{C \cdot 2^{17}}{N^2} \right) \right) \quad (5.1)$$

$$FPpower_{adj} = 1000 \cdot \left(10 + \log_2 \left(\frac{F_1 + F_2 + F_3}{N^2} \right) \right) \quad (5.2)$$

$$Symbols = \frac{N}{PreambleLength} \cdot 60000 \quad (5.3)$$

$$Threshold = \frac{LDE}{PreambleLength} \cdot 1000 \quad (5.4)$$

Keďže pri použití rozhodovacieho prouku nie je nutné upravovať parametre tak, aby mali špeciálny rozmer (nemusí byť použitá štandardizovaná jednotka merania ako napríklad dBm), je možné ľubovoľne upraviť $RXpower$, viď 3.3, a $FPpower$, viď 3.4. Logaritmus použitý v rovnici slúži len na upravenie rozsahu premennej. Tým pádom dekadický logaritmus môže byť nahradený binárnym, keďže binárny logaritmus je jednoduchší na výpočet v systémoch, ktoré pracujú s binárnou reprezentáciou dát. Výsledok logaritmu je následne posunutý tak, aby sa zamedzilo negatívnym číslom. Desatinná čiarka je nakoniec posunutá o tri miesta, čo umožní reprezentáciu výsledku celým číslom bez väčšej straty presnosti.

Hodnota pomeru symbolov je mapovaná na rozsah hodnôt dvoj bytovej premennej. V prípade oneskorenej detekcie oddelovača rámcu, počet prijatých symbolov môže byť väčší ako reálny počet symbolov. Kvôli prípadom, kde počet prijatých symbolov je väčší ako dĺžka preamble, je ich pomer vynásobený o niečo menšou hodnotou ako 2^{16} . Situáciu kde je pomer prijatých symbolov väčší ako 1, je síce možné detegovať a upraviť ich hodnotu tak, aby zodpovedala reálnej hodnote. Pri mojom testovaní však väčšiu úspešnosť pri klasifikácii vykazovala neupravená hodnota.

Posledný parameter, ktorý reprezentuje pomer úrovne začiatku pulzu a dĺžky preamble, je už len posunutý o 3 desatinné miesta, aby sa zachovala dostatočná presnosť pri reprezentácii celým číslom. Vďaka týmto úpravám je šírka rozsahu hodnôt menšia ako 2^{16} , čo umožňuje použitie premennej o šírke 2B na reprezentáciu hraničnej hodnoty.

Reprezentácia pravdepodobnosti

Nakoľko nie je nutné poznať pravdepodobnosť s presnosťou na niekoľko desatinných miest, bude reprezentovaná ako celé číslo s fixnou desatinnou čiarkou. Keďže klasifikátor obsahuje len 2 triedy (NLOS a LOS), ich hodnoty sú komplementárne a nie je nutné ukladať každú z nich do jednotlivých uzlov. Z toho dôvodu výsledná pravdepodobnosť bude reprezentovať pozitívny výsledok, teda LOS signál.

Pravdepodobnosť jednotlivých tried je potrebná pre účely klasifikácie len v prípade ak sa jedná o listové uzly. Zároveň, listové uzly už nemajú žiadnu hraničnú hodnotu, tým pádom je možné tieto dva parametre zlúčiť. Pravdepodobnosť LOS signálu bude teda reprezentovaná celočíselnou premennou, ktorá je široká 2B, a jej hodnota bude posunutá o 3 desatinné miesta na zaručenie dostatočnej presnosti.

Použitý parameter

Na označenie parametrov, pre ktoré platí hraničná hodnota v jednotlivých uzloch, taktiež nie je treba veľa miesta. Keďže klasifikačný model počíta len so štyrmi parametrami, ich označenie sa zmestí do dvoch bitov.

