



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

METODY KLASIFIKACE SIGNÁLU PRO UWB IEEE 802.15.4Z SECURE RANGING

SIGNAL CLASSIFICATION METHODS FOR UWB IEEE 802.15.4Z SECURE RANGING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Klíma

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Radovan Juráň

BRNO 2025

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Petr Klíma

ID: 230586

Ročník: 2

Akademický rok: 2024/25

NÁZEV TÉMATU:

Metody klasifikace signálu pro UWB IEEE 802.15.4z secure ranging

POKYNY PRO VYPRACOVÁNÍ:

Technologie UWB (Ultra-wideband) IEEE 802.15.4z nachází uplatnění v zabezpečených bezdrátových přístupových systémech, automotive pro bezklíčové odemykání, průmyslové lokalizaci, či mobilních telefonech. Standard podporuje tzv. „secure ranging“, zejména díky STS (scrambled timestamp sequence). Odražený signál může vzdálenost zkreslovat, což hraje roli zejména při přesné lokalizaci. Pro detekci přímého (LoS) a odraženého (NLoS) signálu lze využít impulzní odezvu kanálu. Cílem práce je evaluace metod pro klasifikaci přímého a odraženého přijatého signálu. Práce se zaměří na nejnovější UWB kanál CH9 a porovnání s publikovanými výsledky pro starší hardware a nižší kanály.

Přehled a studium publikovaných metod klasifikace v definovaných scénářích a jejich prvotní ověření na dostupných datasetech jsou prerekvizity výstupu diplomové práce, jímž bude výběr a implementace alespoň jedné metody pro klasifikaci LoS/NLoS posouzení vlivu kanálu (frekvence) na přenositelnost algoritmu.

DOPORUČENÁ LITERATURA:

[1] X. Luo, C. Kalkanli, H. Zhou, P. Zhan and M. Cohen, "Secure Ranging with IEEE 802.15.4z HRP UWB," 2024 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2024, pp. 2794-2811, doi: 10.1109/SP54263.2024.00238.

[2] FLUERATORU, Laura; LOHAN, Elena Simona a NICULESCU, Dragos. Challenges in platform-independent UWB ranging and localization systems. Online. 2022, s. 9-15. ISBN 9781450395274. Dostupné z: <https://doi.org/10.1145/3556564.3558238>. [cit. 2023-04-05].

Termín zadání: 10.2.2025

Termín odevzdání: 27.5.2025

Vedoucí práce: Ing. Radovan Juráš

prof. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce posuzuje dosud nedostatečně prozkoumaný vliv frekvenčního kanálu na přenositelnost modelů strojového učení pro detekci přímého (LoS) a odraženého (NLoS) signálu v systémech Ultra-WideBand (UWB) s cílem ověřit publikované metody klasifikace a posoudit vliv volby UWB kanálu (frekvence) na přenositelnost klasifikačních modelů. Detekce odraženého signálu je klíčová pro zajištění nejen přesného, ale i bezpečného měření vzdálenosti.

Bylo analyzováno 23 publikovaných metod pro klasifikaci LoS a NLoS signálu. Na základě analýzy lze publikovaná řešení rozdělit na 2 kategorie – řešení využívající pouze impulsní odezvu kanálu a řešení využívající extrahované charakteristiky z impulsní odezvy kanálu a hodnoty z registrů UWB zařízení. Pro podrobné ověřování bylo vybráno 6 z těchto metod, které byly analyzovány s využitím několika scénářů. Pro účely ověřování byl vypracován přehled dostupných datasetů, a zároveň byla získána i neveřejná datová sada z průmyslového prostředí a kanálu 9. Pro posouzení vlivu frekvence na přenositelnost modelů bylo využito 4 různé scénáře, včetně tréninku a ověřování modelu nejprve na stejném, následně i na různém kanálu.

Z experimentů vyplývá negativní vliv frekvence na výkonnost modelu, výrazný vliv frekvence na jeho přenositelnost a mezi daty jednotlivých kanálů byly odhaleny rozdíly v distribuci. Vliv frekvence se mezi kanály liší, ale byl pozorován trend klesající výkonnosti modelu se zvyšující se frekvencí kanálu. Nejvýraznější pokles byl pozorován při přenosu mezi skupinami nízkofrekvenčních a vysokofrekvenčních kanálů. Při ověřování jednotlivých modelů byl potvrzen významný vliv prostředí na výkonnost jednotlivých modelů. Pro data z jiného prostředí, získaná odlišným zařízením nebo metodikou, se mohou závěry o vlivu frekvence na přenositelnost lišit.

KLÍČOVÁ SLOVA

Bezpečná lokalizace, Klasifikace, Nepřímá viditelnost (NLoS), Přímá viditelnost (LoS), Přenositelnost modelů, Publikované datové sady, Strojové učení, Ultra-WideBand (UWB), Vliv frekvenčního kanálu.

ABSTRACT

This thesis addresses the insufficiently explored impact of the frequency channel on the transferability of machine learning models for detecting Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) signals in Ultra-WideBand (UWB) systems. The work has two main objectives: to verify published classification methods and to assess the influence of UWB channel (frequency) selection on the transferability of classification models. Detecting reflected signals is crucial not only for accurate but also for secure distance measurement.

A total of 23 published methods for LoS/NLoS classification were analyzed. Based on this analysis, the existing solutions can be divided into two categories: methods utilizing only the channel impulse response (CIR) and methods employing extracted features from the CIR combined with register values from UWB devices. For detailed evaluation, six of these methods were selected and tested across multiple scenarios. A comprehensive overview of available datasets was compiled, and an additional proprietary dataset from an industrial environment on channel 9 was obtained. To evaluate the impact of frequency on model transferability, four different scenarios were considered, including training and testing models first on the same channel and then on different channels.

The experiments reveal a negative effect of frequency on model performance, a significant influence of frequency on transferability, and notable distributional differences between data from individual channels. While the impact of frequency varies among channels, a general trend of decreasing model performance with increasing channel frequency was observed. The most pronounced performance drop occurred when transferring models between low-frequency and high-frequency channel groups. Additionally, the evaluation confirmed a significant influence of the environment on model performance. Consequently, conclusions regarding the effect of frequency on transferability may differ when using data from different environments, devices, or methodologies.

KEYWORDS

Channel frequency impact, Classification, Line-of-Sight (LoS), Machine learning, Model transferability, Non-Line-of-Sight (NLoS), Published datasets, Secure ranging, Ultra-WideBand (UWB).

KLÍMA, Petr. *Metody klasifikace signálu pro UWB IEEE 802.15.4z secure ranging*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2025. Vedoucí práce: Ing. Radovan Juráň

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Bc. Petr Klíma
VUT ID autora: 230586
Typ práce: Diplomová práce
Akademický rok: 2024/25
Téma závěrečné práce: Metody klasifikace signálu pro UWB
IEEE 802.15.4z secure ranging

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....
podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu semestrální práce Ing. Radovanu Juráňovi, za odborné vedení, metodickou, pedagogickou a odbornou pomoc, podnětné návrhy a delší cenné rady při zpracování mé diplomové práce. Také bych rád poděkoval zástupcům firmy Sewio networks s.r.o. za poskytnutá data z reálného prostředí. Své poděkování bych rád vyjádřil i své přítelkyni a rodině, která mi byla oporou při tvorbě této práce.

Obsah

Úvod	19
1 Technologie Ultra–WideBand	21
1.1 Lokalizační techniky	22
1.1.1 Time of Arrival	22
1.1.2 Two Way Ranging	23
1.1.3 Time Difference of Arrival	24
1.1.4 Další lokalizační techniky	24
1.2 Standardy	24
1.2.1 IEEE 802.15.4	25
1.2.2 Aple nearby interaction	26
1.2.3 Digital key 3.0	26
1.2.4 FiRa standard	26
1.3 Režimy fyzické vrstvy	26
1.3.1 HRP	27
1.3.2 LRP	27
1.4 Bezpečnost	28
1.4.1 Bezpečná trénovací sekvence	29
1.4.2 Přijímač	29
1.4.3 Útoky	31
1.5 LoS a NLoS signál	33
2 Strojové učení	35
2.1 Data	36
2.1.1 Transformace a předzpracování dat	37
2.2 Klasifikace	39
2.2.1 Rozhodovací strom	40
2.2.2 Metoda podpůrných vektorů	41
2.2.3 Umělé neuronové sítě	42
2.2.4 Hodnocení výkonnosti modelu	46
2.2.5 Generalizace	48
3 Stávající řešení	53
3.1 Dostupné datové sady	54
3.1.1 Neveřejná datová sada	60
3.2 Ověřování publikovaných metod	61
3.2.1 Scénáře pro ověřování	62

3.2.2	CNN LSTM	63
3.2.3	Novel CNN Model	66
3.2.4	Resnet	70
3.2.5	MLP využívající přenesené učení	74
3.2.6	SVM LDA	77
3.2.7	RF	80
3.2.8	Porovnání ověřovaných řešení	83
3.3	Další publikovaná řešení	86
3.3.1	Řešení využívající konvoluční vrstvy	87
3.3.2	Řešení bez konvolučních vrstev	93
3.4	Shrnutí	101
4	Stanovení vlivu frekvence	105
4.1	Metodologie stanovení vlivu frekvence	105
4.2	Scénář 1 – Trénink a testování na stejném kanálu	108
4.3	Scénář 2 – Trénink a testování na různých kanálech	113
4.4	Scénář 3 – Testování na kanálu vyřazeném z tréninku	120
4.5	Scénář 4 – Porovnání extrahovaných charakteristik modelu RF	125
4.6	Shrnutí	127
	Závěr	129
	Literatura	131
	Seznam symbolů a zkratk	143
A	Vybrané výstupy scénářů z kapitoly 4	147
A.1	Scénář 4.3	147
A.2	Scénář 4.4	153
B	Obsah elektronické přílohy	157

Seznam obrázků

1.1	Princip lokalizačních technik „Time of Arrival“ a „Two Way Ranging“.	23
2.1	Znázornění principu algoritmu SVM.	41
2.2	Průběhy vybraných aktivačních funkcí.	44
3.1	Časové průběhy impulsní odezvy kanálu náhodně vybraného LoS vzorku v každém datasetu.	59
3.2	Časové průběhy impulsní odezvy kanálu náhodně vybraného LoS a NLoS vzorku v datasetu 3.1.1.	60
4.1	Krabicový graf vyvážené přesnosti modelů trénovaných a testovaných na stejném kanálu.	111
4.2	Sloupcový graf vyvážené přesnosti modelů trénovaných a testovaných na stejném kanálu.	112
4.3	Heatmapa s výsledky post-hoc testů znázorňující rozdíly v hodnotách metriky „vyvážená přesnost“ napříč kanály.	113
4.4	Heatmapa s hodnotami metriky „vyvážená přesnost“ pro modely trénované a testované na různých kanálech.	116
4.5	Heatmapa s rozdíly ve sledované metrice „přesnost“ pro modely trénované na jiném než testovacím kanálu vůči hodnotám metriky získaných tréninkem a ověřováním na stejném kanálu.	117
4.6	Heatmapa s rozdíly ve sledované metrice „vyvážená přesnost“ mezi 3. a 2. scénářem.	122
4.7	Krabicový graf znázorňující rozdíly v extrahovaných vlastnostech napříč jednotlivými kanály.	125
4.8	Violin graf znázorňující rozdíly v extrahovaných vlastnostech napříč jednotlivými kanály.	127
4.9	Q-Q graf znázorňující rozdíly v rozdělení extrahovaných vlastností napříč jednotlivými kanály.	128
A.1	Heatmapa s hodnotami metriky „přesnost“ pro modely trénované a testované na různých kanálech.	147
A.2	Heatmapa s hodnotami metriky „vyvážená přesnost“ pro modely trénované a testované na různých kanálech.	149
A.3	Heatmapa s hodnotami metriky „F1 skóre“ pro modely trénované a testované na různých kanálech.	149
A.4	Heatmapa s hodnotami metriky „ROC AUC“ pro modely trénované a testované na různých kanálech.	150
A.5	Heatmapa s rozdíly ve sledované metrice „přesnost“ pro modely trénované na jiném než testovacím kanálu vůči hodnotám metriky získaných tréninkem a ověřováním na stejném kanálu.	150

A.6	Heatmapa s rozdíly ve sledované metrice „vyvážená přesnost“ pro modely trénované na jiném než testovacím kanálu vůči hodnotám metriky získaných tréninkem a ověřováním na stejném kanálu.	151
A.7	Heatmapa s rozdíly ve sledované metrice „F1 skóre“ pro modely trénované na jiném než testovacím kanálu vůči hodnotám metriky získaných tréninkem a ověřováním na stejném kanálu.	151
A.8	Heatmapa s rozdíly ve sledované metrice „ROC AUC“ pro modely trénované na jiném než testovacím kanálu vůči hodnotám metriky získaných tréninkem a ověřováním na stejném kanálu.	152
A.9	Heatmapa s rozdíly ve sledované metrice „vyvážená přesnost“ mezi 3. a 2. scénářem.	154
A.10	Heatmapa s rozdíly ve sledované metrice „přesnost“ mezi 3. a 2. scénářem.	154
A.11	Heatmapa s rozdíly ve sledované metrice „F1 skóre“ mezi 3. a 2. scénářem.	155
A.12	Heatmapa s rozdíly ve sledované metrice „ROC AUC“ mezi 3. a 2. scénářem.	155

Seznam tabulek

3.1	Přehled dostupných datasetů použitelných pro účely klasifikace LoS/N-LoS signálu.	58
3.2	Rozložení vzorků z veřejně dostupných datasetů v kontextu použitého UWB kanálu.	60
3.3	Výsledky ověřování publikovaného řešení <i>CNN LSTM</i> ([54]) ve scénářích popsáných v 3.2.1.	65
3.4	Výsledky ověřování publikovaného řešení <i>Novel CNN</i> ([55]) ve scénářích popsáných v 3.2.1.	68
3.5	Výsledky ověřování publikovaného řešení <i>Resnet</i> ([60]) ve scénářích popsáných v 3.2.1.	72
3.6	Výsledky ověřování publikovaného řešení <i>MLP TL</i> ([61]) ve scénářích popsáných v 3.2.5.	75
3.7	Výsledky ověřování publikovaného řešení <i>SVM LDA</i> ([62]) ve scénářích popsáných v 3.2.1.	78
3.8	Výsledky ověřování publikovaného řešení <i>RF</i> ([63]) ve scénářích popsáných v 3.2.1.	82
3.9	Výsledky ověřování 6 publikovaných řešení ve scénáři popsáném v 3.2.8.	84
3.10	Přehled publikovaných řešení popisovaných v rámci této práce.	102
4.1	Přehled frekvencí a šířek pásma odpovídajících jednotlivým HRP kanálům z použité části datasetu 2 dle IEEE 802.15.4z-2020.	106
4.2	Přehled výkonnostních metrik modelů trénovaných a ověřovaných na stejném kanálu.	110
4.3	Rozdíly ve sledované metrice „vyvážená přesnost“ pro modely trénované na jiném než testovacím kanále,	115
4.4	Přehled výkonnostních metrik modelů trénovaných na datech všech kanálů současně vyjma kanálu, na kterém byl model ověřován.	124
A.1	Rozdíly ve sledovaných metrikách pro modely trénované na jiném než testovacím kanálu vůči metrikám získaných tréninkem a ověřováním na stejném kanálu.	148
A.2	Pokles hodnot jednotlivých metrik ve scénáři 4.4 v porovnání s referenční hodnotou získanou tréninkem a ověřováním na stejném kanálu.	153

Úvod

V současné době představuje problematika přesného a bezpečného měření vzdálenosti klíčový prvek moderních bezdrátových systémů a lokalizačních technologií. S rostoucími nároky na přesnost lokalizace, požadovanou jak v automobilovém průmyslu, tak v bezpečnostních a průmyslových aplikacích, se stále více pozornosti věnuje technologii Ultra-WideBand (UWB). Díky využití velmi krátkých pulzů a širokému spektru umožňuje UWB dosahovat mimořádné přesnosti při určování polohy, a to i v prostředích, kde dochází k výraznému rušení.

Významným aspektem UWB technologií je jejich schopnost podporovat režim bezpečné lokalizace, který obsahuje opatření zajišťující integritu měření vzdálenosti. Chybná klasifikace přímé (Line-of-Sight, LoS) a nepřímé (Non-Line-of-Sight, NLoS) signálové cesty může vést k fatálním chybám v autonomních systémech nebo bezpečnostních aplikacích, což činí tuto problematiku kritickou pro reálné nasazení UWB technologií.

Tato práce se zaměřuje na evaluaci moderních metod strojového učení pro klasifikaci LoS a NLoS signálu. Cílem je podrobně analyzovat publikované metody pro klasifikaci se zaměřením na metody využívající impulsní odezvu kanálu s využitím veřejně dostupných dat. Ačkoli metody strojového učení pro klasifikaci LoS/NLoS jsou intenzivně zkoumány, dopad frekvenčního kanálu na přenositelnost modelů mezi různými UWB systémy zůstává téměř neprozkoumán. Dalším cílem této práce je tedy posouzení vlivu frekvence (kanálu), ze které pochází klasifikovaná data, na přenositelnost modelů pro klasifikaci.

Struktura této studie je rozdělena do několika klíčových částí. První kapitola představuje technologické aspekty UWB, včetně lokalizačních technik, standardizačních rámců a bezpečnostních mechanismů. Druhá kapitola se věnuje metodám strojového učení, které jsou nezbytné pro zpracování a klasifikaci dat získaných pomocí UWB technologií. Třetí kapitola poskytuje přehled existujících publikovaných řešení, detailní analýzu dostupných datových sad a mapuje výkonnostní limity jednotlivých metod. Závěrečná část práce diskutuje vliv frekvence na přenositelnost ověřovaných modelů.

1 Technologie Ultra–WideBand

Rádiová komunikace, která používá šířku pásma o velikosti větší nebo blíží se 500 MHz, je označována jako UWB (Ultra-WideBand). Větší pozornost je v dnešní době věnována technologii, která z UWB vychází. Používá velmi krátké rádiové pulzy s dobou trvání v řádech nano či pikosekund a označuje se jako „IR–UWB“ nebo také „UWB–IR“ (Impulse Radio UWB).

Díky tomu, že šířka pásma UWB je velmi vysoká, lze díky vysoké kapacitě kanálu vysílat s nízkým vysílacím výkonem, který zabraňuje vzniku úzkopásmového rušení s jinými bezdrátovými technologiemi. Vliv vícecestného šíření lze na přijímači eliminovat díky detekci a filtraci přicházejících krátkých rádiových pulzů. Navíc lze díky vysokému časovému rozlišení velmi přesně určit čas příchodu signálu, v důsledku čehož je pomocí této technologie a různých technik, které budou představeny v rámci této kapitoly, možné měřit vzdálenost s centimetrovou přesností a velmi nízkou latencí, při použití technik korekce chyb může být velikost chyby měření vzdálenosti dokonce menší než centimetr (58 mm).

Nevýhodou nízkého přenosového výkonu je omezení UWB komunikace na relativně krátké vzdálenosti. UWB pulzy jsou kvůli velké šířce pásma při srovnání s úzkopásmovými signály výrazně zkresleny, což může omezit výkon UWB přijímačů. Množství základnových stanic (kotev) má vliv na přesnost polohování a snaha o zvýšení přesnosti navyšuje i náklady. Základnové stanice i tagy¹ jsou navíc dražší než u jiných polohovacích systémů.

Nevýhodou je i to, že mnoho různých společností napříč různými odvětvími začalo přidávat UWB technologii do svých produktů. Zdaleka ne všechny tyto UWB čipy jsou otevřené pro vývojáře, a navíc mohou podporovat různé standardy. Kvůli tomu je použití technologie omezeno na zařízení používající stejný čip, což může vést ke snížení schopnosti UWB dosáhnout svého plného potenciálu ve všech aplikacích. Příkladem může být UWB čip DW 1000, který se hojně využívá pro výzkumné účely. Chyby polohování jsou závislé také na dostupných algoritmech, kterých pro tento UWB čip existuje velmi málo. Optimalizace takových algoritmů může značně navýšit náklady [1], [2].

UWB technologie se stává stále populárnější a má velmi široké spektrum využití, díky čemuž ji lze nalézt například v následujících aplikacích:

- Sledování v reálném čase – sledování polohy v reálném čase, což může být velmi užitečné například v průmyslových aplikacích při sledování pohybu zboží nebo sledování pohybu zaměstnanců v nebezpečných prostředích. UWB systémy nacházejí využití také v aplikacích automaticky řízených vozidel.

¹Zatímco základnová stanice (kotva) je stacionární bod, značka (tag) představuje bod, jehož pozice (vzdálenost) je měřena [1], [2].

- Bezklíčový přístup – využití pro bezkontaktní přístupové systémy. Mnozí výrobci automobilů (např. Audi a BMW) využívají UWB technologii pro bezklíčové odemykání a startování vozu. Společnost Samsung do svého zařízení „Galaxy Note 20“ implementovala UWB čip, který může sloužit jako digitální klíč k odemčení dveří.
- Kontrola kontaktů a sociální distancování – během pandemie Covid-19 byla UWB technologie použita pro sledování kontaktů a udržování sociálního distancování.
- Mobilní telefony – UWB čipy jsou integrovány do mobilních telefonů, díky čemuž lze vyvíjet další uživatelské aplikace. Například telefony společnosti Apple používají UWB pro získání prostorového povědomí, díky čemuž mohou velmi přesně lokalizovat další zařízení této společnosti.
- Navigace uvnitř budov – UWB systémy umožňující navigaci uvnitř budov, jako jsou letiště, nákupní centra či nemocnice [1], [3], [4], [5].

V následujícím textu této kapitoly budou představeny techniky pro měření vzdálenosti a určování polohy s využitím UWB technologie, stručně popsány budou i UWB standardy a režimy fyzické vrstvy. Poslední část této kapitoly se věnuje zabezpečenému měření vzdálenosti a možným útokům na aplikace využívající tuto technologii.

1.1 Lokalizační techniky

Největší využití UWB technologie je v oblasti lokalizace, potažmo měření vzdálenosti. K tomu slouží lokalizační techniky, jejichž cílem je při použití UWB tagu a více UWB kotev co nejpřesněji určit polohu tagu. Poloha tagu se určuje na základě geometrického vztahu mezi kotvou a značkou a známými informacemi o poloze. V následující kapitole jsou rozebrány jednotlivé techniky lokalizace [1], [2].

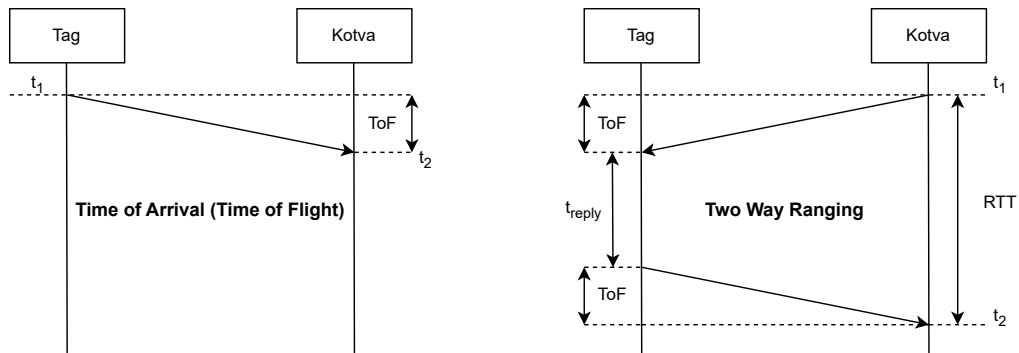
1.1.1 Time of Arrival

K určení vzdálenosti mezi UWB tagem a kotvami používá tato nejběžnější technika pro měření vzdálenosti čas příchodu signálu ToA (Time of Arrival), po kterém také nese své označení. V literatuře se lze setkat rovněž s označením ToF (Time of Flight), který označuje dobu letu signálu, ale princip obou metod je totožný a je ilustrován na obrázku 1.1, který byl převzat z [1]. Tag pravidelně odesílá UWB rámeček, který v payloadu (část s přenášenými užitečnými daty) obsahuje čas odeslání t_1 . Kotva přijme rámeček v čase t_2 a vypočítá ToF jako rozdíl těchto časů. Protože se elektromagnetický signál šíří rychlostí světla, je vzdálenost mezi zařízeními vypočtena

podle rovnice 1.1:

$$d = c \cdot \text{ToF} = c \cdot (t_2 - t_1)[\text{m}]. \quad (1.1)$$

Jakmile je vypočtena vzdálenost mezi tagem a všemi 3 kotvami, lze polohu značky určit pomocí trilaterace². Mezi všemi uzly komunikace musí existovat velmi přesná synchronizace, která má přímý dopad na přesnost měření a činí z nutnosti synchronizace nevýhodu této metody. Tento jednocestný systém lokalizace OWR (One-Way Ranging) je velmi jednoduchý a nijak nelimituje množství tagů, pro které probíhá měření [1], [2].



Obr. 1.1: Princip lokalizačních technik „Time of Arrival“ a „Two Way Ranging“.