Keďže nie je možné použiť premenných menších ako 1B, tento parameter bude zaberat presne 1B v štruktúre, ktorá reprezentuje uzol. Zvyšné bity tejto položky je možné využiť na ďalšie príznaky. Napríklad označenie, či uzol je konečný, teda či sa jedná o list stromu.

Navigácia stromu

Aby bola možná navigácia binárneho stromu, každý uzol musí obsahovať odkaz na svojich synov. Pri pozitívom vyhodnotení aktuálneho uzlu sa klasifikačný algoritmus posunie na ľavý synovský uzol, v opačnom prípade na pravý.

Vďaka optimalizácii reprezentácie binárneho stromu v pamäti je však možné dosiahnuť stav, kedy nebude nutná reprezentácia oboch indexov v každom uzle. Pokiaľ takýto strom bude reprezentovaný polom uzlov tak, aby ľavý synovský uzol sa nachádzal vždy bezprostredne za svojim otcovským uzlom, stačí si uchovať informáciu o pozícii pravého synovského uzlu. Za predpokladu, že celý les bude obsahovať menej ako 2^{16} uzlov, index pravého synovského uzlu je možné reprezentovať premennou o šírke 2B. V prípade, že by celý les obsahoval viac ako 2^{16} uzlov, táto položka môže predstavovať vzdialenosť. Následne sčítaním vzdialenosti a aktuálneho indexu by bol vypočítaný index pravého synovského uzlu.

Nakoniec je potrebné už len uchovať informáciu o pozícii jednotlivých stromov v rámci pola všetkých uzlov. To je dosiahnuté jednoduchým polom, ktoré obsahuje indexy ich koreňov. Pokiaľ celý les bude obsahovať menej ako 2^{16} uzlov, každý koreň bude reprezentovaný 2B premennou.

Export modelu

Na základe vyššie spomínaných špecifikácií, modul `export_to_c` dokáže exportovať natrénovaný model do jazyka C. Tento proces prebieha postupným extrahovaním jednotlivých stromov a uzlov z klasifikačného modelu, ktorý bol natrénovaný pomocou knižnice `scikit-learn`. Tento modul zároveň umožňuje meranie a výpis veľkosti celého modelu pred uložením, vďaka čomu je možné trénovať klasifikátor s obmedzením na veľkosť modelu.

Pred reálnym nasadením je vhodné overiť, či je exportovaný model totožný s natrénovaným modelom. Poslednou funkcionalitou modulu `export_to_c` je vykonanie klasifikácie dát rovnakým algoritmom, ako je použitý v jazyku C.

5.3 Knižnica pre kotvu

Táto knižnica je implementovaná v jazyku C a obsahuje algoritmus na klasifikáciu dát, výpočet parametrov a pomocné funkcie, napr. meranie času. Samotný klasifikačný algoritmus je nezávislý na platforme, niektoré pomocné funkcie sú však naprogramované špecificky pre MCU `TM4C1294KCPDT` od firmy Texas Instruments. Menovite sa knižnica skladá zo súborov `classifier`, `tree_data`, `tree_node` a `utils`.

Pomocné funkcie

Modul `utils` obsahuje dodatočné funkcie, ktoré nie sú priamo potrebné na klasifikáciu, ale umožňujú akceleráciu či diagnostiku. Funkcie `utils_tick_counter_start`, `utils_tick_counter_stop` a `utils_tick_counter_get` sú používané na evaluáciu klasifikačného algoritmu. Ovládajú interný časovač, ktorý je taktovaný na rovnakej frekvencii ako MCU. Tým pádom použitím týchto funkcií je možné presne zistiť počet inštrukčných cyklov, ktoré boli využité na klasifikáciu. Počet inštrukčných cyklov je lepšia metrika ako celkový čas klasifikácie, pretože je nezávislá od frekvencie taktovania procesoru.

Keďže kotva umožňuje nastavenie dĺžky preamble z webového rozhrania, funkcia `utils_get_preamble` vracia jej dĺžku na základe kódu z registra fyzického rozhrania DW1000.

Ďalšou funkciou tohto modulu je `utils_filter_mac`. V testovacom prostredí sa môže nachádzať viacero značiek a úlohou tejto funkcie je filtrovanie na základe mac adresy⁴.

Poslednú funkciu, ktorú tento modul ponúka je `utils_fast_log2`. Výpočet logaritmu s plnou presnosťou zaberá veľmi veľa času pri vyhodnotení vo vstavanom systéme. Pokiaľ ale nie je nutná veľká presnosť, je možné výpočet logaritmu niekoľkokrát urýchliť.

Desatinné čísla typu `float` a `double` sú v počítačových systémoch reprezentované mantisou, exponentom a znamienkovým bitom podľa IEEE 754 [1]. Rovnica 5.5 ukazuje výpočet reálnej hodnoty premennej typu `float`.

$$Value = (-1)^{sign} \cdot 2^{exp-127} \cdot (1 + mantissa) \quad (5.5)$$

Akcelerácia výpočtu následne pozostáva v predvypočítaní tabuľky logaritmov pre všetky možné hodnoty mantisy. Tento proces môže byť vykonaný napríklad po štarte zariadenia, v čase keď rýchlosť nie je kritická. Veľkosť tabuľky, teda počet predvypočítaných hodnôt je daný na základe požadovanej presnosti. Požadovaná presnosť určuje, koľko bitov mantisy je použitých pri výpočte. Špecificky v mojej implementácii je pri výpočte logaritmu použitých len prvých 5 bitov a tabuľka teda obsahuje 2^5 predvypočítaných hodnôt typu `float`. Táto tabuľka teda zaberá 128B miesta v pamäti výmenou za niekoľkonásobné urýchlenie.

Samotný výpočet spočíva už len v extrakcii exponentu a mantisy z desatinného čísla. Exponent je následne pričítaný k predvypočítanej hodnote logaritmu, ktorá je vyhľadaná pomocou mantisy. Znamienkový bit môže byť ignorovaný, keďže logaritmus v reálnych číslach je definovaný len pre pozitívne hodnoty.

Les a jeho reprezentácia

Modul `tree_node` definuje ako vyzerá uzol stromu. Tento uzol je popísaný v kapitole 5.2. Bez implicitnej špecifikácie všetky štruktúry v jazyku C môžu zaberáť viac miesta v pamäti ako kombinovaná veľkosť všetkých položiek. Každá položka v štruktúre má tzv. „padding“, čo umožňuje aby všetky položky mali od seba rovnakú vzdialenosť. Tento prístup zvyšuje rýchlosť práce so štruktúrou, ale zároveň zvyšuje pamäťové nároky na jej uloženie. Keďže mojím cieľom je uložiť celý klasifikačný model do pamäte cieľového zariadenia tak, aby zaberá čo najmenej miesta, celá štruktúra má nadefinovaný atribút „packed“, viď výpis kódu 5.1. To znamená, že jej položky sú „stlačené“, čiže je eliminovaný padding.

Následne modul `tree_data` obsahuje vygenerované uzly Random Forest klasifikátora. Tieto súbory sú vygenerované pomocou skriptu, ktorý je popísaný v kapitole 5.2.

⁴MAC adresa – Unikátny 24B identifikátor každého zariadenia v sieti.


```

1 typedef struct __attribute__((__packed__)) {
2     union {
3         uint16_t probability;
4         uint16_t threshold;
5     } u;
6     uint16_t index_right;
7     uint8_t feature:7;
8     uint8_t is_leaf:1;
9 } tNode;

```

Výpis 5.1: Štruktúra popisujúca uzol rozhodovacieho stromu

Klasifikácia

Finálnou časťou knižnice je klasifikácia dát, ktorá je vykonaná v súbore `classifier`. Funkcia `classifier_predict` obsahuje klasifikačný algoritmus, ktorý prechádza vygenerovaný les po jednotlivých stromoch a nakoniec vráti výslednú pravdepodobnosť LOS signálu, viď algoritmus 1. Táto funkcia je zaobalená funkciou `classifier_predict_packet`, ktorá slúži na prípravu parametrov, meranie časovania a diagnostiku. V prvom rade dopočíta a predprípraví parametre do formátu, ktorý očakáva klasifikátor. Zároveň spúšťa meranie počtu inštrukčných cyklov a diagnostický výpis, ktorý je ovládaný makrom `CLASSIFIER_DEBUG`.