1.1.2 Two Way Ranging

Vylepšením předchozí metody ToA je metoda TWR (Two Way Ranging), která eliminuje potřebu synchronizace mezi tagem a kotvou, čehož je dosaženo použitím časových razítek pouze z 1 zařízení. Kotva vysílá zprávu, která je tagem přijata v čase, který odpovídá době letu (ToF). Tag na tuto zprávu odpovídá po stanovené době, která je zahrnuta v odpovědi, díky čemuž kotva získá RTT³ (Round-Trip Time). Z RTT je získána doba letu ToA, pomocí které je určena vzdálenost mezi kotvou a tagem. Stejně jako u metody ToA je poloha tagu určena pomocí trilaterace díky hodnotám ze 3 kotev. Princip této techniky je ilustrován na obrázku 1.1. Problémem této techniky je, že kotva může komunikovat vždy pouze s 1 tagem, což limituje množství tagů, pro které může probíhat lokalizace. Pokud aplikace vyžaduje, aby svou polohu mohl určit i samotný tag, lze využít oboustrannou variantu TWR – tzv. DS-TWR (Double-Sided TWR), v rámci které jsou místo 2 zpráv vyslány alespoň 3 zprávy, díky kterým mohou vzdálenost mezi uzly určit jak kotva,

²Vzdálenost kotvy od tagu je poloměrem kružnice se středem v poloze kotvy. Více kružnic (3) se středy v různých kotvách se protínají v místě, které odpovídá poloze tagu [2].

³Doba, která uplyne od odeslání signálu do přijetí odpovědi [1], [2].

tak samotný tag. Přesnost této techniky může negativně ovlivnit přesnost časového razítka [1], [2].

1.1.3 Time Difference of Arrival

Protože jsou kotvy v různých vzdálenostech od tagu, tak signál vyslaný tagem dorazí ke kotvám v různých časech. Toho využívá technika TDoA (Time Difference of Arrival). Rozdíl času přijetí signálu ve 2 kotvách může být použit k výpočtu hyperboly. K určení polohy tagu se využívá průsečík alespoň 3 hyperbol. Výhodou je, že v tomto případě není nezbytná synchronizace mezi všemi uzly, ale pouze mezi kotvami [1].

1.1.4 Další lokalizační techniky

Mezi další techniky, které lze použít pro lokalizaci, patří například technika AoA (Angle of Arrival), která polohu určuje pomocí úhlu příchodu signálu, nebo také technika RSSI (Received Signal Strength Indicator – síla přijímaného signálu) využívající jevu, že výkon signálu při postupu prostorem slábne. Ač obě techniky řeší problém nezbytnosti silné synchronizace u některých z předchozích technik, mají rovněž své nevýhody. AoA vyžaduje uspořádání velkého množství anténních matic, což zvyšuje náklady. Nevýhoda RSSI pramení z principu této techniky, tj. že útlum signálu během šíření prostředím je silně ovlivněn právě prostředím, kterým se šíří. Byť jen malá změna v prostředí může vést k velkým změnám ve výsledcích určování polohy. Vzhledem k vysokému frekvenčnímu rozsahu UWB signálu a jeho vysokému časovému rozlišení dosahují podstatně větší přesnosti metody ToA, TDoA a TWR [2].

1.2 Standardy

Pro UWB definovalo standardy hned několik organizací. Kvůli odlišným cílům a rolím těchto organizací slouží i jejich standardy různým účelům a jsou umístěny na různých vrstvách modelu ISO/OSI⁴. Jak už bylo zmíněno dříve, existence různých standardů může způsobovat problémy s kompatibilitou, neboť různé UWB systémy mohou podporovat různé standardy. V této sekci jsou stručně rozebrány některé z nich [1].

⁴Open Systems Interconnection model – Výchozí model komunikační architektury pro realizaci veřejných datových sítí [6].

1.2.1 IEEE 802.15.4

Tento standard je výchozím bodem pro standardizaci UWB. Definuje fyzickou (v literatuře označení „PHY“) a linkovou vrstvu (v literatuře označení „MAC“) UWB. První standardizace proběhla v roce 2007 jako návrh s označením IEEE 802.15.4a, který byl začleněn v roce 2011. Jedná se o standardizaci UWB-IR zaměřenou na energeticky efektivní komunikaci s nízkou přenosovou rychlostí a zejména na přesné měření vzdálenosti [1], [7].

V roce 2012 vznikl návrh IEEE 802.15.4f, který specifikuje další režim fyzické vrstvy – LRP (Low Rate Pulse), a začleněn byl v roce 2015. Standard definuje 2 režimy fyzické vrstvy – HRP (High Rate Pulse) a LRP (oba režimy jsou podrobněji popsány v kapitole 1.3):

- HRP – vysílá pulzy s větší opakovací frekvencí (PRF – Pulse Repetition Frequency), ale menším vysílacím výkonem. Odpovídá specifikaci 802.15.4a.
- LRP – vysílá pulzy s menší opakovací frekvencí, ale větším vysílacím výkonem. V této specifikaci nepodporuje přesné měření vzdálenosti [1].

Nejnovější verze standardu se označuje jako IEEE 802.15.4z–2020. Přináší funkce pro zajištění integrity a přesnosti měření, které umožňují bezpečnější měření vzdálenosti. Definuje techniku tzv. simultánního měření. Každé z odpovídajících zařízení má mírně rozdílné zpoždění, ale jimi zaslané sekvence se mohou překrývat. Technika přijímači umožňuje zaměřit se pouze na 1 z odpovídajících zařízení a akumulovat všechny příchozí šifry po přijetí hlavičky a dat od prvního z nich, ale vyžaduje precizní synchronizaci všech kotevních bodů (odpovídajících zařízení). To může být vhodné například v automobilovém průmyslu při měření vzdálenosti mezi automobilem a klíčem, kdy kotvy umístěné v automobilu zasílají odpověď současně. Tato technika tak může prodloužit výdrž baterie v klíčku od automobilu. Silný důraz je kladen i na snížení celkové doby vysílání UWB zařízení, což může zvýšit kapacitu rádiového kanálu, snížit rušení mezi zařízeními, prodloužit životnost baterie a zvýšit bezpečnost z důvodu kratšího časového okna pro útoky. Rovněž přináší další možnosti kódování a preamble⁵ s obsahem proporcionálně menší sady nulových prvků, což vede ke zlepšení výkonu přijímačů. Bezpečnost HRP zvyšuje přidáním bezpečné trénovací sekvence STS (Secure Training Sequences). U režimu LRP definuje systém pro bezpečné měření vzdálenosti a umožňuje vyšší přenosové rychlosti (až 10 Mbps) [1], [3], [7].

⁵Slouží k identifikaci začátku rámce, předem daná sekvence bitů [6].

1.2.2 Aple nearby interaction

Společnost Apple byla jednou z prvních společností, která přidala UWB technologii do svých produktů. Pro možnost interakce příslušenství třetích stran s UWB čipem ve svých zařízeních společnost definovala specifikace pro zařízení a také vlastní protokol. Společnosti, které chtějí vyvíjet UWB řešení, která jsou kompatibilní s UWB čipem Apple, musí být součástí programu MFi (Made-For-iPhone) a musí následovat jejich protokol. Samotný protokol je definován nad síťovou a transportní vrstvou, je velmi odlehčený a umožňuje jednodušší konfiguraci, navázání a udržování UWB relace pro měření vzdálenosti mezi příslušenstvím třetích stran a zařízením společnosti Apple [1].

1.2.3 Digital key 3.0

Tento standard od konsorcia CCC (Car Connectivity Consortium) definuje UWB konektivitu pro bezdotykový přístup a další polohově založené funkce pro automobily nad síťovou, transportní i aplikační vrstvou. Zajišťuje nejvyšší bezpečnost při lokalizaci zařízení vzhledem k poloze vozidla, čímž umožňuje autorizaci uživatele pro přístup k vozidlu a jeho řízení. Cílem konsorcia je pokročit v technologiích pro propojení chytrých mobilních telefonů s automobily. Členy konsorcia jsou výrobci automobilů a telefonů, dodavatelé automobilových součástek a polovodičů i vývojáři aplikací [1].

1.2.4 FiRa standard

Cílem průmyslového konsorcia FiRa je poskytnout kompletní technické řešení pro služby založené na technologii UWB. Za tímto účelem vyvíjí specifikaci CSML (Common Service Management Layer), která je klíčová pro zajištění vzájemné interoperability mezi zařízeními FiRa, a zároveň poskytuje softwarový rámec a komponenty nezbytné pro nasazení aplikací [1].

1.3 Režimy fyzické vrstvy

V rámci aktuálního standardu IEEE 802.15.4z-2020 jsou definovány 2 režimy fyzické vrstvy UWB. Jedná se o již zmíněné režimy HRP a LRP. První z režimů vysílá s menší četností než ten druhý. Protože se jedná o širokopásmovou technologii, vysílací výkon UWB zařízení je limitován ve snaze zamezit interferenci. Čím vyšší je tedy četnost komunikace, tím nižší musí být použitý vysílací výkon. [1], [3]. V následujícím textu budou blíže popsány oba režimy a jejich vzájemné rozdíly.

1.3.1 HRP

HRP pulzy jsou vysílány s nižším výkonem, než u druhého z režimů, a v rámci standardu je definováno 16 různých kanálů s různými frekvencemi a šířkami pásma, přičemž zařízení nemusí podporovat každý z nich. Pro provoz v pásmu pod 1 GHz je povinná podpora kanálu 0, pro provoz v nízkém pásmu se jedná o kanál 3 a pro provoz ve vysokém pásmu musí zařízení podporovat kanál 9. Parametry povinně podporovaných kanálů dle IEEE 802.15.4z-2020:

- Kanál 0 – střední frekvence 499,2 MHz a šířka pásma 499,2 MHz.
- Kanál 3 – střední frekvence 4992,8 MHz a šířka pásma 499,2 MHz.
- Kanál 9 – střední frekvence 7987,2 MHz a šířka pásma 499,2 MHz.

V rámci množiny všech kanálů je minimální šířka pásma 499,2 MHz, nejnižší střední frekvence (kanál 0) 499,2 MHz a nejvyšší (kanál 15) 9984 MHz. Dosáhnout lze přenosové rychlosti až 27,24 Mbps, maximální frekvence vysílání pulzů je 499,2 MHz a jako modulace je použita BPSK (Binary Phase Shift Keying – Binární fázové klíčování) nebo kombinace BPSK s BPM (Burst Position Modulation – modulace polohy shluku pulzů).

Každý UWB čip může tedy podporovat různé kanály. V rámci publikace [1] byly mj. porovnány podporované kanály některých čipů na trhu (Apple U1, Qorvo DW1000 a DW3000, NXP SR04/SR150 a NCJ29D5, IMEC ULP IR-UWB radio) a všechny porovnávané čipy podporují kanál 5 (střední frekvence 6489,6 MHz, šířka pásma 499,2 MHz), s výjimkou čipu „Qorvo DW1000“ i kanál 9. Podpora zbylých kanálů je různá.

V rámci 802.15.4 se jednalo o jediný režim pro měření vzdálenosti, proto je nejčastěji implementován na komerčních UWB čipech [1].

1.3.2 LRP

Díky tomu, že jsou LRP pulzy vysílány s vyšším vysílacím výkonem, lze na vysílači zachytit a dekodovat jednotlivé pulzy. Standard pro tento režim definuje 10 kanálů ve frekvenčním rozsahu od 5624,32 do 10435,2 MHz. V rámci LRP existuje celkem 5 různých režimů, které jsou definovány kombinací použité modulace a opakovací frekvence, se kterou jsou vysílány pulzy, což ovlivňuje přenosovou rychlost, kvalitu signálu a množství spotřebované energie. Maximální opakovací frekvence jsou 4 MHz, maximální dosažitelná přenosová rychlost je 10 Mbps a mezi použité modulace patří:

- PPM (Pulse Position Modulation) – Pulzně polohová modulace.
- OOK (On Off Keying) – Binární modulace.
- PBFSK (Pulsed Binary Frequency Shift Keying) – Pulzní binární kmitočtové klíčování

Režim LRP lze pro měření vzdálenosti využít až od standardu IEEE 802.15.4z-2020, který implementuje základní schéma pro měření vzdálenosti na bázi měření RTT, tedy na bázi již zmíněné techniky „Two Way Ranging“. Hlavními vlastnostmi zařízení s tímto režimem jsou nízká spotřeba energie a snaha o kompatibilitu napříč výrobci. Velmi se rozšířil v automobilovém průmyslu pro bezdotykové odemykání a startování vozu [1], [3], [7], [8].

1.4 Bezpečnost

UWB se v posledních letech velmi rozšířilo především do systémů pro měření vzdálenosti. UWB čipy jsou nyní součástí mnoha chytrých mobilních telefonů, technologie nachází uplatnění i v automobilovém průmyslu pro bezklíčové odemykání a startování motoru, navíc je využívána i k ochraně stávajících systémů (systémy pasivního vstupu a startování) proti útokům využívajícím prodloužení signálu.

Většina nasazení této technologie následuje standard IEEE 802.15.4z. Jedná se o standard pro bezpečné měření vzdálenosti technologií UWB a nahrazuje standard IEEE 802.15.4a novými funkcemi pro zajištění integrity, které umožňují přesnější a bezpečnější měření vzdálenosti. Bezpečnostní koncept režimu HRP spoléhá na dlouhé náhodné STS, které se mohou skládat až ze 4096 pulzů. Tyto sekvence jsou vysílány vysílačem, a přijímač vypočítává čas ToA korelací přijatého signálu s lokální šablonou STS, která byla vygenerována použitím stejného semene (seedu). Typicky jeden z korelačních vrcholů určuje dobu přijetí paketu. Přijímače následně počítají agregovanou statistiku z přijatých pulzů, aby určily přesný čas příletu signálu (standard způsob výpočtu nespecifikuje). Samotné pulzy však mohou být silně ovlivněny vzájemnou interferencí a šumem [1], [3], [9], [10].

Z důvodu provozu nad širokým spektrem licencovaného pásma, jsou nastavena přísná regulační pravidla. Průměrná spektrální hustota výkonu nesmí překročit $-41,3$ dBm/MHz za 1 ms a výkon měřený v pásmu 50 MHz kolem špičkové frekvence je omezen na 0 dBm. Tato omezení limitují vysílací výkon na pulz, který pro použití na měření vzdálenosti v podmínkách bez přímé viditelnosti nebo na větší vzdálenosti využít nelze. Proto se při provozu za uvedených podmínek agregují informace shromážděné z více pulzů pro odhad času příletu (ToA) a dekodování dat.

Cílem útočníků je ovlivnit vzdálenost, ve které se podle výpočtů přijímače nachází vysílač, a to tak, aby byla menší. Takové útoky se v literatuře obecně označují jako „**distance shortening attacks**“ nebo také „**distance reduction attacks**“. Fyzická vzdálenost je měřena s využitím doby ToA rádiového signálu a případně i jeho vlastností. Útočník může ovládat odhad vzdálenosti manipulací s vyměňovaným signálem a může přijímač oklamat tak, aby změřil jiný, dřívější čas příletu [3]. Proces, který zahrnuje přesnou detekci polohy rádiového vysílače odolný vůči těmto

útokům, se v literatuře označuje jako „**secure ranging**“ (bezpečná lokalizace, bezpečné měření vzdálenosti) [10].

1.4.1 Bezpečná trénovací sekvence

Jedná se o sekvenci pulzů modulovaných pomocí BPSK a generovaných pseudo-náhodným bitovým generátorem – bit „0“ generuje kladný pulz, logická „1“ pulz záporný. Pulzy jsou vyslány s opakovací frekvencí 124,8 MHz. Každý paket pro určení vzdálenosti se může skládat až ze 2 STS sekvencí a každá sekvence může být rozdělena do 2 segmentů. Segmenty jsou zapouzdřeny do mezer trvajících 1 μ s. Přijímač korelací obdrženého signálu s lokálním vzorem STS, jenž byl generován s využitím stejného seedu jako STS vysílače, vypočítá ToA. K odhadu času příletu a ověření jeho integrity může být použit každý STS segment (standard IEEE 802.15.4z nespécifikuje způsob provedení).

Spolu s STS sekvencí mohou pakety pro určení vzdálenosti obsahovat také preambuli a payload. Norma nijak nespécifikuje jejich využití při odhadu ToA. Jejich případné použití by nezvýšilo bezpečnost odhadu, neboť preambuli je možné předpovědět a na payload, modulovaný pomocí BPM spolu s BPSK, lze využít útok typu ED/LC⁶[3].

1.4.2 Přijímač

Přijímač, který má být schopen provádět bezpečný odhad vzdálenosti nad HRP pulzy, musí podporovat přesnou detekci první bezpečné tréninkové sekvence STS. Standard navrhuje vypočítat impulsní odezvu kanálu (CIR) pro přenos STS, tedy že přijímač koreluje STS s lokálně uloženou šablonou. Na základě impulsní odezvy může přijímač přesně určit čas přijetí STS, který je následně předán vyšším vrstvám pro odhad vzdálenosti. Před tímto krokem je ale nezbytné, aby rozhodl, zda impulsní odezva obsahuje vrchol indikující přítomnost STS a zda je vysílač v **přímé (LoS)** nebo **nepřímé (NLoS)** viditelnosti, a teprve poté může určit přesný čas přijetí [3].

V následujícím textu budou popsány aspekty nezbytné pro detekci STS a náběžné hrany a jejich vliv na bezpečnost a výkonnost systému.

Časové okno zpětného vyhledávání

Maximální špička korelovaného signálu nemusí vždy představovat přímou cestu mezi zařízeními. Přímá cesta mezi zařízeními nemusí existovat a na nepřímé cestě může

⁶Útok využívá předvídatelnost vnitřní struktury signálu symbolu a lze jej provést v závislosti na kódování dat a modulaci signálu. Útočník nejprve sbírá (učí se) hodnoty paketu/symbolu, aby je mohl využít později [3].

docházet k interferenci, která způsobí vyšší špičku, než je špička signálu přijatého přímou cestou. Z tohoto důvodu musí přijímače považovat jakýkoli vrchol nad hranici šumu jako možného kandidáta pro signál na odhad vzdálenosti. Jakmile přijímač detekuje špičku určité velikosti, dochází ke srovnání s ostatními špičkami v okolí, které pocházejí z jiné, kratší cesty. Časové okno pro hledání ostatních špiček se v literatuře označuje jako „back-search time window“ a pohybuje se v rozmezí od 100 do 300 ns, což je dáno rozptylem zpoždění kanálu UWB uvnitř a venku. Při detekci špičky určité velikosti tedy dochází k prohledávání okolí v čase 100–300 ns před touto (nejsilnější) špičkou, neboť jakýkoli signál z kratší cesty musí dorazit před ní.

Svou roli zde sehrává i správný odhad hladiny šumu. V ideálním případě jej přijímač získá statistikou velikosti šumu, když je kanál nevyužíván. Pokud je hladina nastavena příliš vysoko, může docházet k záměně vrcholů o menší amplitudě generovaných nepřímou cestou za šum. V opačném případě může být zase šum detekován jako špička, čímž může být ovlivněno určení času přijetí signálu [3], [9], [10].

Prahová hladina pro detekci špiček

Neméně důležitým parametrem je i prahová hladina pro detekci špiček. V důsledku vícecestného šíření signálu může vzniknout více korelačních špiček v různých časových okamžicích a různou amplitudou. Přijímač následně vypočítá špičkový výkon pro všechny korelační špičky, díky čemuž získá sadu dvojic: čas příchodu – výkon. Za účelem identifikace korelační špičky STS může přijímač vypočítat statistiku nad celou sadou dvojic či porovnat čas přijetí a špičkový výkon s předem definovanými prahovými hladinami. Aby mohl být dříve přijatý vrchol (v následujícím výčtu parametrů označen indexem x) identifikován jako pravý počátek přenosu STS, musí splňovat následující podmínky:

- Rozdíl špičkových výkonů nepřekoná hranici MPEP (maximal peak to earlier peak ratio) [3]:

$$\text{MPEP} > \frac{P_x}{P_h}. \quad (1.2)$$

- Špičkový výkon klasifikovaného signálu musí v dostatečné vzdálenosti od šumu. Tato podmínka je vyjádřena prostřednictvím parametru PAPR (peak to average power ratio) [3]:

$$\text{PAPR} \leq \frac{P_x}{P_{rms}}. \quad (1.3)$$

Kde P_h vyjadřuje špičkový výkon nejvyšší špičky korelovaného signálu, P_x špičkový výkon signálu, u kterého je prováděn test, zda se nejedná o dříve přijatý vrchol, a P_{rms} průměrný výkon kanálu (šumu).

Chyba měření vzdálenosti nastává, pokud přijímač vybere ze sady vrcholů ten špatný. Odhadnutá vzdálenost může být v závislosti na tom, zda přijímač vybral vrchol před nebo za prvním výskytem STS vrcholu, analogicky kratší či delší oproti skutečné vzdálenosti. Hlavní metrikou výkonnosti HRP přijímače je právě chyba měření při různých podmínkách kanálu (LoS/NLoS) [3].

1.4.3 Útoky

Následující kapitola se zaměřuje na vybrané typy útoků, které mohou ohrozit aplikace využívající měření vzdálenosti pomocí UWB technologie. Jsou popsány vybrané útoky, jejich principy a důraz je zaměřen zejména na způsob, jakým mohou útočníci konkrétní slabiny využít a manipulovat tak s měřením vzdálenosti.

Vylepšený útok cikád

Útok „Cicada++ Attack“ je vylepšenou verzí původního útoku cikád (Cicada attack). Tento typ útoku využívá slabiny v detekci náběžné hrany přijímačem. Systémy založené na měření ToA spoléhají k dosažení přesného odhadu vzdálenosti na mechanismy pro detekci náběžné hrany. Ve většině implementací přijímač provádí korelaci mezi přijatým a očekávaným signálem. Po nalezení nejvyššího vrcholu korelace se od tohoto vrcholu zpětně pokouší identifikovat náběžnou hranu nad určitým prahem šumu, a to z důvodu, že signál přicházející přímou cestou nemusí být vždy nejsilnější. Útoky tohoto typu využívají právě obtížnosti rozlišení rušení od signálu přicházejícího přímou cestou. Přítomnost útočného signálu narušuje detekci náběžné hrany, čímž zhoršuje výkonnost přijímačů a snižuje odhadnutou vzdálenost.

Vylepšený útok na rozdíl od původního vysílá pulsy s frekvencí, jež je zlomkem opakovací frekvence legitimního signálu. Útočník navíc vysílá oproti původnímu signálu pulsy s vyšší amplitudou. Aby byl útok úspěšný, musí protivník provést následující kroky:

- Pokud je výkon originálního signálu nízký, musí jej zesílit, aby byl po přijetí přijímačem vyhodnocen jako nejvyšší korelační špička.
- Útočník se musí synchronizovat s originálním signálem a vysílat své pulsy tak, aby se shodovali s vysíláním originální STS.
- Útočník potřebuje hrubý odhad síly originálního signálu (RSSI) na přijímači, aby odhadl, jakým výkonem má vysílat. Polarita útočných pulsů není relevantní a může být zvolena náhodně.

Výsledkem útoku je, že přijímač přijímá superpozici originálního a signálu útočníka. Přijímač stále přijímá velké části nezměněné originální STS (když se útočné vysílání neshoduje s originálním) a může tak stále úspěšně korelovat přijatý signál s lokální šablonou STS a pozorovat vysokou korelační špičku. Signál útočníka

vede k nárůstu vedlejších špiček a pro přijímač je obtížnější rozlišit, jestli jsou dřívější špičky generovány např. NLoS signálem nebo jsou způsobeny superponovaným útočným signálem [3], [10].

Adaptive Injection Attack

Cílem tohoto sofistikovaného útoku je umístit falešný signál do časového okna před originálním signálem. V kontextu tohoto útoku se předpokládá, že útočník může přeposílat nebo zablokovat přenos originálního signálu (například pokud jsou zařízení daleko od sebe bez přímé viditelnosti, nemohou komunikovat přímo, ale musí komunikovat přes prostředníka). Díky tomu může útočník selektivně vybírat část signálu, kterou chce přenést, a část signálu, kterou chce blokovat. Dalším předpokladem je schopnost útočníka ovládat impulsní odezvu kanálu na přijímači, například umístěním vlastního vysílače v jeho blízkosti. Útočník, stejně jako v případě útoku popsaném v části 1.4.3, vysílá pulsy s frekvencí, jež je zlomkem opakovací frekvence legitimního signálu. Poté, co útočník přenesl několik originálních pulsů a několik pulsů injektoval, koreluje své útočné pulsy s řetězcem pulsů superponovaných na část originálních impulsů s STS, kterou dosud obdržel. Díky tomu určí, zda může uspět s injekcí špičky do zamýšlené polohy v časovém okně zpětného vyhledávání. Pokud ano, tak přestane s přeposíláním originálního signálu a pokračuje ve vysílání útočných pulsů [3].