Algoritmus 1: Klasifikácia signálov pomocou metódy Random Forest

Input: Features, Forest, Roots
Output: Probability

```

1 result ← 0;
2 foreach root in Roots do
3     index ← root;
4     loop
5         node ← Forest[index];
6         if node is leaf then
7             break;
8         end
9         if Features[node.feature] ≥ node.threshold then
10            index ← index + 1;
11        else
12            index ← root + node.index_right;
13        end
14    end
15    result ← result + node.probability/size(Roots);
16 end

```

5.4 Vyhodnotenie

Ako je popísané v kapitole 4.1, klasifikácia signálov nie je triviálny problém. Pri vytváraní datasetu môžu nastať klamné situácie, ktoré vedú k chybným vzorkám. Zároveň aj pokiaľ je celý dataset perfektný, niektoré signály nebude možné odlišiť. Napríklad pokiaľ by bola priama cesta signálu zatienená kovovým objektom, ktorý sa pri použitých frekvenciách správa ako zrkadlo, aj odrazený signál môže mať vlastnosti priameho. Z toho dôvodu pri testovaní budem pozorovať len situácie ako napríklad zakrytie stenou alebo zatienenie ľudským telom.

Cieľom tejto kapitoly je:

- Zmerať rýchlosť a pamäťové nároky klasifikačného algoritmu vo vstavanom systéme
- Vyhodnotiť, či je možné detegovať objekt medzi kotvou a značkou
- Overiť, či riešenie funguje pre rôzne nastavenia dĺžky preamble.

Evaluácia vlastností algoritmu

Vyhodnotenie prebiehalo na 32 bit MCU⁵ založenom na Arm Cortex-M4⁶, ktoré je taktované na 120 MHz. Celý klasifikačný algoritmus je implementovaný ako knižnice pre existujúci systém lokalizácie pomocou UWB, ktorého základom je FreeRTOS⁷. Vyhodnotenie signálu prebieha ihneď po jeho prijatí.

Pri testovaní som zistil, že implementovaný klasifikačný algoritmus je schopný vyhodnotiť dáta rýchlo a efektívne. Doba klasifikácie je špecifická na hĺbke jednotlivých stromov a na počte použitých stromov. Model, ktorý som využil na testovanie obsahuje 4 stromy, ktoré dohromady obsahujú 800 uzlov. Vyhodnotenie tohto modelu trvá 1200 inštrukčných cyklov. Jeden uzol zaberá 5B pamäte, čiže celý model sa zmestí do 4000B.

Pred klasifikáciou je ešte nutné pripraviť a vypočítať dodatočné parametre, ktoré sú popísané v 5.2. Tento proces trvá až 5000 inštrukčných cyklov, čo je oveľa viac ako samotné vyhodnotenie modelu. Dôvodom dlhšej prípravy parametrov je práca s desatinnou čiarkou, teda s premennými typu `float`. Arm Cortex M4 síce obsahuje vlastné FPU⁸, ale operácie s desatinnými číslami sú časovo náročné. Implementácia vlastnej verzie rýchleho logaritmu podstatne urýchlila dobu výpočtu. Pri použití funkcie `log2` z matematickej knižnice, doba prípravy parametrov sa pohybovala v okolí 23000 inštrukčných cyklov.