Útoky využívající tolerance nedokonalosti hodin

Zařízení využívají hodinový signál odvozený z krystalového oscilátoru k měření časových intervalů, vzorkování příchozích signálů a generování rádiové frekvence. Hodinový signál se vzhledem k nedokonalostem použitých součástek může měnit a s ním i všechny odvozené veličiny. Chyba, která nastává, když zařízení s nominální hodinovou frekvencí f_1 pracuje s hodinovou frekvencí f_2 , se označuje jako drift (posun) a vyjadřuje se v ppm (parts per million – části na milion). Norma připouští drift o maximálním rozsahu ± 20 ppm. Zařízení, které pracuje s negativním driftem, pracuje s frekvencemi pod jejich jmenovitými hodnotami – v případě čekání čeká déle, při měření času mezi 2 událostmi naměří menší hodnotu. Drift zařízení může v čase kolísat, ale během krátkého přenosu jej lze považovat za konstantní.

Přijímač a vysílač obecně vykazují rozdílné hodnoty driftu, což v důsledku způsobuje různé odhady časových intervalů a ovlivňuje přesnost měření vzdálenosti. Přijímač po příjmu signálu může odvodit drift hodin a provést korekci. Tohoto chování využívají následující útoky:

- Mix-Down Attack (MD) – předpokladem pro tento útok je schopnost útočníka manipulovat, blokovat, injektovat a odposlouchávat všechny bezdrátové

signály a současně je schopen nastavit výkon a časování všech svých signálů. Útočník manipuluje s frekvencí nosného signálu odpovědi. Iniciátor komunikace považuje vlastní hodiny za správné a musí odhadnout posun hodin vysílače (měřením nosné frekvence odpovědi), aby došlo ke kompenzaci rozdílu v driftech obou zařízení. Iniciátor upraví hodnotu ToF o kompenzaci, což vede ke změně určené vzdálenosti mezi zařízeními. Útok mění pouze nosnou frekvenci, nikoliv obsah zprávy. Z toho důvodu dokáže překonat i bezpečnostní mechanismy jako je STS a je proveditelný v režimu SS-TWR v rámci 802.15.4z i návrhu 802.15.4ab.

- Stretch and Advance (S&A) – útočník generuje zprávu, která je předsunutá originální zprávě a je časově natažená (stretch). Následně získá čas, aby se naučil a přehrál originální zprávu (advance). Výsledná zpráva bude vypadat jako by byla odeslána z legitimního vysílače s pomalejšími hodinami, což v důsledku sníží odhad vzdálenosti [11].

1.5 LoS a NLoS signál

Zásadním faktorem při určování výkonnosti UWB systému pro určení polohy či vzdálenosti je její přesný odhad. Během tohoto procesu systém získává čas letu signálu (ToA) díky známému času odeslání zprávy s využitím technik popsanych v kapitole 1.1. Díky rozlišení v řádu nanosekund mohou UWB zařízení získat správnou dobu šíření sledováním nejkratší cesty mezi UWB zařízeními při výměně zpráv.

V případě, že mezi vysílačem existuje přímá viditelnost (LoS – Line of Sight), se první přijatý signál považuje za signál přicházející nejkratší cestou, a doba letu získaná s využitím tohoto signálu je velmi blízko skutečné vzdálenosti mezi zařízeními.

V obecném vnitřním prostředí se mezi UWB zařízeními mohou vyskytovat různé překážky, jako jsou např. stěny, nábytek a lidé či jiné objekty. Signál se při šíření v takovém prostředí odráží od různých povrchů a překážek, případně jimi prochází, což vede k jeho útlumu. Velikost útlumu závisí na materiálu a tloušťce překážky, kterou signál prochází, a také na frekvenci UWB signálu. Různé kanály mohou mít různé útlumové charakteristiky, ale obecně jsou na útlum náchylnější kanály s vyšší frekvencí.

V případě odrazu od překážek může nastat efekt vícecestného šíření, při kterém přijímač obdrží více kopií stejného signálu s různými zpožděními, navíc se poloha těchto překážek může v čase rychle měnit. Jev může být výrazně zesílen při nasazení v průmyslovém prostředí s přítomností kovových materiálů, kde dochází k hustému vícecestnému šíření (dense multipath scattering). Tyto skutečnosti činí měření méně

stabilním a ovlivňují jeho přesnost. Signál, který je odražen, má delší dráhu než signál přímý, což vede k jeho zpoždění a pozitivnímu zkreslení při měření vzdálenosti. Chyby způsobené odraženým signálem jsou nahodilé a nelze je tak snadno předpovídat a eliminovat. Takový signál je v literatuře označován jako NLoS (Non-Line of Sight). Při správné identifikaci NLoS signálu lze aplikovat metody korekce chyb a výrazně zvýšit přesnost lokalizace, což spočívá zejména v identifikaci signálu přicházejícího přímou cestou popsané v kapitole 1.4.2 [4], [5], [12], [13].

2 Strojové učení

Strojové učení (ML – Machine Learning) je podmnožinou vědní disciplíny označované jako umělá inteligence (AI – Artificial Intelligence), jejímž cílem je poskytnout algoritmu (stroji, počítači) schopnost vykonávat intelektuální úkoly, které obvykle vykonávají inteligentní bytosti (lidé). Do oboru umělé inteligence lze zařadit mnoho přístupů, z nichž některé nemají s učením nic společného [14], [15]. To ale neplatí o strojovém učení, jehož cílem je vybavit programy schopností učit a přizpůsobovat se. Hlavním úkolem je odvodit z dat, které reprezentují zkušenosti, modely, jež dokážou detekovat určité složité skryté vzory. Algoritmus vidí vstupní data a odpovídající výstupní hodnoty a snaží se odvodit nějaká pravidla (skryté vzory), pomocí kterých lze vstupní data namapovat na odpovídající výstupní hodnoty. Pokud jsou modelu poskytnuta nová vstupní data, jeho výstup je výsledkem odvození, které vyplývá ze skrytých vzorů (pravidel) zachycených modelem během svého učení [14], [16].

Existuje celá řada aplikací využívajících strojové učení. Jedná se například o systémy rozpoznávání řeči a ručně psaných textů, spamové filtry, identifikátory obsahu, klasifikátory obrázků, systémy automatického titulkování či různé doporučovací systémy [17]. Velmi využívané jsou i v oblasti chemie, astronomie, zemědělství, lékařství, modelování geologických a environmentálních procesů a ve spoustě dalších [15]. Například v lékařství napomáhají k určení diagnózy a životních faktorů pacienta z elektronických zápisů psychologa a velmi často se buď již používají, nebo se jejich použití zvažuje k profilování rizika budoucího výskytu nemoci u konkrétního pacienta, díky čemuž se lze na tyto vysoce rizikové pacienty zaměřit dříve a riziko vzniku nemoci u nich snížit, oddálit či jej zcela eliminovat [18].

Na základě informací o učení nebo označení dat používaných k jeho učení, lze metody strojového učení rozdělit na dvě kategorie – učení s učitelem a učení bez učitele. V prvním případě jsou data použita k učení modelu předem označena a jedná se především o úlohy klasifikace¹ a regrese². Cílem těchto metod je naučit se nějaká obecná pravidla pro mapování vstupních dat na známou výstupní hodnotu. Pokud jsou modelu poskytnuta neoznačená vstupní data, dokáže pomocí naučených pravidel určit hodnotu výstupu. Ve druhém případě pro učení nejsou potřeba předem označená data a cílem je naučit se o jejich distribuci. Jedná se například o úlohy takzvaného shlukování³ (clustering), komprese⁴ či extrakce příznaků [16], [17], [18], [19].

¹Cílem je určení, do které z předdefinovaných tříd (kategorií) spadá konkrétní datový vzorek [16], [18].

²Cílem predikce je na rozdíl od klasifikace spojitá hodnota [16], [18].

³Cílem metody je identifikace relevantních podmnožin v rámci souboru dat, aniž by byly předem stanoveny společné vlastnosti, které by prvky jednotlivé skupiny mohly mít [19].

⁴Metoda sloužící k redukci dimenze vlastností vzorku v rámci datasetu [19].

Existuje však mnoho dalších kategorií učení. Jedná se například o hluboké učení, přenesené učení, zpětnovazební učení⁵ a mnoho dalších [16], [17].

2.1 Data

Strojové učení pracuje s daty a datasety (sady dat), které představují fyzickou reprezentaci informace. Dataset se skládá z množiny datových bodů, jenž jsou také označovány jako vzorky. Vzorky reprezentují libovolné entity (například pacienta v nemocnici, signál, ...), které jsou předmětem analýzy. Každý vzorek je složen z množiny vlastností (atributů), které nějakým způsobem popisují entitu. Atributy mohou nabývat numerických či kategorických hodnot a množina hodnot vlastností formuje vektor [16], [19].

V publikaci [16] jsou pro charakteristiku datasetu vymezeny 3 základní vlastnosti:

- Kvalita – hraje klíčovou roli v algoritmech strojového učení. Reálná data obsahují chyby vzniklé při jejich sběru, zpracování či uchovávání z celé řady důvodů. Může se jednat o šum, nekonzistentní hodnoty, chybějící data, duplicitní hodnoty, a mnoho dalších. Jakákoliv chyba v datech degraduje kvalitu celého datasetu a může negativně ovlivnit výkonnost učení.
- Rozměrnost – udává počet atributů, které se podílejí na popisu entity v rámci každého vzorku. Závisí jak na datasetu, tak na metodě zpracování dat. Datasety s větší dimenzí jsou mnohem náročnější na zpracování a často se na ně aplikují techniky pro snížení dimenze (dimension reduction) a extrakci příznaků (feature selection).
- Hustota – v mnoha případech obsahují vzorky v datasetu atributy s nulovými hodnotami. V aplikacích s velkým počtem atributů je při popisu entity velmi často účinně využíváno jen malé množství atributů (např. u transakčních dat, uspořádaných dat a dat sekvenčních transakcí). Problémem je ale i opačný jev, který představuje velké množství dat s velmi malou hustotou – velké množství dat je popsáno relativně malým počtem atributů. Takovým problémům se věnuje samostatná kategorie úloh označovaná jako řídké učení (sparse learning) [16].

Dle publikace [19] by vzorky pro algoritmy ML měly být v ideálním případě nezkrácenou a náhodnou podmnožinou všech vzorků. Jako příklad takového zkrácení

⁵Na rozdíl od metod učení s nebo bez učitele se modely učí optimálně reagovat s neznámým či proměnným prostředím. Model vykonává akce a pozoruje reakce prostředí, ze kterých získává hodnotící zpětnou vazbu, kterou používá k odhadu odměn za provedené akce i nákladů na jejich provedení. Využívá se v oblastech, kde je nezbytné řešit sekvenční rozhodovací problémy, jako jsou např. automatické řízení, plánování procesů a logistika [17].

udává model pro rozeznání fotografie vlka od plemene husky na základě zvířecích charakteristik, který ale v důsledku zkreslení vstupních dat místo zvířat identifikuje stopy ve sněhu na fotografii [19].

Dalším z možných problémů pro algoritmy strojového učení je nevyváženost dat. To znamená, že 1 nebo více tříd je v rámci datasetu nedostatečně zastoupena. Řešením může být například nadvzorkování menšinové či podvzorkování většinové třídy v rámci datasetu [19].

2.1.1 Transformace a předzpracování dat

V následující části budou představeny některé operace pro předzpracování a transformaci dat, jejichž cílem je upravit vstupní data učicího algoritmu tak, aby měla vhodnou podobu, která umožní algoritmu dosahovat lepších výsledků [16].

Čištění a předzpracování dat

Techniky čištění a předzpracování dat slouží ke zvýšení kvality těchto dat. Jevy, které snižují kvalitu dat, byly popsány v části 2.1.

Šum je velmi obtížné detekovat od dat, která nejsou šumem, proto se existující práce zabývají spíše zvýšením odolnosti vůči šumu u učicích algoritmů.

Oproti tomu detekce odlišných dat se stala hlavní oblastí výzkumu v mnoha aplikacích, jako např. detekce spamu a detekce narušení sítě, kdy je cílem detekovat vzorky s neobvyklými hodnotami atributů [16].

Chybějící hodnota může pro některé algoritmy představovat obzvláště velký problém. Existuje řada technik, kterými lze tento problém řešit [16], [19]:

- Odstranění vzorku z datasetu – nejjednodušší řešení, které ale zároveň může z datasetu odstranit důležitý vzorek [16].
- Odhad hodnoty – hodnotu lze odhadnout z hodnot ostatních vzorků, u kterých je tato hodnota definována. Může jít např. o průměr hodnot či nejčastější hodnotu. V některých případech může způsobit jev, kdy je model příliš naučený na testovací data a na jiných datech obsahuje výrazně nižší přesnost (overfitting) [16], [19].
- Ignorace vzorku – vyžaduje robustní učicí algoritmus, který se neobejde bez nezbytných kalibrací, jenž jsou mimo rozsah předzpracování dat [16].

Agregace a vzorkování dat

Pro algoritmy s vysokou časovou náročností není vhodné používat rozsáhlý dataset s velkým množstvím vzorků. K uzpůsobení datasetu pro použití k učení prostřednictvím časově náročných algoritmů se data agregují nebo vzorkují. Při agregaci dat

je kombinováno několik vzorků do jednoho. Cílem je redukovat velikost dat a časovou náročnost jejich analýzy, aby na jejich analýzu mohly být použity právě tyto (časově) náročné algoritmy. Agregace je velmi užitečná u dat majících hierarchickou strukturu. Nevýhodou agregace je, že může dojít ke ztrátě informací ze vzorků dat ve spodní části hierarchie.

Druhým způsobem redukce velkého množství dat je vzorkování, kdy je pomocí vzorkovacích metod vybrána podmnožina dat z celého datasetu, což umožňuje efektivní analýzu dat v krátkém časovém horizontu. Vybraná podmnožina vzorků by měla zachovávat původní vzorce rozložení dat, aby byly pokryty vlastnosti původního souboru dat. Existují 2 hlavní přístupy ke vzorkování:

- Náhodné vzorkování – nejjednodušší přístup, který vybírá vzorky náhodně. Funguje dobře pro většinu vyvážených datasetů (mají vyvážený počet vzorků pro všechny třídy).
- Stratifikované vzorkování – využívá se u nevyvážených datasetů. Vzorky jsou vybírány separátně z každé třídy zastoupené v datasetu, což zajišťuje zastoupení vzorků každé třídy v konečném výběru [16].

Redukce dimenze dat a výběr atributů

Velké množství atributů popisujících vzorky v rámci datasetu může způsobit, že mnoho algoritmů pro učení nebude fungovat, protože jakékoliv navýšení dimenze dat je činí hůře zpracovatelnými. Pro snížení počtu atributů popisujících vzorky slouží techniky pro snížení dimenze a výběr atributů.

První z technik využívá pro snížení dimenze například techniky lineární algebry a statistické metody, ale i spoustu dalších jiných metod. Druhá z technik vybírá podmnožinu atributů, která bude sloužit pro reprezentaci jednotlivých vzorků. Mezi originálními atributy se mohou vyskytovat i redundantní nebo irelevantní atributy, které nejsou pro úlohy strojového učení podstatné. Metody výběru atributů se dělí do 3 skupin:

- Aplikace filtru – před začátkem učícího algoritmu jsou předem vybrány atributy, které jsou nezávislé na učících úlohách.
- Vestavěné přístupy (Embedded Approaches)– některé modely mají schopnost provádět výběr příznaků samostatně.
- Obalový přístup (Wrapper Approaches)– mnoho přístupů používá model pro nalezení nejlepší podmnožiny atributů, která je nejvíce užitečná pro algoritmy strojového učení [16].

Transformace dat

Velmi často nemůže vstupní soubor dat splňovat požadavky některých algoritmů, a proto je potřeba provést jejich transformaci. Operace transformace dat zahrnují binarizaci, diskretizaci a normalizaci dat.

Binarizace se využívá pro kategorické atributy, které algoritmus nejčastěji vyžaduje vyjádřené v binární podobě. Pokud tedy v rámci datasetu existuje atribut s kategorickou hodnotou, která může nabývat m hodnot, lze tuto hodnotu kvantifikovat například binárním kódem o délce n , kde n je nejmenší celé číslo splňující podmínku $n \geq \log_2 m$. Druhým a více používaným způsobem je vyjádřit hodnoty binárním kódem o délce m , neboť oproti prvním způsobu jsou kódy jednotlivých hodnot více „nezávislé“ (pro $m = 5$ např. „001“ a „101“ oproti „00001“ a „10000“). Nedostatek druhého způsobu leží na délce binárního kódu, která je velmi vysoká pro atributy s velkým množstvím hodnot.

V mnoha případech je nezbytná transformace spojitých dat na diskrétní hodnoty, které jsou jednodušší pro klasifikaci, k čemuž se využívá proces **diskretizace** dat. Dojde k určení dělicích bodů k rozdělení rozsahu hodnot na několik intervalů a hodnoty atributů budou vyjádřeny konkrétní diskrétní hodnotou.

Ke standardizaci rozsahu nezávislých proměnných nebo atributů dat se využívá **normalizace**. Pro některé algoritmy strojového učení může být velmi náročné vyřešit daný problém, pokud je rozsah hodnot atributů značně odlišný. Nejjednodušší metodou normalizace je metoda Min-Max:

$$x_n = \frac{x - x_{max}}{x_{max} - x_{min}}. \quad (2.1)$$

Kde x představuje hodnotu atributu, x_n normalizovanou hodnotu atributu, x_{min} a x_{max} nejnižší a nejvyšší hodnotu daného atributu [16].

2.2 Klasifikace

Jak už bylo zmíněno v kapitole 2, řešení klasifikačních úloh spadá do kategorie metod učení s učitelem. U klasifikačních úloh je cílem určení, do které z předdefinovaných tříd (kategorií) spadá konkrétní datový vzorek. Datová sada určená ke tréninku modelu je u těchto metod rozdělena na 3 části:

- Trénovací sada – určena přímo k trénování modelu.
- Validační sada (volitelná) – určena k hodnocení modelu během tréninku pro vylepšování modelu a ladění hyperparametrů⁶.
- Testovací sada – určená k evaluaci naučeného (natrénovaného) modelu [16], [18].

⁶Parametry řídící proces učení modelu [16], [18].

Pro rozdělení datasetu na výše zmíněné části lze využít různé metody a mezi nejnámější patří vícenásobný náhodný výběr (multiple random sampling) a křížové ověřování (cross validation). Metoda vícenásobného náhodného výběru generuje 2 podmnožiny vzorků pro trénování a validaci modelu a zbylé vzorky jsou určeny pro jeho testování. Aby se předešlo tomu, že bude trénovací sada příliš malá na to, aby reprezentovala celý soubor dat, je tento proces proveden vícekrát, díky čemuž jsou pokaždé vytvořeny jiné trénovací a testovací množiny. V případě, že jsou k dispozici pouze malé datasety, se velmi často využívá metoda křížového ověřování. Existuje více typů této metody, nejčastěji používanou je tzv. „ k -násobné křížové ověřování“ (k-fold cross validation), v rámci které je dataset rozdělen na k stejně velkých podmnožin. Takový proces probíhá právě k -krát a v každé iteraci může být jako testovací množina zvolena jiná podmnožina dat [16].

Model naučený na nějakých historických datech může být dále aplikován na klasifikaci tříd pro budoucí neoznačená data [16], [18]. Cílem této práce je také ověřit existující modely, proto je nezbytné se věnovat i analýze metod pro klasifikační úlohy s využitím strojového učení. Mezi takové metody patří např. rozhodovací stromy, metoda podpurných vektorů a umělé neuronové sítě. Zmíněné metody budou podrobněji popsány v následujícím textu.

2.2.1 Rozhodovací strom

Jedním z klasických klasifikačních algoritmů je tzv. rozhodovací strom (decision tree), který je založen na sérii rozhodnutí, která musí algoritmus učinit předtím, než dosáhne konečného rozhodnutí o klasifikaci. Naučený model může být reprezentován jako diagram se stromovou strukturou, kde každý uzel reprezentuje otázku a každá větev představuje rozhodovací možnost. Rozhodovací stromy jsou jedny z nejpožívanějších algoritmů strojového učení pro klasifikaci díky jejich jednoduchosti, efektivitě a účinnosti.

Rozhodovací strom při trénování modelu, což je v případě tohoto algoritmu rekurzivní proces, rozděluje instance do stále podobnějších skupin na základě jejich atributů [16], [18]. Taková strategie je označována jako strategie „rozděl a panuj“ (divide and conquer) [16]. Princip fungování algoritmu lze ilustrovat na příkladu, který se zabývá binární klasifikací⁷ slonů, a byl převzat z [16]. Vzorky datasetu popisují slony a jejich zjištěné vlastnosti, jako je váha, výška, typ kůže (vrásčitá, hladká), velikost uší (malé, velké), počet výběžků na konci chobotu (v [16] označeno jako „trunk finger“) a cílem je určit, zda daný vzorek náleží slonu africkému nebo asijskému. Na základě váhy lze datové instance rozdělit na 2 podmnožiny – první podmnožinou

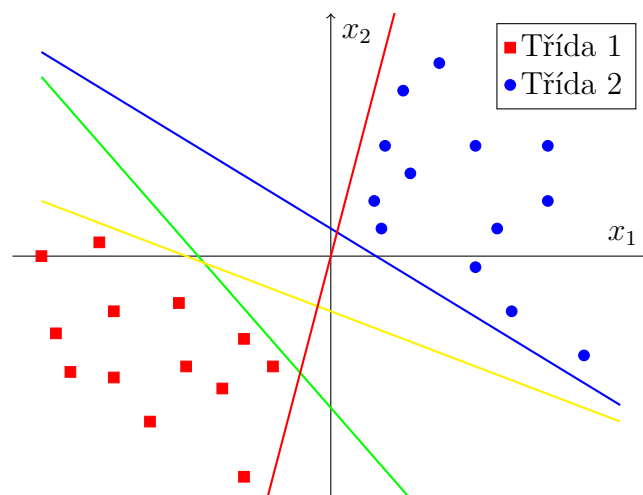
⁷(Přiřazení datového vzorku do jedné ze 2 kategorií (LoS a NLOS, 0 a 1, pozitivní a negativní) [16].

jsou vzorky s váhou 6 a více tun, kam na základě údajů ze vstupního datasetu patří pouze sloni afričtí, zatímco druhou podmnožinu tvoří vzorky s váhou pod 6 tun, kam patří oba druhy slonů. Vzorky ze druhé podmnožiny je z toho důvodu nutné dále dělit (podle výšky, počtu výběžků na chobotu), aby byl model schopen přesně určit druh slona.

Varianta algoritmu označovaná jako „multivariační rozhodovací strom“ (Multivariable decision tree) v testech v rámci rozhodovacích uzlů používá více proměnných. Zatímco se rozhodovací strom s jednou proměnnou v každém rozhodovacím uzlu snaží vybrat optimální atribut pro rozdělení, cílem této varianty rozhodovacího stromu je nalezení optimálního lineárního klasifikátoru (optimální váhy a prahovou hodnotu) [16].

2.2.2 Metoda podpůrných vektorů

Známým klasifikačním algoritmem je i metoda podpůrných vektorů SVM (Support Vector Machine). Datové instance jsou reprezentovány jako body v prostoru příznaků, které lze rozdělit do 2 tříd pomocí hyperplochy, kterou se model naučí. Nové datové vzorky budou mapovány do stejného prostoru a podle toho, na jakou stranu rozhodovací hranice spadají, zařazeny do tříd. Algoritmus se snaží o dosažení co největší mezery mezi rozhodovací hranicí a jednotlivými datovými body, díky čemuž jsou minimalizovány chyby při klasifikaci trénovacích dat [16], [20]. Významnou výhodou SVM je, že během učení získává množinu podpůrných vektorů, jak lze vidět na obrázku 2.1. Podpůrné vektory představují daný klasifikační úkol a jsou tvořeny malou datovou sadou [20].



Obr. 2.1: Znázornění principu algoritmu SVM.

Algoritmy SVM dosahují vysoké přesnosti při klasifikaci a jsou velmi odolné vůči přeučení, a to zejména při práci s malými datovými sadami. Při využití různých jader je lze využít i nelineárně separovatelná data. Při použití velkých datových sad je trénování SVM modelů velmi časově i paměťově náročné. Dalším problémem může být výběr optimálních parametrů a jádra. Výkon algoritmu může být významně ovlivněn, pokud mu jsou pro trénink předloženy nevyvážené datové sady.

Využití algoritmů SVM je velmi široké a hrají významnou roli v rozpoznávání vzorů. Jedná se například o klasifikaci obrazu, analýzu genetických dat (např. struktura proteinů, identifikace genů rakoviny) v bioinformatice a existuje i návrh systému pro zvýšení výkonnosti systémů rozpoznání tváře [20].

2.2.3 Umělé neuronové sítě

Umělé neuronové sítě jsou výpočetní algoritmy, které se snaží modelovat způsob, jakým řeší problémy lidský mozek, pomocí velkého množství propojených umělých neuronů. Lidský mozek je složen ze skupiny neuronů, které jsou vzájemně propojeny, a pomocí těchto propojení mohou přecházet signály z jednoho neuronu na druhý. Nové znalosti se učí prostřednictvím změny síly spojení jednotlivých neuronů. Na tomto principu fungují i umělé neurony, ze kterých jsou složeny umělé neuronové sítě. Výstupní hodnota je dána organizací a váhami spojení mezi umělými neurony [16], [19]. Umělé neurony ve vstupní vrstvě přijímají vstupní data a přenášejí je do dalších neuronů v první skryté vrstvě pomocí vážených spojů. Ve skryté vrstvě dochází k matematickému zpracování dat a přenášení výsledků do dalších skrytých vrstev. Výstup sítě je zprostředkován neurony v poslední vrstvě [19], [21].

Umělý neuron

Umělé neurony přijímají vstupy od ostatních neuronů a v závislosti na hodnotách vstupních signálů x_i a prahové hodnotě (odsazení) θ mohou být aktivovány k přenosu informací na jiný neuron nebo zůstat neaktivní. [16]. Spojení mezi umělými neurony má určitou váhu (w_i), která určuje sílu (váhu) přijatého signálu [21]. Výstup neuronu je vyjádřen rovnicí 2.2, kde funkce f je označována jako aktivační funkce. Učení celé neuronové sítě spočívá v nastavování vnitřních vah spojení mezi neurony tak, aby výstup celé sítě odpovídal požadovanému výstupu pro konkrétní datové vzorky [16], [21].

$$y = f \left(\sum_{i=1}^n w_i x_i - \theta \right) \quad [16]. \quad (2.2)$$

Aktivační funkce

Aktivační funkce v neuronových sítích slouží k transformaci vstupního signálu na výstupní, který bude jako vstup předán další vrstvě. Jakmile neuron vypočítá sumu součinu vstupních signálů a vah upravenou o prahovou hodnotu, je tento dílčí výsledek předán aktivační funkci, která jej upraví tak, aby jej mohl zpracovat další neuron, a jejíž výstup je zároveň výstupem neuronu.

Bez použití aktivační funkce by výstupním signálem byla lineární funkce (polynomiální funkce 1. řádu). Taková síť by sice byla velmi jednoduchá, ale nebylo by možné ji nijak naučit a rozpoznávat komplexní vzory ve vstupních datech. Proto je vhodné, aby v rámci neuronové sítě existovala nelineární funkce, která neuronové síti umožňuje naučit se a vyjádřit jakoukoliv funkci, která je síti předložena. Takovou nelineární funkci reprezentuje většina běžně využívaných aktivačních funkcí. Volba aktivační funkce (případně aktivačních funkcí – v případě, že má neuronová síť více vrstev s různými aktivačními funkcemi) je velmi důležitá v kontextu přesnosti výsledného modelu. Mezi nejčastěji využívané aktivační funkce patří [22]:

- Binární kroková funkce – nejjednodušší aktivační funkce. Nelze použít pro klasifikaci více než 2 tříd. Tato funkce také nemá derivace a její gradient je tedy nulový, kvůli čemuž neumožňuje zpětnou propagaci chyb. Definice funkce je vyjádřena předpisem 2.3 ([22]):

$$f(x) = \begin{cases} 1, \text{ pro } x \geq 0 \\ 0, \text{ pro } x < 0 \end{cases}. \quad (2.3)$$

- Sigmoid – nejpoužívanější aktivační funkce. Protože není symetrická podle nuly, tak transformuje výstup neuronu do intervalu od 0 do 1, což znamená, že všechny výstupní hodnoty budou mít stejné znaménko. Lze ji vyjádřit předpisem 2.4 ([22]):

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (2.4)$$

- Hyperbolický tangens (\tanh) – funkce je velmi podobná sigmoidu, ale je symetrická podle nuly, kvůli čemuž mohou mít výstupní hodnoty různá znaménka. Výstup neuronu mapuje do hodnot v rozsahu -1 a 1. Tato aktivační funkce bývá preferovanější před sigmoidem, neboť gradient je v jejím případě strmější a neomezují se na změny v určitých směrech. Funkce je definována předpisem 2.5 ([22]):

$$f(x) = 2 \cdot f_{sigmoid}(2x) - 1 = \frac{2}{1 + e^{-2x}} - 1. \quad (2.5)$$

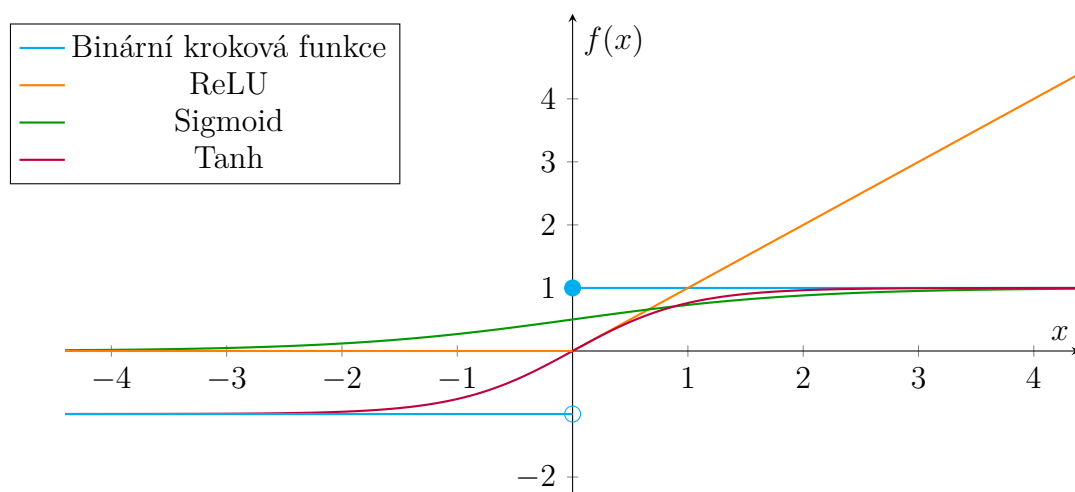
- Usměrněná lineární funkce (ReLU – REctified Linear Unit) – široce používaná aktivační funkce v neuronových sítích. Oproti ostatním aktivačním funkcím je

efektivnější, neboť její výstup zapříčiňuje, že nejsou aktivovány všechny neurony. V některých případech je derivace této funkce 0, což znamená, že i její gradient je 0 a hodnoty vah a aktivačních prahů nejsou aktualizovány během algoritmu zpětného šíření chyb. Funkce je definována předpisem 2.6 ([22]). Měla by být použita pouze ve skrytých vrstvách a nikoliv ve výstupní vrstvě. Existují i varianty této funkce, které se liší oborem hodnot v záporné části definičního oboru a jsou definovány předpisem 2.7 ([22]). Jedná se o parametrickou usměrněnou lineární funkci (PReLU), která ve své definici obsahuje parametr α , jenž je uplatněn právě pro zápornou část definičního oboru funkce. Varianta funkce s hodnotou parametru $\alpha = 0,01$ se označuje jako „leaky ReLu“.

$$f(x) = \begin{cases} x, & \text{pro } x \geq 0 \\ 0, & \text{pro } x < 0 \end{cases}. \quad (2.6)$$

$$f(x) = \begin{cases} x, & \text{pro } x \geq 0 \\ \alpha x, & \text{pro } x < 0 \end{cases}. \quad (2.7)$$

Průběhy výše popsaných aktivačních funkcí jsou znázorněny na obrázku 2.2. Podle publikace [22] dosahují klasifikační úlohy se sigmoidní aktivační funkcí (ve výstupní vrstvě) lepších výsledků, ale zároveň tyto aktivační funkce nejsou vhodné pro skryté vrstvy z důvodu problému s mizejícím gradientem⁸ [22].



Obr. 2.2: Průběhy vybraných aktivačních funkcí.

⁸Gradienty se při zpětném šíření vrstvami sítě často stávají velmi malými. Je to způsobeno tím, že se mezi sebou násobí velmi malá čísla (menší než 1). Parametry se v dřívějších vrstvách (vrstvy blíže výstupu) učí pomaleji a proces učení se stává neefektivním [23].

Architektury umělých neuronových sítí

Existuje celá řada modelů (architektur) umělých neuronových sítí, které se liší vykonávanými operacemi a množinou parametrů nezbytných k určení výstupu. Všechny tyto modely vycházejí ze 3 základních typů neuronových sítí, které budou postupně popsány v následujícím textu [15], [21].

Vícevrstvá dopředná neuronová síť

Jedná se o nejjednodušší formu umělé neuronové sítě. Vstupní data procházejí sítí pouze v 1 směru – od vstupní vrstvy k výstupní vrstvě. Model může i nemusí mít skryté vrstvy. Aktuální stav systému není definován žádným minulým stavem, představují tedy systém bez paměti [21].

Tato architektura se využívá v počítačovém vidění a rozpoznání řeči, hojně je využívána obecně pro klasifikaci i regresi. Výhodou architektury je snadná správa, nevýhodou naopak citlivost na data ovlivněná šumem [15], [21].

Rekurentní neuronové sítě

Rekurentní neuronové sítě (RNN) mohou jako vstupní data využívat sekvenci signálů různých délek. Tyto sítě zohledňují sekvenční závislosti mezi jednotlivými vzorky v rámci sekvence a nacházejí uplatnění především tam, kde hraje roli pořadí nebo časová posloupnost vstupních dat. Aktuální stav sítě je definován kombinací předchozího stavu a aktuálního vstupu, jedná se tedy o síť s pamětí. Jednotlivé váhy udávají, jak daleko v rámci sekvence se má síť dívat.

V závislosti na sekvenci vstupních a výstupních dat mohou být tyto sítě dále členěny do dalších tříd, jako je např. „jeden k mnoha“ (one-to-many) či „mnoho k jednomu“ (many-to-one), což je typ sítě využívaný právě pro klasifikaci – mapuje sekvenci vstupních dat na 1 výstupní hodnotu.

Do rodiny rekurentních neuronových sítí patří také velmi známá LSTM (Long Short-Term Memory) architektura. LSTM sítě uchovávají a zapomínají informace zavedením tzv. „gating units“ (hradla). V rámci LSTM existují celkem 3:

- Vstupní hradlo, které rozhoduje o tom, zda aktuální vstupní signál bude zahrnut do aktualizace stavu sítě.
- Zapomínací hradlo má na starosti kopírování nebo vymazání stavu sítě.
- Výstupní brána, jež rozhoduje o tom, zda se výstupní informace vrátí zpět do systému.

V každém časovém kroku je určeno, jaké množství informace z předchozího kroku a ze vstupního signálu má být uchováno. Systém se učí dlouhodobé časové závislosti

mezi vstupními daty uchováváním informací. Zároveň se ale musí naučit občas vymazat informace o svém aktuálním stavu, čímž je řešen problém s mizejícím gradientem [15], [19].

Konvoluční neuronové sítě

Poslední ze základních architektur umělých neuronových sítí jsou konvoluční neuronové sítě (CNN). Vynikají schopností najít komplexní vzory ve vstupních datech a jsou velmi využívány pro zpracování a klasifikaci obrazu a signálu. Největším přínosem CNN je redukce počtu parametrů umělé neuronové sítě, což umožňuje řešení komplexních úloh, jež kvůli obrovskému množství parametrů není možné řešit standardními architekturami umělých neuronových sítí. Konvoluční neuronová síť se skládá ze 3 vrstev:

- Konvoluční vrstva, která se skládá z filtrů (čtvercové matice), jež nad vstupními daty provádějí diskrétní konvoluci. Filtr slouží k tomu, aby byly ze vstupních dat extrahovány příznaky (např. hrany z obrazu). Filtry jsou kromě svého rozměru definovány také krokem a výplní. Jak moc se bude filtr posouvat po vstupních datech určuje právě parametr krok. Výstup konvoluce bude mít menší rozměry, než vstup. Pokud je žádoucí, aby měl výstup stejné rozměry, jako vstup, je nezbytné použít výplň.
- Sdružovací vrstva, která se stará o podvzorkování vstupních dat za účelem snížení složitosti pro další vrstvy. Nejčastější metodou sdružování je tzv. „max pooling“, při kterém jsou vstupní data rozdělena na submatice (velmi často 2x2), ze kterých je vybrán vzorek s nejvyšší hodnotou. Submatice se mohou vzájemně překrývat. Na této vrstvě dochází ke ztrátě informací.
- Plně propojená vrstva, která odpovídá tradičním neuronovým sítím. Každý neuron v plně propojené vrstvě je přímo spojen s neurony předchozí i následující vrstvy [15], [21], [24].

2.2.4 Hodnocení výkonnosti modelu

V této kapitole budou představeny používané techniky pro hodnocení natrénovaných modelů pro binární klasifikaci, aby bylo možné hodnotit jejich kvalitu a vzájemně je porovnat.

Pro hodnocení modelů určených k (binární) klasifikaci se vychází z tzv. „matice záměny“ (confusion matrix). Při binární klasifikaci je výstupem modelu např. logická „0“ (negativní výstup) a logická „1“ (pozitivní výstup). Matice se skládá ze 4 hodnot:

- True Positive (TP) – počet správně klasifikovaných pozitivních vzorků.
- False Positive (FP) – počet nesprávně klasifikovaných pozitivních vzorků.

- True Negative (TN) – počet správně klasifikovaných negativních vzorků.
- False Negative (FN) – počet nesprávně klasifikovaných negativních vzorků.

Z hodnot této matice jsou odvozeny následující metriky pro hodnocení modelu:

- Přesnost (Accuracy) – odpovídá počtu správně předpovězených hodnot vůči celkovému počtu předpovězených hodnot (rovnice 2.8) [16], [19].

$$\text{Přesnost} = \frac{TP + TN}{TP + FN + FP + TN} \quad (2.8)$$

- Preciznost (Precision) – odpovídá poměru správně předpovězených pozitivních hodnot vůči celkovému počtu pozitivních předpovězených hodnot (rovnice 2.9) [16], [19].

$$\text{Preciznost} = \frac{TP}{TP + FP} \quad (2.9)$$

- Citlivost (Sensitivity nebo také Recall) – odpovídá poměru správně předpovězených pozitivních hodnot vůči celkovému počtu pozitivních hodnot v datasetu (rovnice 2.10). Tento parametr je označován i jako TPR (True Positive Rate). Pro negativní hodnoty nese parametr označení FPR (False Positive Rate) [16], [19].

$$\text{Citlivost} = \text{TPR} = \frac{TP}{TP + FN} \quad (2.10)$$

- Vyvážená přesnost (Balanced Accuracy) – vyjadřuje průměrnou schopnost modelu správně rozpoznat pozitivní i negativní případy. Počítá se jako aritmetický průměr TPR a FPR.
- Skóre F_β – v praxi se používá $\beta = 1$. Parametr je definován rovnicí 2.11 a zohledňuje preciznost i citlivost modelu. Užitečný je zejména v případech nevyvážené datové sady a používá se pro vyjádření rovnováhy mezi precizností a citlivostí. Volba β umožňuje volit větší důraz na jeden z parametrů [16].

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Preciznost} \cdot \text{Citlivost}}{\beta^2 \cdot \text{Preciznost} + \text{Citlivost}} \quad (2.11)$$

- ROC (Receiver Operating Characteristic) křivka – znázorňuje závislost TPR na FRP při různé velikosti rozhodovací úrovně. Ideální je stav, kdy jsou všechny hodnoty klasifikovány správně
- ROC-AUC (Receiver Operating Characteristic-Area Under the Curve) – plocha pod ROC křivkou shrnuje schopnost modelu správně rozlišit mezi 2 třídami. Představuje pravděpodobnost, že náhodně vybraný pozitivní vzorek bude klasifikátorem ohodnocen vyšší pravděpodobností než náhodně vybraný negativní vzorek. Vyšší hodnota znamená vyšší klasifikační výkon, a zároveň hodnota této metriky není závislá na rozhodovacím prahu a hodnotí tak model globálně [16], [19], [25].

2.2.5 Generalizace

Většina aplikací modelů umělé inteligence předpokládá, že zdrojová data (pro trénink) a cílová data, se kterými model pracuje, jsou nezávislá a identicky rozdělená. Tento předpoklad ale velmi často neplatí v reálných prostředích, neboť mezi zdrojovými a cílovými daty z jiných oblastí existují doménové či distribuční posuny. Modely následně vykazují sníženou výkonnost při nasazení v podmínkách odlišných od těch tréninkových. Doménová generalizace (DG) řeší tento problém tak, že se snaží učit modely na různých a odlišných zdrojových datech se schopností generalizace na nové, dosud neznámé domény bez nutnosti dalšího doladování modelu. Některé publikované metody využívají právě metod doménové generalizace, a z toho důvodu budou v následujícím textu stručně představeny 3 hlavní oblasti technik doménové generalizace.

Manipulace s daty

Tato technika sdružuje metody pro manipulaci se vstupními daty pro trénink modelů umělé inteligence. Hlavním cílem těchto metod je zvýšit množství a zlepšit rozmanitost vstupních dat pro lepší schopnost generalizace. Metody pro manipulaci s daty se dělí do 2 skupin:

1. Generování nových vzorků dat s využitím generativních modelů umělé inteligence, jako jsou **variační autoenkodéry** (VAE – Variational autoencoder) a **generativní adversariální sítě** (GAN – Generative Adversarial Network).
2. Rozšiřování dat zahrnující geometrické a barevné transformace, náhodné mazání či permutace s cílem zvýšení rozmanitosti dat. Jsou zaměřeny primárně na počítačové vidění a vizuální aplikace, ale mohou být přizpůsobeny i pro bezdrátovou komunikaci, například pro aplikaci na konstelační symboly.

Variační autoenkodér je založený na hlubokých neuronových sítích a kombinuje principy autoenkodérů a pravděpodobnostního modelování. První částí, ze které se skládá, je enkodér. Jeho cílem je mapovat vstupní data do latentního prostoru reprezentovaného pravděpodobnostní distribucí. Druhou částí je dekodér, který z této latentní reprezentace rekonstruuje vstupní data. Tato architektura umožňuje generování nových dat, jejichž statistické vlastnosti odpovídají vlastnostem tréninkové sady.

Generativní adversariální síť je založena na soupeření 2 neuronových sítí – generátoru a diskriminátoru. Zatímco generátor vytváří syntetické vzorky dat, diskriminátor provádí klasifikaci všech vzorků s cílem rozlišit reálné vzorky od vygenerovaných. Obě sítě jsou trénovány současně a cílem generátoru je vygenerovat taková data, která budou s co největší pravděpodobností diskriminátorem označena

jako reálná. GAN je velmi široce využívaná díky své schopnosti generovat detailní a vizuálně realistické vzorky dat.

Hlavním rozdílem mezi těmito 2 přístupy je přístup k modelování datové distribuce. Zatímco VAE se zaměřuje na explicitní modelování pravděpodobnostního rozdělení v latentním prostoru, GAN řeší problém prostřednictvím soupeření sítí bez explicitního odhadu distribuce. Díky tomu bývají VAE stabilnější během tréninku, zatímco během tréninku GAN může docházet k problémům s vyvážením mezi oběma sítěmi, což činí trénink nestabilním [26], [27], [28].

Učení doménově invariantních reprezentací

Tato technika se zaměřuje na extrakci reprezentativních rysů dat pro zlepšení schopnosti modelů generalizovat. Cílem následujících 3 metod je naučit se rysy, které jsou invariantní vůči různým doménám:

1. Učení invariantních reprezentací zahrnuje použití kernelových metod, jako je například SVM.
2. Během adversariálního učení je 1 nebo více generativních modelů trénováno tak, aby postupně generovaly neznámé domény učení se relevantních reprezentací invariantních napříč doménami.
3. Disentanglování rysů umožňuje rozdělení rysů na doménově invariantní a doménově specifické rysy. Tato technika tak umožňuje explicitně rozlišit rysy na ty, které jsou společné pro všechny domény, a na ty, které jsou naopak pro jednotlivé domény specifické.

Učební paradigmatata

Učební paradigmatata představují různé metodologické přístupy k trénování modelů strojového učení s cílem dosáhnout robustní a generalizované výkonnosti na neznámých datech a doménách. Jedná se o strategie, které umožňují modelům adaptaci na různé distribuční posuny mezi trénovacími a testovacími daty bez nutnosti do-datečného doladování.

Meta učení, známé také jako učení učit se, se zaměřuje na trénování modelů s cílem extrahovat znalosti, které mohou být použity k rychlé adaptaci na nové úlohy. Jednou z nejznámějších technik je MAML (Model Agnostic Meta Learning), která optimalizuje modely pro rychlou adaptaci na nové úlohy pomocí několika gradientních kroků.

Technika **samoučení** využívá neoznačená data k posílení schopnosti modelu generalizovat. Model je nejprve trénován na malém množství označených dat. Predikce modelu na neoznačených datech jsou považovány za pravdivé a následně jsou použity

k dalšímu trénování modelu. Technika je velmi užitečná v oblastech, kde se hojně vyskytují neoznačená data, zejména pokud je proces označování obtížný a nákladný.

Skupinové učení se zaměřuje na kombinaci více modelů za účelem dosažení robustnějších a přesnějších predikcí. Tato technika využívá různé přístupy k agregaci výsledků z různých modelů. Kombinací výstupů z několika modelů se zvyšuje přesnost a odolnost vůči variabilitě dat.

Generalizace tedy předpokládá, že nejsou k dispozici data pocházející z nového neznámého prostředí. V případě, že je k dispozici alespoň nějaké množství dat z cílové domény, lze využít další přístupy k učení modelu popsané v následujícím textu. Tyto přístupy však nemusí splňovat jednu ze zásad generalizace, a to, že jsou modely učené bez nutnosti dalšího ladění.

Přenesené učení

Přenesené učení (Transfer Learning) se zaměřuje na přenos získaných znalostí z jedné domény do druhé, k čemuž využívá předtrénovaný model. Tento přístup může výrazně snížit požadavky na čas a data, neboť k doladování modelu využívá pouze malé množství dat v porovnání s původní tréninkovou sadou. Přenositelnost modelu může být limitována výraznou odlišností domén.

Adaptace domény

Adaptace domény (DA – Domain Adaptation) předpokládá existenci cílových vzorků v době tréninku, které využívá pro zlepšení výkonu modelu v cílových doménách. Výchozí přístup adaptace domén předpokládá neoznačené cílové vzorky a sdílený prostor značek mezi cílovou a zdrojovou doménou (homogenní adaptace domén).

Učení s nulovým počtem snímků

Učení ZL (Zero-shot Learning) se zabývá především posunem domény pro štítky. Jeho cílem je rozpoznat třídy nebo cílové hodnoty, které nejsou k dispozici během tréninku.

Adaptace/trénink v době testu

Technika adaptace tréninku v době testu (Test-time Adaptation/Training) umožňuje natrénovanému modelu adaptaci na nové vstupy v reálném čase, aniž by bylo nutné přistupovat ke zdrojovým datům a značkám. Model je trénován k provádění 2 úloh (hlavní a pomocné). Pomocná úloha je využita v době testování pro vytváření štítků pro neoznačená testovací data, díky čemuž může být pro doladění modelu využito pouze velmi malé množství vzorků.

Průběžné učení

Průběžné, nebo také celoživotní učení (Continual/Lifelong learning) umožňuje modelu učit se nové úlohy nebo se adaptovat na nové domény bez zapomenutí již nabytých znalostí. Tento přístup předpokládá, že model nemá přístup k datům předchozích úloh a aktualizuje parametry pomocí označených dat z nových úloh nebo domén [29], [30], [31].

3 Stávající řešení

Tato kapitola se věnuje zejména evaluaci metod pro klasifikaci přímého a odraženého signálu. Pro účely ověření je v části 3.1 zpracován přehled veřejně dostupných sad vhodných pro účely klasifikace LoS a NLoS signálu, kapitola 3.1.1 popisuje získaná neveřejná data získaná měřením na kanálu 9. Následuje přehled a popis publikovaných metod klasifikace – ověření vybraných metod v definovaných scénářích v části 3.2, představení metod, které z různých důvodů nebylo možné detailně ověřit v kapitole 3.3, a závěrem je porovnání všech zmíněných metod v kapitole 3.4.