Celý proces klasifikácie a prípravy dát trvá 6000 inštrukčných cyklov, čo predstavuje len 50 μ s. Klasifikácia teda nespomaľuje systém, ani nenarúša kritické procesy ako napríklad synchronizáciu času alebo meranie vzdialenosti.

⁵V kotve je použitý mikrokontrolér TM4C1294KCPDT, viď www.ti.com/product/TM4C1294KCPDT.

⁶Arm Cortex-M4 – Viď developer.arm.com/Processors/Cortex-M4.

⁷FreeRTOS – Jednoduchý operačný systém pre mikrokontroléry, ktorý poskytuje rôznu funkcionality, okrem iného aj prácu s procesmi a ich sledovanie. Viď freertos.org.

⁸FPU – Floating-Point Unit je matematický koprocesor zameraný na prácu s desatinnými číslami.

Vyhodnotenie funkčnosti

Po redukcii počtu parametrov a ich upravení tak, aby boli nezávislé na preambule, úspešnosť klasifikácie na datasete dosahuje 88%. Vyhodnotenie úspešnosti v reálnom svete nie je možné perfektne vyčíslieť. Z toho dôvodu toto vyhodnotenie bude subjektívnejšie. Evaluácia bude pozostávať z dvoch prípadov:

- Značka sa nachádza v inej miestnosti ako kotva, teda je zakrytá stenou
- Značka je zatienená ľudským telom

Pri prvom testovaní klasifikačný model nebol schopný korektne detegovať LOS a NLOS signály. Až pri zatienení niekoľkými stenami bol model schopný správne klasifikovať vzorky ako NLOS signál. Takáto presnosť je však nedostačujúca.

Dataset bol síce nazbieraný vo viacerých miestnostiach, aj napriek tomu model však môže byť natrénovaný špecificky na tieto miestnosti. V [18] bolo otestovaných viacero klasifikačných metód. Ich úspešnosť v miestnostiach, ktoré boli použité na tréning dosahovala 99%. Na druhú stranu v miestnostiach, ktoré neboli videné pri tréningu, sa úspešnosť klasifikácie razantne znížila a pohybovala v okolí 60%. Tento trend som pozoroval aj pri tréňovaní rozhodovacích stromov na jednotlivých častiach datasetu. Úspešnosť klasifikácie bola vždy vyššia, pokiaľ klasifikačný model videl niekoľko vzoriek z danej miestnosti.

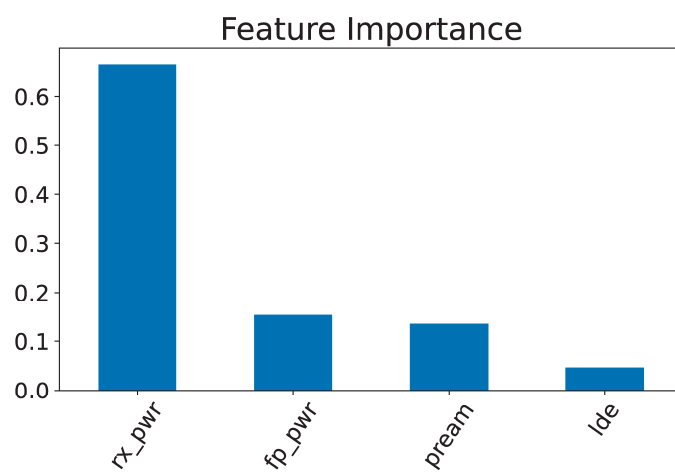
Elimináciu tohto problému som vyriešil opätovným natrénovaním modelu s novými vzorkami. Z kotvy, na ktorej prebiehala klasifikácia, som extrahoval 10 vzoriek a manuálne ich označil ako LOS alebo NLOS. Tieto vzorky následne neboli použité priamo pri tréňovaní, ale pri evaluácii. Ideálne parametre modelu sú hľadané pomocou triedy `GridSearchCV` z balíčku `skikit-learn`, podrobnejšie v 5.1. Funkcia, ktorá prideluje skóre každému vytvorenému modelu bola upravená tak, aby vyhodnocovala len tieto nové dáta. Model bol následne exportovaný a opätovne nahraný do kotvy. Táto verzia modelu už dokázala presne určiť, či je značka v inej miestnosti, alebo je zatienená ľudským telom.