Metody pro klasifikaci byly pro účely ověřování v rámci této práce implementovány s využitím knihoven Tensorflow 2.10.1 [32] a Scikit-learn 1.5.2 [33] v programovacím jazyce Python ve verzi 3.9.20. Pokud dokumentace publikovaného řešení obsahuje dostatečné množství informací, jsou pro dané řešení určeny parametry, s jejichž pomocí lze odvodit paměťovou a výpočetní náročnost daného řešení. V opačném případě je v závislosti na dostupných informacích o daném řešení paměťová a výpočetní závislost pouze teoreticky diskutována. Parametry, s jejichž pomocí je v rámci této práce odvozena náročnost jednotlivých řešení pro účely jejich vzájemného porovnání, z nichž některé byly převzaty z [34], [35] a [36], jsou:

- Počet parametrů modelu – počet trénovatelných parametrů modelu. Ukazatel paměťové náročnosti a složitosti modelu. Počet trénovatelných parametrů modelu přímo ovlivňuje jeho paměťovou náročnost, a zároveň kvantifikuje schopnost modelu zachytit určité komplexní vzory. Větší počet parametrů zvyšuje riziko overfittingu.
- Velikost modelu – skutečná velikost modelu v paměti. Parametr slouží k přesné kvantifikaci reálných nároků na paměť.
- FLOPs (Floating Point Operations) – počet operací s plovoucí řádovou čárkou. Představuje teoretickou míru výpočetní náročnosti. Může být použit k teoretickému srovnání efektivity různých algoritmů, odhadu energetické náročnosti, odhadu rychlosti inference, návrhu a optimalizaci modelu. Zároveň ale nezhledňuje rozdíly ve složitosti operací a ignoruje hardwarovou optimalizaci některých operací.
- Špičková paměťová náročnost – maximální hodnota velikosti alokované operační paměti během inference 1 vzorku. Parametr je kritický v kontextu posouzení nasaditelnosti modelu na zařízení s omezeným množstvím operační paměti. Měření je end-to-end a začíná načtením modelu do paměti, následuje načtení (vytvoření) vstupních dat a samotná inference, kterou měření končí. Parametr je pro účely případného budoucího porovnávání získán s využitím 3 různých metod (knihoven):

1. Psutil 7.0.0 – umožňuje měření paměťové náročnosti procesu [37]. Vý-

sledky s využitím této metody jsou zároveň prezentovány v tabulce 3.10, která porovnává všechna popsaná řešení.

2. Tensorflow – umožňuje měření VRAM paměti alokované prostřednictvím Tensorflow. Pro měření tímto způsobem je nezbytné povolit běh modelu (programu) na GPU.
 3. Tracemalloc – tato nativní knihovna umožňuje měření paměti alokované pythonem [38].
- Průměrný čas klasifikace 1 vzorku – klíčový parametr pro aplikace běžící v reálném čase, který poskytuje pohled na rychlost (výpočetní náročnost) modelu v reálných podmínkách. Do času klasifikace není zahrnut čas nezbytný pro předzpracování dat (transformace, normalizace, výpočty, ...), neboť z důvodu chybějících vstupních dat nelze ve všech případech předzpracování dat věrně replikovat. Klasifikace je prováděna s náhodným vzorkem, který reálnému vzorku odpovídá pouze svým rozměrem. Čas klasifikace je závislý na parametrech fyzického zařízení, na kterém je model nasazen.
 - Optimální velikost dávky – velikost dávky, při které model vykazuje nejrychlejší čas klasifikace. Parametr umožňuje porovnat efektivitu paralelizace modelů, kdy některé z nich mohou být lépe optimalizované pro dávkové zpracování dat, a jeho hodnota je přímo závislá na použitém fyzickém zařízení. Experimentováno je s dávkami o velikosti 2^N , pro $N = \langle 0; 11 \rangle$.
 - Propustnost – počet klasifikovaných vzorků za 1 s při optimální velikosti dávky. Parametr je důležitou metrikou pro aplikace vyžadující zpracování velkého množství dat. Součástí popisu jednotlivých řešení, v rámci kterých bylo možné získat tyto parametry, jsou pro účely vzájemného porovnání uvedeny i hodnoty propustnosti s některými nižšími velikostmi dávek. Při prezentaci hodnot bude pro označení jednotky použito „S/s“ (Samples per second) vyjadřující právě počet vzorků za sekundu.

Hodnoty parametrů byly získány s využitím stolního počítače s procesorem Intel Core i5-7500 se 4 jádry a frekvencí 3,4 GHz a 24 GB operační paměti typu DDR4. Všechny hodnoty parametrů s výjimkou alokace VRAM byly měřeny pouze s využitím CPU, využití grafické karty bylo pro proces softwarově zablokováno.

3.1 Dostupné datové sady

Pro účely ověřování dostupných řešení pro klasifikaci přímého a odraženého UWB signálu byl sestaven přehled veřejně dostupných datasetů vhodných pro účely klasifikace přímého a odraženého UWB signálu. Vhodný dataset pro účely klasifikace musí primárně obsahovat označení (klasifikaci), zda daný vzorek náleží scénáři LoS či NLoS, případně musí obsahovat klíč k takovému označení. Přehled těchto datasetů

je prezentován v tabulce 3.1 a neslouží jako jejich celková charakteristika. Obsahuje pouze parametry relevantní pro účely ověřování stávajících řešení v kontextu této práce, a hodnoty těchto parametrů mohou být ze stejného důvodu rovněž upraveny. V následujícím textu budou podrobně představeny veřejně dostupné datasety a na konci této kapitoly bude představen získaný neveřejný dataset, který je také součástí zmíněné tabulky a je uveden pod číslem 7.

1. Dataset [39] čítá celkem 42000 vzorků získaných měřeními na kanálu 2 s využitím modulu DWM 1000, který dosahuje přesnosti při měření vzdálenosti uvnitř budov až 10 cm. Měření probíhalo v různých vnitřních prostředích, které dělí celý dataset na 7 podmnožin – 2 kancelářské prostory, malá dílna, 2 bytové prostory, kuchyně s jídelnou a kotelna. Každé prostředí má své specifické charakteristiky vícecestného šíření, čímž se autoři publikace [34], kteří jsou rovněž autoři datasetu, snaží zamezit jevu, při kterém je model pro klasifikaci příliš přizpůsoben jednomu prostředí. Každý vzorek datasetu obsahuje odhad vzdálenosti, sílu signálu, úroveň šumu a impulsní odezvu kanálu, která se v závislosti na PRF skládá z 992 (PRF = 16 MHz) či 1016 (PRF = 64 MHz) vzorků – v tabulce 3.1 označeno jako CIR WL (Window Length – délka/velikost okna). Analýzou datasetu bylo zjištěno, že dataset obsahuje 1016 sloupců pro CIR, kdy pro kratší CIR jsou poslední sloupce vyplněny nulami. Dataset je vyvážený a obsahuje stejné množství vzorků pro oba typy signálu [34], [39], [40].
2. Dataset [41] od stejných autorů s celkem 35000 vzorky byl stejně jako první dataset byl získán s využitím modulu DWM 1000. Dataset vznikl pro účely publikace [34], konkrétně pro část, která se zabývá využitím konvolučních neuronových sítí pro odhad chyby měření vzdálenosti, na což je také tento dataset orientován. Měření probíhalo ve 2 prostředích – ve 2 sousedících kancelářských místnostech (kanál 2) se spojovací chodbou a v kancelářském prostředí s více místnostmi (kanály 1, 3, 4, 5 a 7). Každý vzorek obsahuje spoustu informací, které jsou využity pro odhad chyby měření, a jejich výčet lze najít u vlastního popisu datasetu v rámci publikace. Mimo tyto informace obsahuje i 1016 CIR vzorků a index detekovaného prvního výskytu přijatého signálu v rámci vzorků CIR [34], [40], [41]. Analýzou datasetu byly o výskytu kanálů napříč celým datasetem zjištěny následující informace:
 - Kanál 2 – 17000 vzorků, z toho 11900 (70 %) NLoS.
 - Kanály 1, 3, 4, 5, 7 – pro každý kanál shodně 3600 vzorků, z toho vždy 3000 (83,33 %) NLoS.
 - Celkový počet NLoS vzorků v datasetu je 26900, což odpovídá 76,86 %. Dataset lze považovat za nevyvážený, a bez aplikace metod zmíněných v 2.1.1 nevhodný k tréninku modelu.

3. Dataset [42] obsahuje 36700 vzorků naměřených s využitím modulu DWM 1001 na kanálu 5. Měření probíhalo celkem v 11 různých prostředích – 5 různých laboratořích, 2 chodby, vzdělávací centrum a různé kancelářské prostory. Dataset neobsahuje přímo absolutní hodnotu CIR, ale její reálnou a imaginární část o délce 1016 vzorků. Autoři v popisu datasetu uvádějí ukázkový zdrojový kód v jazyce python, pomocí kterého lze absolutní hodnotu vypočítat, a který zároveň dokáže zarovnat první výskyt signálu na pevný index, aby byly jednotlivé naměřené vzorky jednodušeji porovnatelné. Autoři publikace [43], v rámci které vznikl tento dataset, uvádějí, že bylo naměřeno 35050 vzorků. Vlastní analýzou datasetu bylo však zjištěno, že skutečný počet je mírně vyšší. Měření probíhalo ve 3 scénářích:

- LoS – mezi oběma zařízeními nebyly žádné překážky.
- NLoS – mezi zařízeními se nacházely překážky, které významně ovlivňují šíření signálu (např. tlusté stěny).
- WLoS (Weak LoS) – mezi zařízeními se nacházely překážky, které šíření signálu neovlivňují tak výrazně (např. lidé, monitory, židle, . . .) [40], [42], [43].

Protože se tato práce zabývá problematikou detekce přímého a odraženého signálu, je nezbytné WLoS vzorky pro účely této práce z datasetu vyloučit (stejně jako např. [34] a [40]). Po této úpravě má dataset 24892 vzorků, ze kterých 12348 odpovídá NLoS (49,61 %). Dataset je tedy vyvážený a vhodný pro případný trénink modelu.

4. Vzorky datasetu [44] od tvůrců publikace [45] byly získány za využití 3 různých zařízení. Jednalo se o Decawave DWM 3000, 3db Access 3DB6830C a TDSR P452A. První 2 zmíněná zařízení jsou v souladu s některou z verzí standardu IEEE 802.15.4, zatímco TDSR P452A používá proprietární fyzické rozhraní. Měřicí scénáře probíhaly vždy totožně a docházelo pouze k výměně UWB zařízení. Z důvodů softwarové limitace u DWM 3000 obsahuje dataset výrazně méně vzorků s využitím tohoto typu zařízení. Měření bylo provedeno v 9 různých prostorech univerzity v Tampere s následujícím nastavením UWB modulů:

- DWM 3000 – Frekvence 6,5 GHz a šířka pásma 600 MHz. Nastavení dle [1] odpovídá kanálu 7 (střední frekvence 6489,6 MHz a šířka pásma maximálně 1081,6 MHz). V oficiální dokumentaci výrobce modulu [46] je však uvedeno, že modul DWM 3000 podporuje kanály 5 a 9. Z kontextu publikace [45] není jasné, jakým způsobem byla zjištěna použitá šířka pásma, a pro účely této práce bude jako referenční hodnota pro určení kanálu použita střední frekvence, neboť pomocí níž lze v kombinaci s dokumentací UWB modulu určit použitý kanál, a to konkrétně jako kanál 5. S vyu-

žitím tohoto modulu bylo získáno celkem 11206 vzorků, z nichž je 5604 klasifikováno jako NLoS (45,50 %).

- 3db Access 3DB6830C – Frekvence 6,5 GHz a šířka pásma 380 MHz. Nastavení dle [1] odpovídá kanálu 5 (střední frekvence 6489,6 MHz a šířka pásma maximálně 499,2 MHz). Jako UWB je označována technologie, která dle definice uvedené v úvodu kapitoly 1 používá šířku pásma o velikosti větší nebo blíží se 500 MHz, případně dle [47] je jako UWB technologie označována i technologie, která používá šířku pásma o velikosti odpovídající alespoň 20 % své střední frekvence. Z dokumentace výrobce dostupné z [48] vyplývá pouze informace, že zařízení podporuje frekvence od 6 do 8 GHz. Uvedená informace o použité šířce pásma je tedy zavádějící stejně, jako informace o použité šířce pásma u zařízení DWM 3000, a z toho důvodu je nezbytné konstatovat, že není jednoznačně možné určit použitý kanál. Vzhledem k tomu, že ale neexistují informace, které by vyvracely údaje o použité střední frekvenci, která byla u předchozího zařízení pravděpodobně uvedena správně, bude pro účely této práce použitý kanál klasifikován jako kanál 5. Dataset získaný měřením s využitím tohoto modulu čítá 56805 vzorků, z nichž 25849 (45,50 %) bylo klasifikováno jako NLoS.
- TDSR P452A – Frekvence 4,3 GHz a šířka pásma 620 MHz. Protože zařízení není v souladu s IEEE 802.15.4(z), nelze v kontextu této práce přiřadit této části datasetu UWB kanál a pro účely této práce nebudou uvažována data získaná s využitím tohoto zařízení. Po frekvenční stránce je použité nastavení nejbližší kanálu 4 (střední frekvence 3993,6 MHz, maximální šířka pásma 1331,2 MHz) dle IEEE 802.15.4(z) [44], [45].

S využitím zařízení DWM 3000 a 3db Access 3DB6830C bylo tedy získáno celkem 68011 vzorků, z nichž bylo 31453 (46,25 %) klasifikováno jako NLoS.

5. Dataset [49] obsahuje 28473 neoznačených vzorků získaných měřením s využitím modulu DWM 1000 na kanálu 1. Měření probíhalo na 21 místech uvnitř průmyslového skladu. Počet vzorků pro CIR je 1016. Pro větší část vzorků je součástí i výčet kotev, které byly v přímé viditelnosti, díky čemuž lze tuto část vzorků označit. Označený dataset čítá 20360 vzorků, v rámci nichž 13821 (67,88 %) náleží NLoS scénáři a lze tedy konstatovat, že se jedná o nevyvážený dataset [49], [50].
6. Dataset [51] byl naměřený pro účely publikace [52], která se zabývá detekcí NLoS signálu, díky které lze zmírnit chyby odhadu vzdálenosti, při měření uvnitř budov s použitím jednoho z finančně méně náročnějších UWB zařízení. K měření byl využit modul od firmy *Pozyx*, který je vystavěn nad známým čipem DWM 1000 od společnosti *Decawave*. Ačkoli v [52] a ani v samotném

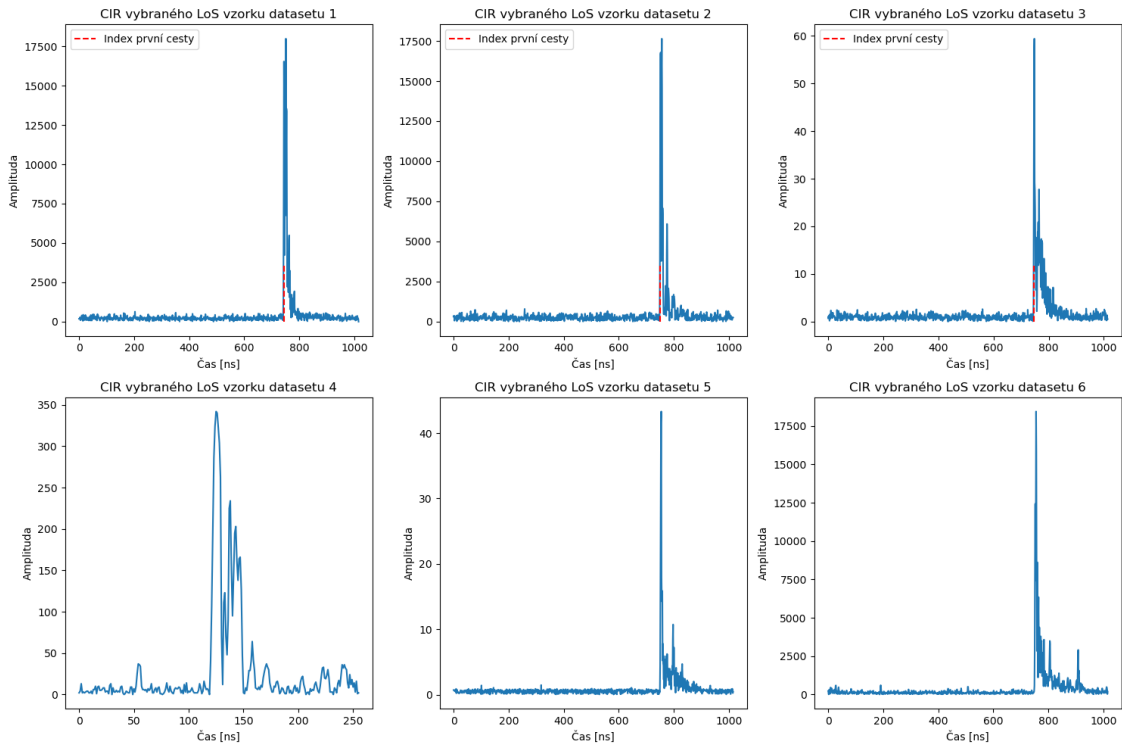
Dataset	Kanál	Zařízení	Prostor měření	CIR WL	Počet vzorků	NLoS vzorků [%]
1 [39]	2	DWM 1000	Kancelářské a bytové prostory, malá dílna, kuchyň s jídelnou, kotelna	1016	42000	50,00
2 [41]	1, 2, 3, 4, 5, 7	DWM 1000	Kancelářské prostory	1016	35000	76,86
3 [42]	5	DWM 1001	Laboratoře, chodby, vzdělávací centrum, kancelářské prostory	1015	24892	49,61
4 [44]	5	DWM 3000, 3db Access 3DB6830C	Různé prostory univerzity Tampere	1016, 256	68011	46,25
5 [49]	1	DWM 1000	Průmyslový sklad	1016	20360	67,88
6 [51]	5	Pozyx	Prostor mezi chodbami	1016	3069	0
7	9	Qorvo QM33	Průmyslový sklad	1016	7588	70,31

Tab. 3.1: Přehled dostupných datasetů použitelných pro účely klasifikace LoS/NLoS signálu.

datasetu není uvedena informace vedoucí k identifikaci použitého kanálu, z informací dostupných na webových stránkách výrobce [53] lze předpokládat, že zařízení má pevně přednastavený kanál na kanál číslo 5. Měření bylo provedeno v oblasti uprostřed 2 chodeb s přilehlou místností, a bylo získáno celkem 4796 vzorků. Autoři však uvažovali i scénář NLoS soft, který odpovídá scénáři wLoS u datasetu 3, avšak na rozdíl od uvedeného datasetu nejsou vzorky pro tento scénář nijak odlišeny a dataset obsahuje pouze označení „LoS“ a „NLoS“, přičemž pod druhé z nich pravděpodobně spadá i NLoS soft scénář. Toto tvrzení potvrzuje fakt, že autoři publikace vizualizací dat přišli na chybu u kotev pro NLoS a NLoS soft scénář, kdy tyto kotvy chybně identifikovali vzdálenost při skutečné vzdálenosti mezi zařízeními přes 9 metrů. Z toho důvodu vyřadili z této části datasetu vzorky pro vzdálenost přesahující 9,5 metru. Vlastní analýzou datasetu bylo zjištěno, že NLoS část datasetu obsahuje pouze vzorky do vzdálenosti 9,1 metru, zatímco LoS část do vzdálenosti 16 metrů. Toto zjištění zvyšuje pravděpodobnost, že je předchozí hypotéza pravdivá, a že NLoS soft vzorky jsou zahrnuty v NLoS části datasetu. Z důvodu uvedeného u popisu datasetu 3 budou pro účely této práce z datasetu vyřazeny vzorky NLoS soft, což v konečném důsledku znamená vyřazení všech NLoS vzorků. Po této úpravě čítá dataset celkem 3069 (LoS) vzorků [51], [52].

Obrázek 3.1 znázorňuje průběhy impulsní odezvy kanálu náhodně vybraného LOS vzorku napříč každým z popisovaných veřejně dostupných datasetů. Pro účely klasifikace s využitím CIR je žádoucí mít přehled o rozložení CIR v čase, jelikož se

může pro jednotlivé datasety lišit a některá řešení, jako např. [34] pracují například pouze s prvními 150 CIR vzorky. Pokud je součástí datasetu i hodnota představující index výskytu první cesty, je tato hodnota rovněž znázorněna na obrázku. Přestože dataset 5 tuto hodnotu obsahuje, tak není na obrázku zobrazena, neboť leží výrazně mimo očekávaný interval hodnot.



Obr. 3.1: Časové průběhy impulsní odezvy kanálu náhodně vybraného LoS vzorku v každém datasetu.

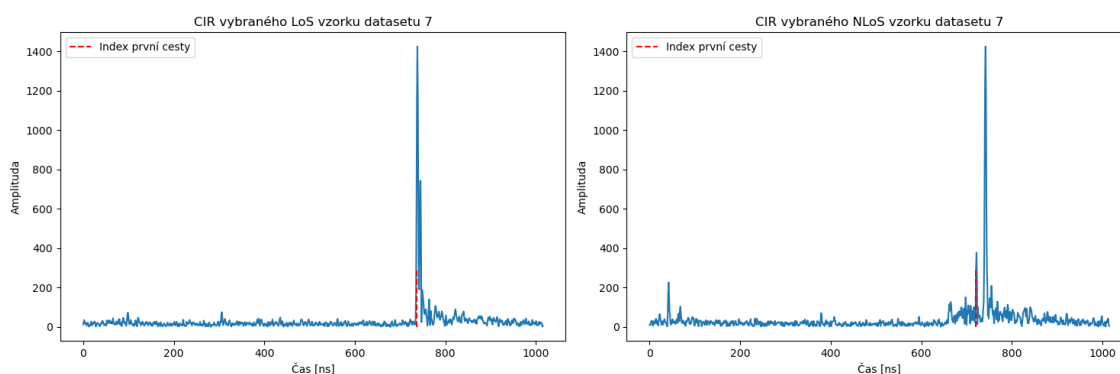
Tabulka 3.2 rozděluje popsané veřejně dostupné datasety, případně jejich části, podle UWB kanálu, na kterém probíhalo měření. Z tabulky je patrné, že dostupné datasety vhodné pro klasifikaci LoS/NLoS signálu byly primárně měřeny na kanálech 5, 2 a 1. To je dáno především typem použitých zařízení, kdy většina datasetů byla měřena převážně s využitím zařízení DWM 1000, které podporuje kanály 1, 2, 3, 4, 5 a 7. Publikace, pro jejichž účely byly datasety pořízeny, většinou nijak neodůvodňují výběr konkrétního kanálu a pokud ano, tak volí kanál s nejvyšším výkonem. Pro kanál 9 nebyl nalezen žádný veřejně dostupný dataset. V souvislosti s daty pro kanál 9 byl navázán kontakt s jedním z autorů publikace [40] s žádostí o poskytnutí datasetu, který pro účely své práce naměřili. Dataset, který naměřili s využitím kanálu 9, obsahuje téměř 96 tisíc vzorků z průmyslového prostředí. Do okamžiku publikování této práce nebylo na tuto žádost odpovězeno.

Kanál	Počet vzorků	CIR WL	NLoS vzorků [%]
1	23960	1016	70,20
2	59000	1016	55,76
3	3600	1016	83,33
4	3600	1016	83,33
5	99572	256, 1016	47,00
7	3600	1016	83,33

Tab. 3.2: Rozložení vzorků z veřejně dostupných datasetů v kontextu použitého UWB kanálu.

3.1.1 Neveřejná datová sada

Datovou sadu obsahující vzorky naměřené na UWB kanálu 9 se podařilo získat od firmy *Sewio Networks*, která se zabývá vývojem a výrobou systému pro lokalizaci v reálném čase uvnitř budov (RTLS – Real-Time Location System). Dataset pochází z reálného průmyslového skladu a k jeho měření byl použit modul Qorvo QM33. Měření probíhalo dynamicky – člověk s UWB tagem umístěným na helmě a hrudi se kontinuálně pohyboval po průmyslovém skladu. Dataset je anonymizovaný a obsahuje pouze informace o indexu první cesty, výkonu první cesty a impulsní odezvu kanálu o délce 1016 vzorků. Pro účely této práce byl firmou poskytnut i klíč k označování dat. Označovaný dataset čítá 7588 vzorků, v rámci nichž se ve 5335 (70,31 %) případech jedná o NLoS scénář. V rámci tabulky 3.1 s přehledem dostupných datasetů použitelných pro účely klasifikace LoS/NLoS je tento dataset uveden pod číslem 7.



Obr. 3.2: Časové průběhy impulsní odezvy kanálu náhodně vybraného LoS a NLoS vzorku v datasetu 3.1.1.

Obrázek 3.2 znázorňuje průběhy impulsní odezvy kanálu náhodně vybraného

LoS a NLoS vzorku tohoto datasetu analogicky jako u popisovaných veřejně dostupných datasetů. Na obrázku je rovněž vyznačena hodnota indexu první cesty, jejíž hodnota odpovídá očekávanému intervalu a která svou pozicí odpovídá hodnotám u ostatních popisovaných datasetů, jež tento index obsahují a obsahují stejný počet vzorků pro CIR.

3.2 Ověřování publikovaných metod

Tato část práce se zaměřuje pouze na metody, které jako vstupní data používají impulsní odezvu kanálu nebo z ní extrahované rysy. Z širokého souboru publikovaných metod je pouze několik z nich vhodných k ověření v rámci této práce, neboť většina řešení postrádá potřebné informace. Mezi takové informace mohou patřit podrobnější struktura navrženého řešení, chybějící parametry použitých algoritmů nebo cokoliv jiného, co znemožňuje ověření modelu. Další limitací při ověřování dostupných řešení představují dostupné datasety popsané v kapitole 3.1, jejichž vzorky nemusí obsahovat parametry nezbytné k ověření řešení.

Struktura této části práce je následující:

- V části 3.2.1 jsou definovány scénáře, pomocí kterých budou publikované metody individuálně ověřovány.
- Detailní popisy struktury navržených modelů a dosažených výsledků individuálního ověřování jsou uvedeny v částech 3.2.2– 3.2.7.
- Část 3.2.8 obsahuje srovnání ověřovaných řešení z této části práce.

Publikované metody budou ověřeny s využitím dostupných datových sad popsaných v kapitole 3.1. Pro účely ověřování publikovaných řešení proběhla také unifikace dostupných veřejných datasetů, která spočívala zejména ve sjednocení formátu dat. Bylo sjednoceno označení jednotlivých parametrů a ke každému vzorku byla přidána informace o datasetu, ze kterého vzorek pochází, kanálu, na kterém byl naměřen, a zařízení, pomocí kterého byl naměřen. Sjednoceno bylo také označení jednotlivých vzorků – „0“ pro LoS a „1“ pro NLoS vzorky. V kontextu hodnocení výkonnosti modelů popsaného v kapitole 2.2.4 NLoS vzorky tedy odpovídají pozitivním vzorkům. Uvedené hodnoty preciznosti lze interpretovat jako množství LoS kvalifikovaných jako NLoS (falešně pozitivní chyby) a hodnoty citlivosti jako množství správně identifikovaných NLoS vzorků ze souboru všech skutečných NLoS vzorků. Při nízkých hodnotách citlivosti dělá model vysoké množství falešně negativních chyb (NLoS identifikuje jako LoS).

V rámci tabulek představujících výsledky ověřování či případně textů popisujících způsob ověřování jednotlivých řešení jsou použity následující zkratky:

- $D\{X\} - \{X\}$ vyjadřuje číslo datasetu dle tabulky 3.1, např. D1 je označení pro dataset 1.
- D_{ALL} – představuje dataset vzniklý sloučením všech datasetů. Dataset obsahuje 200 921 vzorků, z nichž 55,17% odpovídá NLoS scénáři.
- D_O – představuje ověřovaný dataset. Pokud je jako ověřovaný dataset uveden např. D1, tak D_O představuje právě D1.
- $D_{ALL} - D_O$ – představuje dataset vzniklý sloučením všech datasetů vyjma datasetu, vůči kterému je řešení ověřováno.

3.2.1 Scénáře pro ověřování

Pro možnosti ověření a porovnání publikovaných řešení byly jednotlivé přístupy ověřovány prostřednictvím 4 scénářů. Při ověřování byly následovány parametry tréninku i rozdělení tréninkového datasetu uvedené v publikaci pro konkrétní řešení. U každého ověřovaného řešení je zároveň uvedeno, které parametry uvedeny nebyly a jakým způsobem byly determinovány. Následující text obsahuje popis ověřovacích scénářů.

1. Trénování na původním datasetu. Cílem scénáře je ověřit úspěšnou implementaci publikovaného řešení a publikované výsledky, kterých bylo s jeho pomocí dosaženo. V případě, že je původní dataset neveřejný, bude použit jiný veřejný dataset, jenž by mohl nejlépe odpovídat původnímu datasetu. Pokud v publikaci pro konkrétní ověřovanou metodu popis datasetu, na kterém byl model trénován, chybí, bude řešení trénováno na datasetu 1 (D1), neboť se v ověřovaných řešeních objevuje nejčastěji.
2. Trénování na datasetu z předchozího scénáře a testování na všech zbylých datasetech. Tento scénář představuje situaci, kdy je dostupný model natrénovaný na jiném než vlastním datasetu. Cílem je zjistit míru schopností modelu klasifikovat data z jiného prostředí, zařízení a kanálu.
3. Trénování a testování na stejném datasetu. Tento scénář představuje situaci, kdy je dostupná architektura modelu bez vah, případně s váhami a je z libovolného důvodu požadavek či možnost model trénovat na vlastních datech. Díky tomuto scénáři je možné porovnat výkonnost modelu při tréninku na různých datech. Do scénáře není zahrnut dataset D6 z důvodu, že obsahuje pouze malé množství vzorků, které jsou navíc všechny pro scénář LoS. U dat určených pro trénink (a validaci během učení) je navíc zajištěna vyváženost datové sady.
4. Trénování na datasetu vzniklém sloučením všech datasetů mimo dataset, vůči kterému je publikované řešení testováno (označení $D_{ALL} - D_O$). Scénář je

testován pro všechny datasety a představuje situaci, kdy je dostupný model natrénovaný na větším množství dostupných datasetů. U dat určených pro trénink (a validaci během učení) je navíc zajištěna vyváženost datové sady.

3.2.2 CNN LSTM

Řešení publikované v [54] představuje architekturu hlubokého učení, která používá konvoluční neuronovou síť k extrakci rysů z impulzní odezvy kanálu, jež jsou následně předány jako vstup LSTM, která provádí konečnou klasifikaci. Autoři publikace nejprve experimentovali s množstvím konvolučních vrstev. Byly provedeny testy s 1 až 4 konvolučními vrstvami a pro výsledný model byla zvolena varianta se 2 vrstvami, protože oproti ostatním vykazovala nejvyšší úspěšnost. To mohlo být dle autorů způsobeno faktem, že model s více než 2 konvolučními vrstvami mohl v tomto případě vykazovat overfitting a hlubší struktura neuronové sítě mohla vést k nižší efektivitě učení. Publikovaná přesnost u architektury se 2 konvolučními vrstvami je 81,56 %.

Další experimenty byly zaměřeny na LSTM část architektury. Ve 3 případech byla v kombinaci s LSTM použita architektura se 2 konvolučními vrstvami popsaná v předchozím odstavci. Byly porovnávány 4 různé modely, které byly trénovány na stejných datech, jako konvoluční část modelu:

- LSTM – přesnost 51,11 %.
- CNN + LSTM – přesnost stejná, jako u řešení obsahující pouze CNN, a to tedy 81,56 %.
- CNN + Obousměrná (Bi-directional) LSTM – přesnost 78,93 %.
- CNN + Vícevrstvá (Stacked) LSTM – oproti ostatním architektuám dosahuje mírně vyšší přesnosti, konkrétně 82,14 %.

Navržený model byl trénován na datech z datasetu 1. Data tohoto datasetu byla naměřena v 7 různých prostředích, v každém prostředí bylo získáno shodné množství vzorků pro oba scénáře, celkem 6000 vzorků. Pro účely tréninku bylo náhodně vybráno 5000 vzorků z každého prostředí, z nichž bylo 25000 použito pro učení a 10000 pro testování. Vstupní vektor konvoluční vrstvy má rozměr 1x1016 a představuje impulsní odezvu kanálu.

Struktura publikované architektury je následující:

1. Konvoluční vrstva s 10 filtry, velikostí jádra 4 a aktivační funkcí ReLu. Nevyužívá padding (pro zachování rozměrů vstupních dat).
2. Konvoluční vrstva s 20 filtry, velikostí jádra 5 a aktivační funkcí ReLu. Nevyužívá padding.

3. Maxpooling¹ s velikostí okna 2 a krokem 2.
4. Dropout² 0.5.
5. LSTM se 32 skrytými jednotkami.
6. LSTM se 32 skrytými jednotkami.
7. Dropout 0.5.
8. Plně propojená vrstva se 128 neurony a aktivační funkcí ReLu.
9. Plně propojená (výstupní) vrstva s 1 neuronem a sigmoid jako aktivační funkcí.

Architektura byla trénována s rychlostí učení 0,001 v celkem 10 epochách³ a použitá velikost dávky⁴ (batch size) byla 64. Zvolená optimalizační funkce⁵ není v publikaci uvedena, pro účely ověření architektury byl zvolen optimalizátor Adam⁶ (Adaptive Moment Estimation).

Výsledky ověřování publikovaného řešení jsou prezentovány v tabulce 3.3 a byly získány s využitím scénářů popsaných v předchozí kapitole. Autory prezentovaná přesnost řešení při tréninku na D1 je 82,14%. Z publikace vyplývá, že bylo $\frac{5}{6}$ dat použito pro trénink (5000 vzorků z každé místnosti) a $\frac{1}{6}$ dat pro testování. Není ale uvedeno, zda a případně jaká část z trénovacích dat byla použita jako validační. Byly provedeny experimenty s hodnotou velikosti validační části datasetu mezi 0 a 20% při tréninku a ověřování na datech datasetu D1, během kterých zaznamenával model přibližně konstantní přesnost klasifikace v okolí 83%. Protože předmětem této části práce není vylepšování výsledků publikovaného řešení, a zároveň jsou veřejně dostupná data, na kterých byl model trénován, byla pro ověřování tohoto řešení stanovena hodnota velikosti trénovací části dat na $\frac{4}{6}$, validační části na $\frac{1}{6}$ a hodnota velikosti testovací části dat na $\frac{1}{6}$. Tento způsob rozdělení datasetu byl zvolen s ohledem na běžný přístup, kdy se volí shodné velikosti testovací a validační sady, a zároveň je tento způsob použit například i autory ověřovaného řešení 3.2.5. Tento přístup zároveň snižuje možnost overfittingu dat, který by mohl být zapříčiněn významným nepoměrem mezi velikostmi trénovací a testovací sady, a mohl by tak negativně ovlivnit 2. a 4. scénář. S tímto rozdělením datasetu bylo u prvního scénáře dosaženo přesnosti 83,49%, která je v porovnání s autory prezentovanou přesností dokonce mírně vyšší.

¹Slouží ke snížení rozměrů dat a extrakci nejdůležitějších rysů. Z každého okna vstupních dat vybere maximální hodnotu, kterou předá další vrstvě. Parametr krok (stride) určuje, o kolik se okno posune [55].

²Výsledkem dropoutu je, že je část neuronů ignorována a vyřazena z tréninku, díky čemuž lze snížit riziko overfittingu dat [54].

³Tréninková epocha představuje kompletní průchod celým souborem trénovacích dat a aktualizací vah na základě chyb během 1 tréninkového cyklu [56].

⁴Označuje počet trénovacích vzorků, které jsou zpracovány modelem před aktualizací vah [57].

⁵Její cílem je minimalizovat ztrátovou funkci, což znamená najít nejlepší možné hodnoty parametrů (vah) modelu, které vedou k co nejpřesnějším předpovědím výsledků [58].

⁶Jeden z optimalizačních algoritmů, který je velmi efektivní a má nízké paměťové nároky [59].

Původní dataset: D1		Publikovaná				Dosažená		
Testovací Přesnost [%]		82,14				83,49		
		Ověřováno na						
Trénováno na		D1	D2	D3	D4	D5	D6	D7
D1	Přesnost [%]		49,07	26,43	53,76	29,40	16,29	62,41
	Preciznost [%]		81,89	28,89	50,03	45,21	–	41,20
	Citlivost [%]	–	43,31	33,04	17,81	18,90	–	62,27
	F1 skóre [%]		56,65	30,83	26,27	26,65	–	49,59
D_O	Test. Přesnost [%]	83,49	79,85	94,92	59,26	87,39		67,77
	Preciznost [%]	91,07	97,34	97,43	88,71	93,47	–	44,52
	Citlivost [%]	74,26	75,85	92,18	13,64	87,54		34,57
	F1 skóre [%]	81,81	85,25	94,73	23,64	90,41		38,92
$D_{ALL} - D_O$	Přesnost [%]	53,53	70,67	51,25	52,31	45,96	69,18	67,59
	Preciznost [%]	66,54	75,78	81,42	49,08	84,97	–	42,98
	Citlivost [%]	14,20	90,89	2,24	83,87	24,78	–	28,01
	F1 skóre [%]	23,40	82,65	4,35	61,93	38,37	–	33,92

Tab. 3.3: Výsledky ověřování publikovaného řešení *CNN LSTM* ([54]) ve scénářích popsaných v 3.2.1.

Výsledkem druhého scénáře je, že by případný model s pevnými váhami použitý pro klasifikaci jiných než trénovacích dat, nebyl o příliš lepší než náhodný výběr, a to pouze pro klasifikaci na datech datasetů D4 a D7. Z ostatních sledovaných parametrů je patrné, že má model obecně problém klasifikovat skutečné NLoS vzorky – nejvíce pro vzorky datasetu D4, kdy model zachytí pouze necelých 18 % všech NLoS vzorků. Vzhledem k tomu, že datasety D1 a D3 jsou naměřeny na stejném kanálu a modulu stejného výrobce, lze velmi odlišné výsledky a zároveň nízké schopnosti modelů přisuzovat spíše odlišnostem v prostorové doméně, a to i přesto, že byl dataset D1 získán měřením v 7 různých vnitřních prostředích. Odlišnosti nelze najít ani v obecném charakteru vnitřních prostředí, neboť až na dataset D5 pochází všechny vzorky z typologicky podobných oblastí (laboratoře, chodby, kanceláře).

Při ověřování schopností modelu tréninkem a ověřováním na stejném datasetu byla až na výjimku datasetů D4 a D7 publikovaná přesnost modelu řádově zachována či dokonce navýšena (79,85 – 90,90 %). Nejlepších výsledků je dosaženo u datasetů, jejichž vzorky model v předchozím scénáři klasifikoval s nejmenší přesností. Při tréninku a ověřování na datech datasetů D4 a D7 došlo k výraznému poklesu přesnosti modelu v porovnání s tréninkem a ověřováním na zbylých datasetech. Model v obou případech detekuje velmi malé množství NLoS vzorků. V případě vzorků datasetu D4 se pravděpodobně nedokáže vypořádat s diverzitou UWB zařízení napříč datase-tem, u vzorků datasetu D7 vykazuje navíc vysokou četnost falešně pozitivních chyb

a může mít problém s různorodostí vzorků způsobenou charakterem prostředí jejich původu.

Z posledního scénáře je patrné, že trénink modelu na širší datové sadě vede k celkové vyšší přesnosti klasifikace než při tréninku na 1 datasetu a ověřování na jiném. Sledování pouze údajů o přesnosti klasifikace může v tomto případě zkreslovat skutečné schopnosti modelu. Model totiž vykazuje obecně nízkou schopnost identifikace NLoS vzorků, která je nejvýraznější u datasetů D1 a D3, v případě datasetu D3 se schopnost detekce dokonce blíží 0.

Na základě dosažené přesnosti při ověřování s využitím stejného datasetu je toto publikované řešení klasifikováno jako úspěšně ověřené. Vlastní implementace modelu nese v rámci souboru s implementovanými řešeními označení *CNNLSTM*. Způsobem popsaným v 3 byly zjištěny hodnoty následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Počet parametrů modelu: 934 767.
- Velikost modelu: 115,05 kB.
- FLOPs: 20 327.
- Špičková paměťová náročnost metodou PSUtil: 39,22 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 3 689,84 kB.
- Špičková paměťová náročnost metodou Tensorflow: 33 286,75 kB.
- Průměrný čas klasifikace 1 vzorku, optimální velikost dávky a propustnost: Nebylo možné určit.

I přes nízký počet FLOPs a relativně nízký počet trénovatelných parametrů modelu je výpočetní náročnost modelu vysoká. Z toho důvodu se stanoveným způsobem nepodařilo určit metriky propustnosti, optimální velikosti dávky a průměrného času klasifikace 1 vzorku. Model se podařilo ověřit pouze při jeho nasazení na GPU. Příčinou může být i horší optimalizace příslušných tensorflow knihoven pro výpočty realizované prostřednictvím CPU. Paměťová náročnost během běhu při porovnání s ostatními řešeními patří rovněž mezi vyšší. Výhodou modelu je charakter vstupních dat, který umožňuje jeho přenositelnost na data pocházející z jiného zařízení. Přestože velikost samotného modelu v paměti je nízká, jeho nasazení na zařízeních s omezenými zdroji je bez další optimalizace vyloučeno. Nevýhodou je vysoká výpočetní náročnost při spíše průměrné klasifikační přesnosti. Z provedených scénářů vyplývá, že je model použitelný pouze v případech, kdy je trénován a provozován na datech ze stejné domény.

3.2.3 Novel CNN Model

Publikace [55] navrhuje řešení (v publikaci označené jako „Novel CNN Model“) využívající konvoluční neuronové sítě, které provádějí klasifikaci z impulsní odezvy

kanálu. Cílem konvolučních vrstev je extrakce rysů z CIR a jejich předání plně propojeným vrstvám.

Autoři prováděli experimenty s hodnotou dropoutu. Byly porovnávány výsledky pro hodnoty 0,5 a 0,8. Pro dropout 0,5 je publikovaná přesnost architektury vyšší (92,67 % proti 90,06 %) za cenu o přibližně 8 % delšího tréninkového času. Struktura publikované architektury je následující:

1. Konvoluční vrstva s 10 filtry, velikostí jádra 4 a aktivační funkcí ReLu. Nevyužívá padding pro zachování rozměrů vstupních dat. Filtr se posouvá o 1 pixel (parametr „strides“ (kroky)).
2. Konvoluční vrstva s 20 filtry, krokem 2, velikostí jádra 5 a aktivační funkcí ReLu. Využívá padding v podobě přidání nul kolem okrajů vstupních dat pro zachování rozměrů dat.
3. Maxpooling s velikostí okna 2 a krokem 1.
4. Konvoluční vrstva s 20 filtry, krokem 1, velikostí jádra 4 a aktivační funkcí ReLu. Nevyužívá padding.
5. Konvoluční vrstva s 40 filtry, krokem 2, velikostí jádra 4 a aktivační funkcí ReLu. Využívá padding v podobě přidání nul kolem okrajů.
6. Maxpooling s velikostí okna 2 a krokem 1.
7. Plně propojená vrstva se 128 neurony a aktivační funkcí ReLu.
8. Plně propojená vrstva s 64 neurony a aktivační funkcí ReLu.
9. Plně propojená (výstupní) vrstva s 1 neuronem a sigmoid jako aktivační funkce.

Vstupní vektor konvoluční vrstvy má rozměr 1×1016 a je tvořen impulsní odezvou kanálu. Architektura byla trénována s využitím neveřejného datasetu, který byl získán vlastním měřením autory řešení. Měření proběhlo s využitím modulu DWM 1000 ve 4 různých prostředích. Použitý kanál a parametry tréninku nejsou v publikaci uvedeny. K dispozici jsou pouze grafické průběhy učení, ze kterých vyplývá, že přesnost modelu byla po provedení 20 epoch téměř konstantní. Z důvodu chybějících informací o tréninku byly zbylé tréninkové parametry převzaty z ověřovaného řešení 3.2.2 a pro první a druhý scénář byl jako tréninkový dataset z důvodů popsaných v 3.2.1 zvolen D1. Protože publikace [55] neuvádí způsob rozdělení datasetu na trénovací, validační a testovací, bylo toto rozdělení stanoveno porovnáním výkonnosti modelu při následujících hodnotách rozdělení:

- $\frac{1}{6}$ datasetu jako testovací data, zbylá část jako trénovací a validační. Způsob stanovení velikosti trénovací a validační části byl shodný, jako u ověřovaného řešení 3.2.2. V celém intervalu testovaných velikostí validační části dosahoval model přesnosti klasifikace v okolí 85 %. Shodně jako u zmíněného předchozího ověřovaného řešení byla pro účely porovnávání s druhým přístupem velikost trénovací sady nastavena na $\frac{4}{6}$ a velikosti testovací a validační sady shodně

Původní dataset: D1		Publikovaná				Dosažená		
Testovací Přesnost [%]		92,67				83,70		
		Ověřováno na						
Trénováno na		D1	D2	D3	D4	D5	D6	D7
D1	Přesnost [%]		70,24	60,75	46,52	67,31	12,19	37,11
	Preciznost [%]		84,93	57,21	46,33	69,96	–	28,41
	Citlivost [%]	–	74,50	82,86	98,46	90,86	–	73,59
	F1 skóre [%]		79,38	67,68	63,05	79,05	–	41,00
D_O	Test. Přesnost [%]	85,25	93,35	99,52	96,53	95,52		92,18
	Preciznost [%]	86,44	98,14	99,90	96,63	98,16	–	82,90
	Citlivost [%]	83,60	93,11	99,13	95,84	95,18		92,82
	F1 skóre [%]	85,00	95,56	99,51	96,24	96,65		87,58
$D_{ALL} - D_O$	Přesnost [%]	55,65	23,64	56,77	64,07	39,12	98,86	64,51
	Preciznost [%]	58,24	97,81	68,11	65,38	70,27	–	39,04
	Citlivost [%]	39,92	0,67	24,15	47,41	17,87	–	34,80
	F1 skóre [%]	47,37	1,32	35,66	54,96	28,50	–	36,80

Tab. 3.4: Výsledky ověřování publikovaného řešení *Novel CNN* ([55]) ve scénářích popsaných v 3.2.1.

na $\frac{1}{6}$. Při tomto rozdělení datasetu bylo při ověřování modelu dosaženo přesnosti 85,25 %.

- Rozdělení datasetu na testovací, validační a trénovací v poměru 6 : 2 : 2, které bylo převzato z ověřovaného řešení 3.2.5. Při tomto rozdělení datasetu bylo při ověřování modelu dosaženo přesnosti 84,87 %.

Z provedených experimentů týkajících se rozdělení datasetu na trénovací, validační a testovací vyplynulo, že vliv validační části dat na výkonnost tohoto modelu není zásadní. Všechny scénáře byly provedeny při obou variantách rozdělení datasetu a bylo dosaženo téměř shodných výsledků. Ze stejných důvodů, jako v případě ověřovaného řešení 3.2.2, byla pro ověřování tohoto řešení stanovena hodnota velikosti trénovací části dat na $\frac{4}{6}$, validační části dat na $\frac{1}{6}$ a testovací části dat na $\frac{1}{6}$. Výsledky ověřování publikovaného řešení pro zmíněné rozdělení jsou prezentovány v tabulce 3.4.

Autory prezentovaná přesnost při tréninku na neveřejném datasetu je 92,67 %, při ověřování modelu s využitím datasetu D1 bylo dosaženo přesnosti 85,25 %. Rozdíl v přesnosti klasifikace je s největší pravděpodobností způsoben rozdílem dat, na kterých byly ověřovány schopnosti modelu. V publikaci není specifikováno, z jakých typů vnitřních prostředí data pochází. V rámci prvního scénáře byl model trénován a ověřován pouze na datech z prvních 4 místností datasetu D1, aby se scénář co nejvíce přiblížil podmínkám z publikace, přičemž přesnost klasifikace těchto dat

dosáhla 83,70 %.

Ze druhého scénáře vyplývá, že model trénovaný na datech datasetu D1 dokáže zachytit velkou část NLoS vzorků ostatních datasetů, ale zároveň produkuje nemalé množství chyb, kdy například pro vzorky datasetu D4 byla více než polovina modelem klasifikovaných NLoS vzorků ve skutečnosti LoS. Celková přesnost klasifikace se napříč datasety liší a její maximální hodnota je přibližně 70 %. Přesnost klasifikace pro vzorky datasetu D6 vybočuje z přesností modelu při klasifikaci vzorků jiných datasetů, což je způsobeno tím, že dataset obsahuje pouze LoS vzorky a model vykazuje problémy s falešně pozitivní detekcí. Velmi nízká přesnost je dosažena i při klasifikaci dat z datasetu D7, což může být způsobeno velmi rozdílným charakterem prostředí, jiným typem použitého UWB modulu a jiným kanálem.

Při ověřování schopností modelu tréninkem a ověřováním na stejném datasetu bylo s výjimkou datasetu D1 v porovnání s publikovanou přesností modelu dosaženo vyšší nebo téměř shodné přesnosti. Model vykazuje výborné výsledky pro klasifikaci dat z datasetů D2, D3, D4, D5 i D7 a rozdíly mezi výkonností modelu na těchto datasetech jsou minimální. Ve všech těchto případech (stále s výjimkou D1) zachytí většinu NLoS vzorků a také činí velmi malé procento chyb. Nejvyšší míru chyb činí pro vzorky datasetu D7, a to dokonce napříč všemi scénáři. Příčinou může být vysoká rozmanitost vzorků stejných tříd způsobená charakterem prostředí jejich původu.

Při tréninku na širší datové sadě vykazuje model téměř ve všech případech horší výsledky než při tréninku na datasetu D1. Největším rozdílem oproti 2. scénáři je, že model nedokáže zachytit většinu NLoS vzorků a v případě ověřování na datech datasetu D2 nezachytí téměř žádné NLoS vzorky.

Publikované řešení je klasifikováno jako úspěšně ověřené zejména na základě výsledků 3. scénáře, v rámci kterého model dosahoval shodné nebo vyšší než publikované přesnosti u 5 z 6 datasetů. Vlastní implementace modelu nese v rámci souboru s implementovanými řešeními označení *NovelCNN*. Způsobem popsáním v 3 byly zjištěny hodnoty následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Počet parametrů modelu: 1 299 499.
- Velikost modelu: 5 111,69 kB.
- FLOPs: 6 985 115.
- Špičková paměťová náročnost metodou PSUtil: 33,87 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 2 114,90 kB.
- Špičková paměťová náročnost metodou Tensorflow: 38 569,75 kB.
- Průměrný čas klasifikace 1 vzorku: 9,326 ms.
- Optimální velikost dávky: 1024.
- Propustnost s velikostí dávky 4: 381 S/s.
- Propustnost s velikostí dávky 16: 1 184 S/s.