Po analýze oboch modelov som zistil, že hlavný rozdiel medzi nimi tvorí podiel parametru $RXpower$ na klasifikácii. Pokiaľ je tento parameter dominantný, klasifikačný model je schopný detegovať zatienenie, viď obrázok 5.1. Pokiaľ sa ale model spolieha na iné parametre, úspešnosť detekcie LOS a NLOS signálov pri reálnom nasadení prudko klesá. Tento fakt som overil vytvorením viacerých modelov bez použitia mnou nazbieraných vzoriek. Pokiaľ dôležitosť parametru $RXpower$ bola dominantná, pri reálnom nasadení bolo možné klasifikovať signály. Aj napriek tomu, že sa klasifikácia spolieha hlavne na parameter $RXpower$, bez použitia ostatných parametrov nebolo možné úspešne klasifikovať signály.

Funkcionalita pri rôznej dĺžke preambuly

Jednou z najväčších obáv pri tréňovaní datasetu bola skutočnosť, že celý tréňovací dataset bol nazbieraný len pri použití dlhých preambúl. Namerané vzorky mali dĺžku 1024 a 1536 symbolov. V moderných systémoch lokalizácie sa ale používa už len preambula dlhá 128 symbolov. Výhody a nevýhody, ktoré plynú z dĺžky preambuly sú popísané v kapitole 2.4. Všetky použité parametre sú síce normalizované vzhľadom na dĺžku preambuly, reálne dáta sa však môžu líšiť.

Pri použití validného modelu, ktorý je popísaný v predchádzajúcej sekcii, bolo možné detegovať zatienenie pri použití dlhšej aj krátkej preambuly. Pri použití krátkej preambuly sa úspešnosť javila vyššia ako pri dlhšej, aj napriek tomu, že vzorky s krátkymi preambulami sa nenachádzajú v datasete.



Obr. 5.1: Dôležitosť jednotlivých parametrov funkčného klasifikačného modelu. Úspešnosť klasifikácie na datase je 88.5%.

Kapitola 6

Záver

Cieľom tejto práce je klasifikácia signálov vo vstavanom systéme. Aj napriek tomu, že sa nejedná o triviálny problém, sa mi podarilo jednoznačne dokázať, že takáto klasifikácia je možná aj v zariadení s obmedzeným výpočtovým výkonom. Postup navrhnutý v tejto práci má lepšiu úspešnosť ako postup, ktorý je momentálne využívaný v industriálnom prostredí. Zároveň som ukázal, že komplikovaná extrakcia a výpočet parametrov nemá žiadny významný dopad na úspešnosť klasifikácie.

Navrhnutý postup dokáže s úspešnosťou 88% klasifikovať typ signálu pri klasifikácii na nazbieraných vzorkách z viacerých prostredí. Pri reálnom nasadení je model schopný spoľahlivo detegovať prekážku medzi kotvou a značkou. Táto klasifikácia je možná v reálnom čase, keďže trvá len 6000 inštrukčných cyklov. To znamená, že mikroprocesor v kotve dokáže tento signál klasifikovať za 20 μ s. Použitý model zaberá len 4kB pamäte. Kotvy použité v industriálnom prostredí síce nie sú optimalizované na spotrebu energie, no implementovaný algoritmus vyhodnotenia dát je univerzálny a môže byť použitý v akomkoľvek MCU, tým pádom aj v zariadeniach s nízkou spotrebou, ktoré sú napájané z batérie.