- Propustnost s velikostí dávky 64: 2 672 S/s.
- Propustnost s optimální velikostí dávky: 5 613 S/s.

Přestože velikost samotného modelu je velmi nízká, z uvedených parametrů vyplývá, že jeho paměťová i výpočetní náročnost je v rámci souboru publikovaných řešení spíše nadprůměrná. Rozměr vstupního vektoru, počet konvolučních vrstev a hloubka architektury vedou k vysokému počtu parametrů modelu a FLOPs. Výsledkem je inferenční čas přesahující 9 ms a v porovnání s dalšími publikovanými řešeními i nižší propustnost. Nasazení modelu na zařízení s omezenými zdroji by bez další optimalizace mohlo být problematické, zejména s ohledem na klasifikační čas a propustnost modelu. Největší výhodou modelu je jeho vysoká přesnost a možnost přenositelnosti na data pocházející z jiného zařízení díky povaze vstupních dat. Nevýhodou modelu je vyšší výpočetní náročnost a nízká propustnost. Použití předtřénované varianty modelu závisí na povaze cílové aplikace, kdy by mohl být použit např. jako podpůrný klasifikátor. Model je vhodný zejména pro nasazení na centrální prvek (s dostatečným výkonem) určený ke klasifikaci vzorků pocházejících ze stejné domény jako data, na kterých byl model trénován.

3.2.4 Resnet

Publikace [60] představuje architektury využívající konvoluční vrstvy – Resnet, plně konvoluční síť *FCN* (Fully Convolutional Network) a model *Encoder* (kombinace konvolučních a plně propojených vrstev), které v testovaných scénářích dosahují téměř shodných výsledků. Metody byly porovnávány v různých scénářích. Prvním scénářem byla klasifikace vzorků pocházejících ze simulované chodby, druhý scénář ověřoval generalizační schopnosti modelů. V prvním scénáři dosahovala nejlepší výsledků poslední ze zmíněných metod, v generalizačním scénáři naopak architektura Resnet. Rozdíly napříč navrženými metodami byly minimální a dosahovaly maximálně 1 %. Pro ověřování v rámci této práce byla zvolena architektura Resnet, neboť se v souboru ověřovaných řešení jedná o nový typ architektury. Navržené metody *FCN* a *Encoder* jsou popsány v kapitole 3.3, která diskutuje další publikované architektury, jež nebyly zvoleny k ověřování.

Resnet (Residual Network) je typ architektury, která se místo učení mapování mezi vstupem a výstupem učí reziduální funkce. Residuální sítě jsou složeny z konvolučních vrstev, mezi kterými existují přímé cesty (skip connections), které slouží k přímému přenosu informací mezi vrstvami a pomáhají zmírnit problém mizejícího gradientu, díky čemuž je možné použít hlubší architektury. Publikovaný model se skládá ze 3 reziduálních bloků, jejichž struktura je následující:

1. Konvoluční vrstva s N filtry, velikostí jádra M a krokem 1. Padding není použit.

2. Aplikace dávkové normalizace a aktivační funkce ReLu.
3. Konvoluční vrstva s N filtry, velikostí jádra M a krokem 1. Padding není použit.
4. Aplikace dávkové normalizace a aktivační funkce ReLu.
5. Konvoluční vrstva s N filtry, velikostí jádra M a krokem 1. Padding není použit.
6. Aplikace dávkové normalizace a aktivační funkce ReLu.

Po 3 reziduálních blocích je aplikován globální pooling a výstupní vrstva je tvořena 2 neurony, použitou aktivační funkcí je softmax. Pro první reziduální blok je počet filtrů konvolučních vrstev $N = 64$ a počet jader $M = 8$, pro druhý blok $N = 128$ a $M = 5$ a pro poslední blok $N = 128$ a $M = 3$.

Vstupní data modelu jsou tvořena impulsní odezvou kanálu ořezanou na délku 170 bodů v okolí nejvyššího vrcholu. Navržené metody byly testovány na vlastních datech naměřených v simulované chodbě a v reálném průmyslovém skladu na nevedeném kanálu s využitím modulu DWM 1000. Prvním scénářem byla klasifikace dat z chodby, na kterých bylo použitím této metody dosaženo přesnosti 93 %. Cílem druhého scénáře bylo otestovat schopnosti generalizace modelu na datech z průmyslového skladu, kdy byl model trénován pouze na datech z 5 přijímačů a ověřován na datech z 6. přijímače. V tomto scénáři bylo dosaženo klasifikační přesnosti 92 % a klasifikační přesnost tohoto scénáře bude použita jako referenční pro účely ověřování modelu v kontextu této práce. Rozdělení datasetu vyplývá z popisu scénáře – $\frac{5}{6}$ pro trénink a $\frac{1}{6}$ pro ověřování. Validací sada byla zvolena shodně, jako v případě 3.2.2, tedy jako $\frac{1}{6}$ a velikost trénovací sady analogicky odpovídá $\frac{4}{6}$. Model byl trénován ve 100 epochách a použitá velikost dávky byla 128. Zbylé tréninkové parametry (optimalizátor a rychlost učení) byly nastaveny shodně jako v případě 3.2.2. Pro první 2 scénáře byl jako tréninkový dataset zvolen D1 z důvodů uvedených v 3.2.1.

Autory prezentovaná přesnost řešení při tréninku modelu na neveřejném datasetu je 92,00 % a při tréninku a testování na datech z chodby 93,00 %. Při trénování a testování modelu na datasetu D1 bylo v rámci ověřování pro účely této práce dosaženo klasifikační přesnosti 82,09 %. Při trénování a ověřování na datech pocházejících z 1 místnosti datasetu D1 bylo dosaženo klasifikační přesnosti 74,95 %. Rozdíl v přesnosti klasifikace je s největší pravděpodobností způsoben rozdílem dat, na kterých byly ověřovány schopnosti modelu. Výsledky ověřování jsou prezentovány v tabulce 3.2.4.

Výsledky druhého scénáře jsou pro model trénovaný na datech datasetu D1 velmi různorodé. Model ve většině případů dokáže zachytit velké množství skutečných NLoS vzorků. Z hodnot preciznosti pro některé datasety však vyplývá, že tak činí způsobem, při kterém označuje většinu vzorků jako NLoS a činí tak vysoké množství chyb. Toto tvrzení podporuje i klasifikační přesnost vzorků z datasetu D6, který

Původní dataset: D1		Publikovaná				Dosažená		
Testovací Přesnost [%]		92,00				82,09		
		Ověřováno na						
Trénováno na		D1	D2	D3	D4	D5	D6	D7
D1	Přesnost [%]		65,04	70,07	46,28	79,24	1,66	31,22
	Preciznost [%]		83,86	66,37	46,26	81,65	–	30,13
	Citlivost [%]	–	67,51	80,39	99,94	89,54	–	99,78
	F1 skóre [%]		74,80	72,71	63,25	85,42	–	53,30
D_O	Test. Přesnost [%]	82,09	96,86	99,66	94,43	99,12		81,33
	Preciznost [%]	85,57	99,27	99,85	96,47	99,52	–	85,18
	Citlivost [%]	77,21	96,63	99,47	91,30	99,18		75,85
	F1 skóre [%]	81,17	97,93	99,66	93,82	99,35		80,24
$D_{ALL} - D_O$	Přesnost [%]	60,55	56,61	60,27	47,28	68,81	30,50	46,11
	Preciznost [%]	64,16	74,16	66,18	46,70	83,46	–	32,83
	Citlivost [%]	47,81	66,83	40,73	99,13	67,40	–	77,90
	F1 skóre [%]	54,79	70,31	50,42	63,49	75,48	–	46,19

Tab. 3.5: Výsledky ověřování publikovaného řešení *Resnet* ([60]) ve scénářích popsaných v 3.2.1.

obsahuje pouze LoS vzorky, které jsou téměř všechny klasifikovány špatně.

Při ověřování schopností modelu tréninkem a ověřováním na stejném datasetu bylo s výjimkou datasetů D1 a D7 v porovnání s publikovanou přesností modelu dosaženo dokonce vyšší přesnosti. Model dosahuje téměř absolutní přesnosti při klasifikaci dat z datasetů D3 a D5. Výborných výsledků dosahuje i při klasifikaci vzorků datasetů D2 a D4, u kterých klasifikační přesnost modelu přesahuje 94 %. U vzorků zbylých datasetů (D1 a D7) model špatně klasifikuje NLoS vzorky jako LOS v přibližně 25 % případech a klasifikační přesnost dosahuje v obou případech mírně přes 80 %. Příčinou může být vysoká rozmanitost vzorků stejných tříd způsobená charakterem prostředí jejich původu – v případě D1 se jedná o několik různých vnitřních prostředí, v případě D7 o průmyslový sklad.

Při tréninku na širší datové sadě vykazuje model v polovině případů horší výsledky než při tréninku na datasetu D1. Zatímco model činí přibližně stejné procento falešně pozitivních chyb, došlo (s výjimkou vzorků datasetů D2 a D4) k poklesu počtu zachycení skutečných NLoS vzorků. Ve druhé polovině případů jsou výsledky modelu zachovány či mírně navýšeny. Například pro vzorky datasetu D7 činí model méně falešně pozitivních chyb, ale zároveň zachytí výrazně nižší počet skutečných NLoS vzorků. Celkový počet správně klasifikovaných vzorků je však o 15 % vyšší.

Publikované řešení je klasifikováno jako úspěšně ověřené zejména na základě výsledků 3. scénáře, v rámci kterého model dosahoval vyšší než publikované přesnosti

u 4 z 6 datasetů. Vlastní implementace modelu nese v rámci souboru s implementovanými řešeními označení *Resnet*. Způsobem popsáním v 3 byly zjištěny hodnoty následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Počet parametrů modelu: 465 282.
- Velikost modelu: 1 930,38 kB.
- FLOPs: 156 569 484.
- Špičková paměťová náročnost metodou PSUtil: 32,11 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 2 114,90 kB.
- Špičková paměťová náročnost metodou Tensorflow: 4 295 505,25 kB, což je 4,10 GB.
- Průměrný čas klasifikace 1 vzorku: 30,047 ms.
- Optimální velikost dávky: 256.
- Propustnost s velikostí dávky 4: 107 S/s.
- Propustnost s velikostí dávky 16: 265 S/s.
- Propustnost s velikostí dávky 64: 471 S/s.
- Propustnost s optimální velikostí dávky: 589 S/s.

Z uvedených parametrů vyplývá, že i přes nízký počet parametrů modelu a nízkou velikost modelu v paměti je jeho výpočetní a paměťová náročnost během samotného běhu obrovská. Zajímavá je hodnota špičkové paměťové náročnosti získaná metodou tensorflow, která přesahuje 4 GB. Tento údaj udává vysokou hodnotu velikosti paměti alokované během běhu modelu, která není viditelná prostřednictvím ostatních nástrojů. Hluboká architektura vede k vysokému počtu FLOPs a inferenčnímu času přes 30 ms. Z důvodu vysoké paměťové náročnosti není možné dosáhnout vysokých hodnot velikosti dávek a analogicky ani propustností. Model je zcela nevhodný k nasazení na zařízeních s omezenými zdroji. Výhodou modelu je povaha a menší délka vstupních dat, která umožňuje jeho přenositelnost, a také vysoká klasifikační přesnost. Vstupní CIR data jsou ořezána na délku 170 bodů v okolí nejvyššího bodu. V kapitole 1.4.2 bylo popsáno časové okno zpětného vyhledávání, jehož velikost se dle [3], [9] a [10] pohybuje v rozmezí 100–300 ns. Může tedy nastat situace, že ve vstupních datech modelu nemusí být obsažena skutečná první cesta. Nevýhodou modelu je vysoká výpočetní a paměťová náročnost, vysoký inferenční čas a nízká propustnost. Hloubka modelu je překážkou pro snadnou interpretaci a ladění. Model je vhodný zejména pro nasazení na centrální prvek s dostatečným výkonem určený ke klasifikaci vzorků pocházejících ze stejné domény jako data, na kterých byl model trénován.

3.2.5 MLP využívající přenesené učení

Využití přeneseného učení (TL – Transfer Learning) v kombinaci s metodami hlubokého učení je představeno v publikaci [61]. Představeno bylo řešení využívající samotnou dopřednou neuronovou síť (v publikaci označeno jako „MLP“ – Multilayer Perceptron) a také řešení využívající tuto síť v kombinaci s konvoluční neuronovou sítí. Motivací pro využití přeneseného učení je v tomto případě snaha o přesnější adaptaci modelu na nová prostředí, která bude zároveň časově méně náročná. Publikovaný scénář předpokládá naměření alespoň malého množství vzorků v novém prostředí, jež budou využity k úpravě vah některých vrstev modelu, díky čemuž by měl model dosahovat lepších výsledků při klasifikaci vzorků z nového prostředí.

Model byl trénován i ověřován na nevěřejném datasetu, který byl naměřen v prostředí laboratoře a třídy. V rámci experimentu se mimo LoS a NLoS pracovalo rovněž s weak NLoS scénářem, který je pro účely ověřování v rámci této práce vynechán. Pro každý scénář bylo v každé místnosti naměřeno shodně celkem 3100 vzorků s využitím zařízení EKV1000. 3000 vzorků bylo použito pro trénink modelu a zbylých 100 na přenesené učení. Vzorky pro trénink modelu byly náhodně rozděleny v poměru 6:2:2 na trénovací, testovací a validační data.

Autoři publikace porovnávali výsledky navržených modelů bez a s použitím přeneseného učení. Model byl naučen na vzorcích z 1 místnosti a testován na vzorcích ze 2. místnosti. V případě přeneseného učení byl ještě doučen na 300 vzorcích (100 LoS, 100 NLoS, 100 wLoS) ze 2. místnosti. U obou navrhovaných architektur bylo dosaženo výrazně vyšší přesnosti při použití přeneseného učení. U dopředné neuronové sítě je prezentovaná přesnost s využitím přeneseného učení 75% při tréninku na datech z laboratoře a 90% při trénování na datech ze třídy, zatímco bez využití přeneseného učení přesnost dosahuje pouze 44,11 a 45,67%. U CNN architektury je navýšení přesnosti ještě znatelnější, neboť bez využití přeneseného učení dosahuje publikovaná přesnost hodnot 40,72 a 43,89%, zatímco s přeneseným učením 96,67 a 98,33%. Autoři rovněž uvádějí, že trénování nového modelu pro neznámé prostředí se stejným množstvím vzorků zabere v případě CNN architektury přibližně 5x více času.

Z publikovaného řešení nelze v plném rozsahu určit parametry architektury využívající konvoluční neuronovou síť, byť je popisovaná struktura velmi podobná struktuře publikovaného řešení popsaného v 3.2.3. Z toho důvodu bude pro ověření v rámci této práce použit prezentovaný model dopředné neuronové sítě, který má kromě vstupní a výstupní vrstvy další 3 skryté vrstvy se 128 neurony. Jako trénovatelné jsou pro účely přeneseného učení nastaveny pouze poslední 2 vrstvy modelu (poslední skrytá vrstva a výstupní vrstva). Aktivační funkcí pro skryté vrstvy je ReLu a jako tréninkový optimalizátor byl zvolen Adam. Vstupní data publikova-

Původní dataset: D1		Publikovaná				Dosažená		
Testovací Přesnost [%]		75,00/90,00				82,36/84,27		
		Ověřováno na						
Trénováno na		D1	D2	D3	D4	D5	D6	D7
D1	Přesnost [%]		73,66	88,31	85,76	86,94	100,00	88,63
	Preciznost [%]		91,33	92,18	86,50	91,79	–	74,68
	Citlivost [%]	–	72,61	83,52	82,00	88,68	–	93,35
	F1 skóre [%]		80,91	87,64	84,19	90,21	–	82,98
$D_{ALL} - D_O$	Přesnost [%]	79,41	63,66	84,30	84,78	81,24	100,00	72,22
	Preciznost [%]	84,10	89,18	84,50	86,24	84,10	–	52,22
	Citlivost [%]	72,53	60,00	83,69	79,82	89,25	–	75,95
	F1 skóre [%]	77,89	71,73	84,10	82,91	86,60	–	61,88

Tab. 3.6: Výsledky ověřování publikovaného řešení *MLP TL* ([61]) ve scénářích popsaných v 3.2.5.

ného modelu tvoří reálná a imaginární část impulzní odezvy kanálu, každá o délce 1016 vzorků. Protože některé datasety použité pro ověřování publikovaného řešení nemají k dispozici jednotlivé složky CIR, byla vstupní vrstva pro účely ověřování této architektury upravena tak, aby jako vstup přijímala CIR data o rozměru 1x1016. Rozdělení datasetu pro trénink je totožné jako u publikovaného řešení. Chybějící parametry tréninku byly nastaveny shodně jako u 3.2.2 a ověřovací scénáře byly pro použití s přeneseným učením upraveny následovně:

1. Trénování na datech z 1 místnosti a ověření na datech ze druhé místnosti datasetu D1 a opačně. Cílem upraveného scénáře je ověřit publikovanou přesnost modelu. V rámci publikace jsou využity vzorky ze třídy a laboratoře, pro ověřování jsou využity vzorky ze 2 různých kanceláří, neboť v rámci datasetu neexistuje podobnější obecné vnitřní prostředí.
2. Scénář ponechán beze změn a jako trénovací dataset zvolen D1.
3. Vypuštěn, neboť využití přeneseného učení pro testování na vzorcích ze stejného datasetu odpovídá rozšíření původní trénovací sady.
4. Scénář ponechán beze změn.

Pro účely ověřování v rámci této práce bylo pro trénink modelu v rámci přeneseného učení využito 5 % a 10 % LoS i NLoS vzorků z každého datasetu a rovněž bylo zajištěno použití vyvážené datové sady. Výsledky ověřování publikovaného řešení modelu s využitím 5 % vzorků pro přenesené učení (autoři řešení použili pro přenesené učení 3,23 % vzorků) jsou prezentovány v tabulce 3.6. Při ověřování řešení s využitím 10 % vzorků pro přenesené učení došlo v některých případech k navýšení přesnosti klasifikace maximálně v řádu jednotek procent, ve zbylých byla přesnost

klasifikace zachována.

Autory prezentovaná přesnost řešení při tréninku i ověřování na neveřejném datasetu je 75 a 90 % v závislosti na původu trénovací datové sady. Přesnost dosažená při ověřování tohoto řešení je 82,36 % (trénováno na první místnosti D1, ověřováno na druhé) a 84,27 % (trénováno na druhé místnosti, ověřováno na první), což odpovídá průměru autory prezentovaných přesností. Model během tohoto scénáře vykazoval vysoké hodnoty všech dalších sledovaných metrik (preciznost, citlivost a F1 skóre), které se pohybovaly průměrně v okolí 80 %.

Při ověřování scénáře 2 na modelu naučeném na celém datasetu D1 a doučeném na zbylých datasetech dosahoval model velmi vysoké přesnosti nad 85 % (D3, D4, D5 a D7) s výjimkou ověřování na datech datasetu D2, během kterého bylo dosaženo přesnosti klasifikace 73,66 %. Výsledky pro ostatní datasety korespondují s výsledky 1. scénáře, výsledky pro všechny datasety odpovídají publikovaným hodnotám. Doučený model velmi přesně identifikuje většinu NLoS vzorků, a zároveň vykazuje minimální výskyt falešně pozitivních výsledků.

Během posledního scénáře, při kterém byl model trénován na širší datové sadě, model dosahuje řádově stejných výsledků, jako v předchozích scénářích. Výjimkou jsou pouze vzorky datasetu D2, při jejichž klasifikaci měl model problém s falešně negativními chybami, kdy označoval NLoS vzorky jako LoS. Ve všech případech došlo alespoň k mírnému poklesu výkonnosti. Rozdíl mezi oběma scénáři je ve velikostech původní trénovací sady, zatímco velikost trénovací sady pro přenesené učení zůstává stejná.

Na základě uvedených výsledků ověřování je publikované řešení klasifikováno jako úspěšně ověřené, neboť ve všech případech ověřování je dosaženo řádově stejných výsledků jako u publikovaného řešení. Vlastní implementace modelu nese v rámci souboru s implementovanými řešeními označení *MLP_TL*. Způsobem popsaným v části 3 byly zjištěny hodnoty parametrů pro odvození výpočetní a paměťové náročnosti modelu. Hodnoty byly zjišťovány pro model s původní i upravenou velikostí vstupu. V rámci tabulky 3.10 je prezentována hodnota pro původní velikost vstupu (1016 reálných a imaginárních CIR dat), zatímco v následujícím výčtu parametrů jsou pro možnosti porovnání vlivu vstupu na náročnost modelu prezentovány obě varianty ve formátu „hodnota pro původní velikost vstupu – hodnota pro upravenou velikost vstupu“.

- Počet parametrů modelu: 293 377 – 163 329.
- Velikost modelu: 1 166,26 – 658,26 kB.
- FLOPs: 586 369 – 326 273.
- Špičková paměťová náročnost metodou PSUtil: 25,67 – 22,54 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 1 424,85 – 419,32 kB.
- Špičková paměťová náročnost metodou Tensorflow: 3 050,50 – 1 526,50 kB.

- Průměrný čas klasifikace 1 vzorku: 2,451 – 2,376 ms.
- Optimální velikost dávky: 2048.
- Propustnost s velikostí dávky 4: 871 – 1 355 S/s.
- Propustnost s velikostí dávky 16: 4 802 – 7 608 S/s.
- Propustnost s velikostí dávky 64: 16 735 – 23 480 S/s.
- Propustnost s optimální velikostí dávky: 177 015 – 233 363 S/s.

Ze zjištěných parametrů je zřejmé, že výpočetní i paměťová náročnost modelu je poměrně nízká. Rozdíl v hodnotách jednotlivých parametrů odpovídá očekávání. Pro delší vstup je vyšší počet parametrů modelu i FLOPs a paměťová náročnost modelu. Z důvodu vyššího počtu FLOPs je inferenční čas vyšší a propustnost modelu naopak nižší. Nízký počet FLOPs a jednoduchá architektura umožňuje dosažení nízkých inferenčních časů a vysoké propustnosti. Výhodou modelu je jednoduchá architektura, která umožňuje jeho jednoduchou implementaci a usnadňuje ladění modelu. Nízká náročnost umožňuje nasazení i na zařízení s omezenými zdroji. Charakter vstupních dat umožňuje jeho přenositelnost, což je klíčová vlastnost v kontextu využitého principu přeneseného učení. Nevýhodou modelu vyplývá z povahy navrženého řešení. Model bez aplikace přeneseného učení dosahuje podprůměrných výsledků a pro aplikaci přeneseného učení je nezbytná existence alespoň malého množství oštitkovaných vzorků z cílové domény. V praxi to znamená nutnost provedení alespoň malé měřicí kampaně v cílovém prostředí, což může přinést nárůst složitosti a délky implementačního procesu. Publikované řešení a provedené experimenty prokazují, že s využitím přeneseného učení lze využít i předtrénovaný model a s malým množstvím vzorků jej dotrénovat tak, aby dosahoval vysoké klasifikační přesnosti.

3.2.6 SVM LDA

Řešení spadající mimo algoritmy hlubokého učení je představeno v publikaci [62]. Navržené řešení využívá ke klasifikaci lineární diskriminační analýzu LDA (Linear Discriminant Analysis) v kombinaci s SVM. LDA je použita k redukcí dimenze vstupních dat před jejich klasifikací pomocí SVM.

Model byl trénován na neveřejném datasetu malého rozsahu. Pro získání dat byl pro účely publikace využit kit EVB1000 a data byla naměřena ve 2 různých vnitřních prostředích. Bylo naměřeno přibližně 2000 vzorků a jako vstupní data byla využita impulsní odezva kanálu, která byla zarovnána na 100 vzorků následujících po první vzestupné hraně. Metodologie detekce vzestupné hrany není v publikovaném řešení uvedena.

Vstupní data byla rozdělena v poměru 1:7 na trénovací a testovací. Autoři provedli porovnání dosažených výsledků za použití různých jader pro SVM, přičemž

Původní dataset: D1		Publikovaná				Dosažená		
Testovací Přesnost [%]		100,00				74,61		
		Ověřováno na						
Trénováno na		D1	D2	D3	D4	D5	D6	D7
D1	Přesnost [%]		48,51	71,29	57,09	68,28	70,48	63,10
	Preciznost [%]		99,06	85,00	70,25	97,41	–	41,95
	Citlivost [%]	–	33,32	51,16	12,51	54,72	–	63,25
	F1 skóre [%]		49,87	63,87	21,23	70,08	–	50,44
D_O	Test. Přesnost [%]	74,61	68,99	80,83	57,80	86,08		59,74
	Preciznost [%]	95,26	97,83	94,82	86,30	97,09	–	41,24
	Citlivost [%]	51,79	61,01	64,89	10,39	81,94		83,64
	F1 skóre [%]	97,10	75,15	77,05	18,55	88,88		55,24
$D_{ALL} - D_O$	Přesnost [%]	56,02	76,14	65,02	56,57	38,39	89,41	75,29
	Preciznost [%]	68,45	79,83	81,20	75,59	88,67	–	69,33
	Citlivost [%]	22,34	92,28	38,37	9,00	10,59	–	30,09
	F1 skóre [%]	33,68	85,60	52,11	16,08	18,92	–	41,97

Tab. 3.7: Výsledky ověřování publikovaného řešení *SVM LDA* ([62]) ve scénářích popsaných v 3.2.1.

nejlepších výsledků bylo dosaženo při použití kvadratického a polynomiálního jádra 3. stupně. S využitím LDA byl zaznamenán výrazný pokles tréninkového času, ale negativním efektem byl pokles přesnosti modelu při trénování na malém množství vzorků (pokles byl autory zaznamenán při trénování na 594 a méně vzorcích).