V tejto práci by sa dalo pokračovať zmenou metodológie, pomocou ktorej boli nazbierané datasety. Viaceré štúdie naznačujú, že detekcia priamych (LOS) a zatičených (NLOS) signálov môže byť zo svojej podstaty nedostatočná. Keďže typ signálov je klasifikovaný ručne, dataset môže obsahovať chybné vzorky. Navyše podľa [12] chyba pri meraní vzdialenosti môže nastať ako pri NLOS signáloch, tak aj pri LOS. Zároveň nie všetky NLOS signály spôsobujú kritickú chybu merania a je možné ich využiť pri lokalizácii bez straty presnosti. V niektorých štúdiách, ako napríklad [7] a [15], autori navrhujú detekciu viacerých druhov šírenia signálu. Medzi tie patrí multipath¹ a rozdelenie NLOS signálov na „Hard NLOS“ a „Soft NLOS“, teda či boli signály zatičené úplne alebo len čiastočne.

Keďže chyba merania je analógová veličina, lepšie výsledky by mohla mať klasifikácia, ktorá vyhodnocuje priamo túto chybu. To by však vyžadovalo konštrukciu špeciálneho zariadenia, ktoré by bolo schopné presne zaznamenávať chybu merania vzdialenosti. Zároveň, každá miestnosť vykazuje vlastné charakteristické správanie pri odrazoch signálu. Z toho dôvodu klasifikačný model, ktorý je natrénovaný na dátach z danej miestnosti, môže dosiahnuť lepšiu úspešnosť ako univerzálny. Preto by bolo prospešné preskúmať možnosť trénovania klasifikačných modelov priamo v kotve až pri jej nasadení. Rozhodovacie stromy sú zároveň jedným z klasifikačných algoritmov, ktoré je možné trénovať priebežne, a preto by boli vhodné aj na túto úlohu.

¹Multipath – Medzi vysielačom a prijímačom síce existuje priama viditeľnosť, ale sú zvýšené podmienky na odraz signálu.

Literatúra

- [1] IEEE Standard for Floating-Point Arithmetic. *IEEE Std 754-2008*. 2008. DOI: 10.1109/IEEEESTD.2008.4610935.
- [2] IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*. 2011. DOI: 10.1109/IEEEESTD.2011.6012487.
- [3] *UWB Transceiver*. DW1000. Decawave, 2017. Rev. 2.18.
- [4] IEEE Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques. *IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020)*. 2020. DOI: 10.1109/IEEEESTD.2020.9179124.
- [5] AGGARWAL, C. C. *Data Classification: Algorithms and Applications*. CRC Press, 2014. ISBN 9780429102639.
- [6] ALBAWI, S., MOHAMMED, T. A. a AL ZAWI, S. Understanding of a convolutional neural network. In: *2017 International Conference on Engineering and Technology (ICET)*. 2017. DOI: 10.1109/ICEngTechnol.2017.8308186.
- [7] BARRAL, V., ESCUDERO, C. J., GARCÍA NAYA, J. A. a MANEIRO CATOIRA, R. NLOS Identification and Mitigation Using Low-Cost UWB Devices. *Sensors*. 2019, zv. 19, č. 16. DOI: 10.3390/s19163464. ISSN 1424-8220.
- [8] BERRAR, D. Bayes' Theorem and Naive Bayes Classifier. In: . Január 2018. DOI: 10.1016/B978-0-12-809633-8.20473-1. ISBN 9780128096338.
- [9] BREGAR, K., HROVAT, A. a MOHORCIC, M. NLOS Channel Detection with Multilayer Perceptron in Low-Rate Personal Area Networks for Indoor Localization Accuracy Improvement. In: . Máj 2016.
- [10] BREGAR, K., HROVAT, A. a MOHORCIC, M. NLOS Channel Detection with Multilayer Perceptron in Low-Rate Personal Area Networks for Indoor Localization Accuracy Improvement. In: . Máj 2016.
- [11] FEDERAL COMMUNICATIONS COMMISSION. *Revision of Part 15 of the Commission's Rules Regarding Ultra-Wideband Transmission Systems*. April 2002. Docket 98-153, FCC 02-48. Dostupné z: www.fcc.gov.
- [12] FLUERATORU, L., WEHRLI, S., MAGNO, M. a NICULESCU, D. On the Energy Consumption and Ranging Accuracy of Ultra-Wideband Physical Interfaces.

In: *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. 2020. DOI: 10.1109/GLOBECOM42002.2020.9347984.

- [13] HAYKIN, S., HAYKIN, S. a HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999. International edition. ISBN 9780132733502.
- [14] JIANG, C., SHEN, J., CHEN, S., CHEN, Y., LIU, D. et al. UWB NLOS/LOS Classification Using Deep Learning Method. *IEEE Communications Letters*. 2020, zv. 24, č. 10. DOI: 10.1109/LCOMM.2020.2999904.
- [15] LIAN SANG, C., STEINHAGEN, B., HOMBURG, J., ADAMS, M. a HESSE, M. Identification of NLOS and Multi-Path Conditions in UWB Localization Using Machine Learning Methods. *Applied Sciences*. Jún 2020, zv. 10, s. 3980. DOI: 10.3390/app10113980.
- [16] MUSA, A., NUGRAHA, G. D., HAN, H., CHOI, D., SEO, S. et al. A decision tree-based NLOS detection method for the UWB indoor location tracking accuracy improvement. *International Journal of Communication Systems*. 2019, zv. 32, č. 13. DOI: <https://doi.org/10.1002/dac.3997>. e3997 dac.3997.
- [17] OPPERMANN, E., L., I. E. H. m. a EDITOR:JAR. *UWB: Theory and applications*. John Wiley & Sons, 2005.
- [18] PARK, J., NAM, S., CHOI, H., KO, Y. a KO, Y.-B. Improving Deep Learning-Based UWB LOS/NLOS Identification with Transfer Learning: An Empirical Approach. *Electronics*. Október 2020, zv. 9. DOI: 10.3390/electronics9101714.
- [19] QORVO. *Ultra-Wideband For Dummies*. John Wiley & Sons, Inc., 2021. ISBN 978-1-119-80959-3.
- [20] TANK, B. *Identification and Mitigation of NLOS based on Channel Information Rules for Indoor UWB Localization*. Diplomová práca.
- [21] ZANDIAN, R. a WITKOWSKI, U. NLOS Detection and Mitigation in Differential Localization Topologies Based on UWB Devices. In: *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2018. DOI: 10.1109/IPIN.2018.8533781.

Príloha A

Obsah priloženého pamäťového média

Priložené pamäťové médium obsahuje nasledujúce súbory:

```
/
├── src/
│   ├── scripts/
│   ├── notebooks/
│   ├── library/
│   └── requirements.txt
├── text/
├── poster/
├── demonstration-video.mp4
├── poster.pdf
├── xchalk02-Klasifikacia-signalov.pdf
└── README
```

- **src** – Priečink obsahujúci zdrojové súbory implementovaného riešenia.
- **scripts** – Skripty napísané v jazyku Python3, ktoré sú využívané v jednotlivých notebookoch.
- **notebooks** – Súbory typu Jupyter notebook, ktoré obsahujú evaluáciu dát a tréovanie klasifikačných metód.
- **library** – Knížnica pre kotvu napísaná v jazyku C.
- **requirements.txt** – Zoznam závislostí jazyka Python.
- **poster** – Priečink obsahujúci zdrojové súbory plagátu.
- **text** – Priečink obsahujúci zdrojové súbory tejto práce.
- **video.mp4** – Video demonštrujúce funkčnosť implementácie.
- **poster.pdf** – Stručný plagát popisujúci prácu.
- **xchalk02-Klasifikacia-signalov.pdf** – Súbor obsahujúci túto prácu.
- **README** – Súbor obsahujúci popis súborov na pamäťovom médiu.