Publikovaná přesnost tohoto řešení při trénování na více než 700 vzorcích je 100 % s i bez použití LDA, ale LDA může výrazně snížit výpočetní čas, což může být důležité pro nasazení v reálných systémech [62].

Rozdělení vstupních dat na trénovací a testovací odpovídá způsobu použitému v publikovaném řešení. Velikost validační sady není v publikaci uvedena. Byly provedeny experimenty s velikostí validační sady stejně jako u řešení 3.2.2. Přítomnost validační sady během tréninku neměla vliv na výkonnost modelu a z důvodu malé velikosti trénovací sady nebyla validační sada zahrnuta do tréninku modelu pro účely ověřování v rámci této práce. Pro první 2 scénáře byl jako tréninkový dataset zvolen D1 z důvodů uvedených v 3.2.1.

Všechny ověřovací scénáře byly provedeny s využitím kvadratického a polynomiálního jádra 3. stupně, přičemž lepších výsledků bylo ve většině případů dosaženo při použití druhého z nich, a ty jsou také prezentovány v tabulce 3.7. Autory prezentovaná přesnost řešení při tréninku na neveřejném datasetu je 100 %. Při stejném rozdělení datasetu na trénovací a testovací část jako u publikovaného řešení a trénování na D1 bylo při ověřování modelu dosaženo přesnosti 74,61 %.

Výsledky druhého scénáře pro vzorky z datasetů D2 a D4 ukazují, že by případný model s pevnými váhami dosahoval horších nebo podobných výsledků jako náhodný výběr. Pro vzorky zbylých datasetů se přesnost klasifikace modelu pohybuje v okolí 70 %, pro vzorky datasetu D7 je přesnost 63 %. Model má u vzorků veřejných datasetů problémy primárně s identifikací NLoS vzorků, naopak nevykazuje vysoké množství falešně pozitivních předpovědí. U neveřejného datasetu D7 je situace opačná, model zachytí přibližně $\frac{2}{3}$ všech NLoS vzorků, ale zároveň vykazuje vysoké množství falešně pozitivních chyb.

Při ověřování schopností modelu tréninkem a ověřováním na stejném datasetu byla přesnost ve většině případů navýšena či zachována. Nejvyšší nárůst přesnosti lze pozorovat u vzorků datasetů D2, D3 a D5. Zlepšení vykazuje zejména parametr citlivost a model tak detekuje větší počet skutečných NLoS vzorků. K mírnému poklesu došlo u vzorků datasetu D7, které je způsobeno nárůstem falešně negativních chyb, kdy model špatně klasifikuje LoS vzorky jako NLoS. Stejně jako u vzorků předchozích datasetů i v tomto případě model vykazuje výrazné zlepšení citlivosti.

Výsledky posledního scénáře jsou různorodé stejně jako výsledky předchozích scénářů. Pro vzorky datasetů D2, D6 a D7 vykazuje model trénovaný na širší datové sadě lepší výsledky než při tréninku na datasetu D1. Klasifikace vzorků datasetů D4 je přibližně stejně přesná, u vzorků datasetů D3 a D5 je pokles přesnosti znatelnější. Model vykazuje stejný problém, jako ve druhém scénáři – velmi nízké množství zachycených skutečných NLoS vzorků.

Publikované řešení je klasifikováno jako úspěšně ověřené, protože implementovaná verze tohoto řešení dosahuje stabilních výsledků, i přestože se tyto výsledky výrazně liší od publikovaných, kterých bylo dosaženo tréninkem a ověřováním na neveřejném datasetu malého rozměru.

Vlastní implementace modelu nese v rámci souboru s implementovanými řešeními označení *SVM*. Vzhledem k povaze algoritmu nelze objektivně určit parametry popsané v 3 bez vlastního tréninku algoritmu a jejich počet se může lišit v závislosti na rozložení poskytnutých vstupních dat určených k tréninku modelu. Vzhledem k povaze algoritmu nelze rovněž stanovit počet parametrů ani počet FLOPs, neboť u algoritmů tohoto typu se místo zmíněných metrik používá jako parametr počet podpůrných vektorů. S využitím modelu natrénovaném na datech datasetu D1 byly zjištěny hodnoty následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Počet podpůrných vektorů: 3 540.
- Velikost modelu: 101,33 kB.
- Špičková paměťová náročnost metodou PSUtil: 20,52 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 273,99 kB.
- Průměrný čas klasifikace 1 vzorku: 0,419 ms.

- Optimální velikost dávky: 32.
- Propustnost s velikostí dávky 4: 5 690 S/s.
- Propustnost s velikostí dávky 16: 11 568 S/s.
- Propustnost s optimální velikostí dávky: 14 281 S/s.

Ze zjištěných parametrů lze výpočetní a paměťovou náročnost modelu hodnotit jako velmi nízkou, což jej činí vhodným k nasazení na zařízení s omezeným výpočetním výkonem. Přestože má model inferenční čas menší než 0,5 ms, jeho propustnost je poměrně nízká. Hodnota propustnosti od velikosti dávky 32 je přibližně konstantní. Díky nízkému inferenčnímu času i nízké paměťové náročnosti se jako nejpravděpodobnější příčina jeví nízká úroveň dávkové optimalizace použitých SVM knihoven, které pracují spíše iterativně a sériově. Hlavní výhodou modelu je nízká paměťová a výpočetní náročnost, nízký inferenční čas a díky povaze vstupních dat i přenositelnost. Propustnost řešení je sice nižší v porovnání s algoritmy podobného typu, přesto je dle [34] více než dostačující pro klasifikaci na koncovém zařízení. Propustnost přes 14 000 S/s může být dostačující i pro klasifikaci na centrálním prvku. Nevýhodou je závislost výkonu na výběru jádra, kdy pro některé vzorky může být model výkonnější s použitím jiného typu jádra. Nasazení modelu by tak mělo předcházet testování různých jader.

3.2.7 RF

Do algoritmů strojového učení spadá i řešení představené v [63]. Pro klasifikaci využívá algoritmus náhodného lesa RF (Random Forest). Ten pro dosažení přesnější klasifikace kombinuje výstupy více rozhodovacích stromů – algoritmus vybere třídu, kterou zvolí většina stromů. Princip rozhodovacích stromů je popsán v kapitole 2.2.1.

Vstupními daty klasifikátoru jsou 4 charakteristiky (features – v dalším textu může být použito rovněž označení „vlastnosti“, „rysy“ a „příznaky“) extrahované z impulsní odezvy kanálu. V rámci publikace bylo experimentováno s množstvím charakteristik, které byly modelu předloženy během tréninku, v rozsahu 1–4. Nejlepších výsledků model dosahuje při poskytnutí 3–4 charakteristik, kdy rozdíl v přesnosti klasifikace mezi těmito možnostmi dosahuje pouze hodnoty 2,3%.

Před extrakcí dat z impulsní odezvy kanálu proběhla nejprve normalizace těchto dat tak, aby jejich maximální hodnota byla rovna jedné. Z těchto normalizovaných dat byly vypočteny následující parametry, které slouží jako vstup modelu:

- Průměr (Mean) μ .
- Směrodatná odchylka (Standard deviation) σ .
- Šikmost (Skewness) s podle rovnice 3.1.

$$s = \frac{\sum_{i=0}^{L-1} (|h_i| - \mu)^3 / L}{\sigma^3}. \quad (3.1)$$

- Kurtóza (špičatost – kurtosis) κ podle rovnice 3.2.

$$\kappa = \frac{\sum_{i=0}^{L-1} (|h_i| - \mu)^4 / L}{\sigma^4} \quad (3.2)$$

V rovnicích 3.1 a 3.2 představuje h_i normalizovanou amplitudu i -tého vzorku CIR a obě byly převzaty z publikace [63], v rámci které je představeno popisované řešení.

Model byl trénován i ověřován na neveřejném datasetu, který vznikl měřením v chodbě kancelářského prostředí. Pro měření byla využita univerzální platforma softwarově definovaného rádia (SDR – Software Defined Radio) – konkrétně model N210 jako vysílač i přijímač. Použita byla frekvence 5,75 GHz a šířka pásma 25 MHz. Publikovaná přesnost navrženého modelu při tréninku a ověřování na tomto datasetu je při použití všech 4 charakteristik 97,3% a při použití 3 z nich (směrodatná odchylka, šikmost a kurtóza) 95%.

Navržená architektura používá 16 stromů a jako kritérium pro rozdělení je použita Shannonova entropie. Kritérium rozdělení slouží v těchto algoritmech jako základ pro konstrukci stromů a definici klasifikačních pravidel. Vstupní data byla rozdělena v poměru 1:4 na trénovací a testovací. Validací sada není přímo použita, místo ní je využita tzv. chyba Out-Of-the-Bag (OOB). Během tréninku každého stromu jsou trénovací vzorky vybírány náhodně s opakováním (bootstrap). Přibližně třetina vzorků tak zůstává nepoužita (Out-of-Bag vzorky), a právě tyto vzorky se používají k validaci stromu.

Rozdělení vstupních dat na trénovací a testovací odpovídá popsanému způsobu použitému v publikovaném řešení. Pro první 2 scénáře byl jako tréninkový dataset zvolen D1 z důvodů uvedených v 3.2.1. Během prvního scénáře byl model trénován a ověřován na vzorcích pocházejících pouze z 1 místnosti, aby došlo k co nejpřesnější replikaci podmínek, na kterých byl model testován v rámci publikace. V rámci vlastního ověřování bylo stejně jako v publikaci experimentováno i s počtem charakteristik, které jsou modelu předloženy, a výsledky korespondují s výsledky experimentu provedeného autory – při poskytnutí 3 došlo k mírnému poklesu výkonnosti modelu. Výsledky ověřování publikovaného řešení s využitím všech 4 charakteristik jsou prezentovány v tabulce 3.8.

Při tréninku na vzorcích pocházejících z 1 místnosti datasetu D1 bylo dosaženo klasifikační přesnosti 74,13%. Téměř 25% rozdíl je pravděpodobně způsoben zejména původem původních vstupních dat, které byly získány měřením na zařízení používající šířku pásma pouze 25 MHz.

Výsledkem druhého scénáře je, že by případný model s pevnými váhami použitý pro klasifikaci jiných než trénovacích dat nebyl o příliš lepší než náhodný výběr. Příčina vyplývá z principu použitého algoritmu, který se učí rozdělení extrahovaných rysů z impulsní odezvy trénovacích dat. Z povahy algoritmu lze tedy očekávat, že

Původní dataset: D1		Publikovaná				Dosažená		
Testovací Přesnost [%]		97,30				74,13		
		Ověřováno na						
Trénováno na		D1	D2	D3	D4	D5	D6	D7
D1	Přesnost [%]		55,70	56,68	56,61	78,09	41,87	49,13
	Preciznost [%]		88,64	55,51	70,49	87,44	–	36,64
	Citlivost [%]	–	48,58	63,83	10,64	79,09	–	97,78
	F1 skóre [%]		62,76	59,38	18,49	83,06	–	53,30
DO	Test. Přesnost [%]	76,28	73,57	82,99	80,46	87,03		76,54
	Preciznost [%]	82,84	92,99	85,93	79,71	94,29	–	57,95
	Citlivost [%]	66,29	70,96	78,58	77,46	86,11		76,65
	F1 skóre [%]	73,65	80,50	82,09	78,57	90,01		66,00
D ALL - DO	Přesnost [%]	63,88	65,83	64,86	60,45	78,60	41,74	56,33
	Preciznost [%]	65,06	88,37	64,03	58,52	86,76	–	38,59
	Citlivost [%]	59,96	63,95	66,52	49,73	80,81	–	79,63
	F1 skóre [%]	62,41	74,20	65,25	53,77	83,68	–	51,98

Tab. 3.8: Výsledky ověřování publikovaného řešení *RF* ([63]) ve scénářích popsanych v 3.2.1.

tento model bude vhodný zejména pro statické scénáře. Výsledky se ve 2. scénáři vyjímají pouze pro vzorky datasetu D5, kdy model zachycuje velkou část všech NLoS vzorků a činí nízký počet falešně pozitivních chyb, přestože oba datasety byly naměřeny v jiném prostředí a na jiném kanálu.

Při ověřování schopností modelu tréninkem a ověřováním na stejném datasetu model dosahuje napříč všemi datasety řádově podobných výsledků. Silnou stránkou modelu je zejména nízký počet falešně pozitivních detekcí, který je nejnižší pro vzorky datasetů D2 a D5. Problémy modelu představuje zejména citlivost. Nejlepších výsledků, stejně jako ve 2. scénáři, dosahuje model při klasifikaci vzorků datasetu D5.

Při tréninku na širší datové sadě vykazuje model ve všech případech shodné nebo lepší výsledky než při tréninku na datasetu D1. V případě navýšení schopností modelu lze pozorovat nárůst zejména již zmiňované citlivosti modelu. Zachování či navýšení schopností modelu je důsledkem tréninku na mnohonásobně větším počtu extrahovaných rysů, které odrážejí charakteristiky různých prostředí, kanálů a zařízení.

Publikované řešení je klasifikováno jako úspěšně ověřené, přestože se výkonnost modelu v žádném ze scénářů neblíží publikovaným hodnotám. Model však dosahuje konzistentních výsledků a v rámci publikace byl testován na datech, která nepocházejí z UWB zařízení. Vlastní implementace modelu nese v rámci souboru s imple-

mentovanými řešeními označení *RF*. Vzhledem k povaze algoritmu nelze objektivně určit parametry popsané v 3 bez vlastního tréninku algoritmu a jejich počet se může lišit v závislosti na rozložení poskytnutých vstupních dat určených k tréninku modelu. U algoritmů tohoto typu se také místo počtu parametrů a FLOPs používají parametry jako počet uzlů a hloubka stromu. S využitím modelu natrénovaném na datech datasetu D1 byly zjištěny hodnoty následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Maximální hloubka stromu: 42.
- Počet uzlů stromu: 36 998.
- Velikost modelu: 2 898,07 kB.
- Špičková paměťová náročnost metodou PSUtil: 5,73 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 3 156,40 kB.
- Průměrný čas klasifikace 1 vzorku: 1,787 ms.
- Optimální velikost dávky: 2048.
- Propustnost s velikostí dávky 4: 2 310 S/s.
- Propustnost s velikostí dávky 16: 9 321 S/s.
- Propustnost s velikostí dávky 64: 39 394 S/s.
- Propustnost s optimální velikostí dávky: 681 874 S/s.

Z uvedených parametrů vyplývají hlavní přednosti tohoto řešení – velmi nízká paměťová i výpočetní náročnost. Velmi rychlý inferenční čas a malá paměťová náročnost modelu umožňují jeho nasazení i na zařízeních s omezenými zdroji. V inferenčním času není započten čas nezbytný pro extrakci charakteristik ze vstupních dat, ale lze odhadovat, že i s extrakcí rysů by na tomto hardwaru klasifikace 1 vzorku trvala méně než 5 ms. Největší výhodou tohoto řešení je tedy jeho nízká paměťová i výpočetní náročnost, nízký inferenční čas a vysoká propustnost. Vstupní data lze extrahovat z impulsní odezvy kanálu bez znalostí dalších parametrů, což činí řešení snadno implementovatelným a přenositelným. Nevýhodou algoritmu je průměrná klasifikační přesnost a přímá závislost struktury jednotlivých stromů na vstupních datech.

3.2.8 Porovnání ověřovaných řešení

V rámci předchozích kapitol bylo detailně popsáno a ověřeno 6 publikovaných řešení. V této kapitole budou shrnuty výsledky ověřování a jednotlivá řešení budou porovnána na stejném scénáři.

Pro závěrečné porovnání byl využit upravený scénář prezentovaný v rámci publikace [12]. Aby byla zjištěna schopnost modelu detekovat vzorky pocházející z jiného vnitřního prostředí, bude provedeno ověřování s využitím následujícího scénáře:

Ověřováno na datech místnosti č.		Ověřované řešení					
		CNN LSTM 3.2.2	Novel CNN 3.2.3	Resnet 3.2.4	MLP TL 3.2.5	SVM LDA 3.2.6	RF 3.2.7
5	Přesnost [%]	82,95	84,15	81,55	83,80	74,93	75,90
	Preciznost [%]	90,77	86,03	86,19	89,56	94,63	80,78
	Citlivost [%]	72,94	81,13	74,65	76,09	52,28	67,27
	F1 skóre [%]	80,88	83,50	80,01	82,28	67,35	73,41
6	Přesnost [%]	83,17	84,12	81,05	83,67	74,10	75,77
	Preciznost [%]	90,36	86,16	85,11	88,78	95,03	80,79
	Citlivost [%]	74,08	81,11	75,05	76,92	50,60	67,31
	F1 skóre [%]	81,41	83,56	79,77	82,42	66,04	73,44
7	Přesnost [%]	83,27	84,23	82,12	84,39	75,15	76,27
	Preciznost [%]	90,79	86,76	86,38	87,66	95,24	81,73
	Citlivost [%]	74,66	81,41	76,95	80,64	53,80	68,66
	F1 skóre [%]	81,94	84,00	81,39	84,00	68,76	74,63
5, 6, 7	Přesnost [%]	83,18	84,17	81,45	84,14	74,76	75,98
	Preciznost [%]	90,57	86,28	85,78	90,97	95,03	81,11
	Citlivost [%]	74,04	81,26	75,42	75,81	52,26	67,76
	F1 skóre [%]	81,50	83,70	80,26	82,70	67,44	73,83
Trénovací čas [s]		80,20	15,25	216,23	7,77 + 0,93/1,72 TL	0,24	0,24 + 0,89 FE
Velikost natrénovaného modelu [kB]		115	5111	1930	658	0,55	0,64

Tab. 3.9: Výsledky ověřování 6 publikovaných řešení ve scénáři popsaném v 3.2.8.

- Každý model bude natrénovaný na vzorcích ze 4 prostředí (2x kancelář, malý byt, malá dílna) datasetu D1.
- Přesnost modelu bude ověřována na vzorcích 5., 6. a 7. prostředí, a zároveň na vzorcích ze všech těchto 3 (kuchyň s obývacím pokojem, ložnice, kotelna) prostředí.
 - Pro řešení využívající přenesené učení bude pro dotrénování modelu využito 100 LoS a 100 NLoS vzorků z jednotlivých prostředí.
- Trénink a následné ověřování proběhne pro každý model 10x, prezentovány budou průměrné výsledky.
- Pro možnosti širšího porovnávání jednotlivých řešení bude měřen i čas tréninku jednotlivých modelů, a zároveň bude uvedena i velikost výsledného naučeného modelu. Modely budou trénovány na stolním počítači s procesorem Intel Core i5 750, grafickou kartou Nvidia GeForce GTX1070 s 8 GB paměti a 24 GB operační paměti DDR4.

Výsledky ověřování výše popsaným scénářem jsou uvedeny v tabulce 3.9. Pro účely hodnocení lze ověřované metody rozdělit do 3 kategorií:

1. Kategorie představuje metody hlubokého učení, do kterých lze zahrnout metody *CNN LSTM*, *Novel CNN* a *Resnet*. Tyto metody využívají architektury hlubokých neuronových sítí.

Nejen v této kategorii vykazuje nejvyšší přesnost model *Novel CNN*. Mezi ře-

šeními v této kategorii skvěle kombinuje nižší výpočetní náročnost s vysokou přesností klasifikace, a zároveň nejlépe balancuje mezi přesností a citlivostí modelu. Klasifikační čas pod 10 ms a vysoká přesnost klasifikace dokazují vysokou efektivitu tohoto řešení. Ze všech ověřovaných řešení v této kapitole je však velikost modelu nejvyšší.

Řešení s označením *CNN LSTM* vykazuje na srovnávacím scénáři mírně nižší přesnost klasifikace, ale rozdíly jsou v porovnání s předchozím modelem minimální. Model činí velmi malé množství falešně pozitivních chyb, ale zachytí přibližně pouze 75 % všech NLoS vzorků. Přestože je výsledný model malý, výpočetní náročnost tohoto řešení je vysoká.

Nejméně přesným modelem z této kategorie je *Resnet*, který v porovnání s nejlepším řešením ztrácí až 3 % v klasifikační přesnosti. Metriky preciznosti jsou srovnatelné s nejlepším modelem v této kategorii, ale metriky citlivosti naopak s řešením *CNN LSTM*. Hluboká architektura činí model výpočetně velmi náročným, kdy trénovací čas tohoto modelu je mnohonásobně vyšší než pro ostatní řešení. Nejvyšší z množiny ověřovaných řešení v této kapitole je i inferenční čas, který přesahuje 30 ms, a zároveň má model nejnižší propustnost, která nedosahuje ani 500 S/s. Model je vhodný k nasazení pouze na centrální klasifikační prvek s vysokým výkonem.

2. Samostatnou kategorií je model *MLP TL*, který využívá přeneseného učení. V rámci srovnávacího scénáře dosahuje velmi podobné přesnosti jako *Novel CNN* při téměř polovičním čase potřebném k tréninku modelu a výrazně menší paměťové náročnosti. Přístup s využitím přeneseného učení může být velmi výkonný v situaci, kdy jsou k dispozici, případně kdy je možné získat malé množství dat z prostředí, ve kterém je UWB systém provozován. Toto tvrzení lze podložit porovnáním výsledků tohoto řešení s výsledky ověřovaných řešení v předchozí kategorii ve scénáři 2, kdy je model trénován na datasetu D1 a ověřován na ostatních datasetech. V rámci přeneseného učení je model vždy dotrénován na konkrétním datasetu, díky čemuž dosahuje vyšší a konzistentnější přesnosti klasifikace oproti ostatním řešením.

Efekt přeneseného učení je pozorovatelný i při modelu natrénovaném na D1 a ověřovaném na D6. Zatímco u řešení z 1. skupiny přesnost klasifikace nepřekročí 16,29 %, v případě přeneseného učení je klasifikace v tomto případě bezchybná. Nicméně je nezbytné uvést, že dataset D6 obsahuje pouze LoS vzorky, které jsou při doučení modelu předloženy. V tabulce 3.9 je prezentován i čas nezbytný pro dotrénování modelu na celkem 200 (0,93 s), případně pro ověřování na 3 místnostech 600 (1,72 s) vzorcích. Model je ze skupiny řešení spadající pod neuronové sítě ověřovaných v rámci této kapitoly nejrychlejší s klasifikačním časem 2,45 ms a propustností přes 177 000 S/s.

3. Poslední samostatnou kategorií jsou tradiční algoritmy stojového učení, do kterých patří řešení *SVM LDA* a *RF*. Výhodou těchto řešení je, že mohou být natrénována s využitím i relativně malého počtu vstupních dat. V rámci tohoto scénáře dosahují mezi všemi ověřovanými řešeními nejhorších výsledků, byť při nejnižší výpočetní i paměťové náročnosti.

Přestože v průběhu ověřování *SVM LDA* bylo většinou dosaženo lepších výsledků při použití polynomiálního jádra 3. stupně, vyskytovaly se i případy, kdy lepší výsledky generoval model s kvadratickým jádrem. Volba jádra by tedy neměla být jednoznačná a mělo by jí předcházet ověření modelů s různými jádry na konkrétním scénáři. Zároveň tento model vykazuje v tomto scénáři nejmenší počet falešně pozitivních chyb, ale zachytí pouze 50 % všech NLoS vzorků. Parametry pro odvození výpočetní a paměťové náročnosti jsou závislé na konkrétních trénovacích datech. Pokud byly modelu předloženy jako trénovací data všechny vzorky datasetu D1, byla velikost modelu *SVM LDA* 101 kB a průměrný čas klasifikace 0,419 ms.

Model *RF* dosahuje přibližně o 1 % vyšší přesnosti, lépe balancuje mezi precizností a citlivostí modelu a jako jediné řešení využívá rysy extrahované z CIR. Pokud byly modelu předloženy jako trénovací data všechny vzorky datasetu D1, byla velikost tohoto modelu 3 156 kB a průměrný čas klasifikace 1,787 ms. Velikost obou modelů v tomto scénáři byla menší než 1 kB, zatímco velikost modelů natrénovaných na celém datasetu D1 je mnohonásobně větší, což potvrzuje přímou závislost struktury obou modelů na trénovacích datech. Obě řešení jsou vhodná k nasazení na zařízení s omezenými zdroji.

3.3 Další publikovaná řešení

Existuje celá řada dalších publikovaných řešení pro klasifikaci LoS/NLoS, jež lze obecně rozdělit do následujících kategorií ([64]):

- Statistická analýza – využívá rozdíly ve statistických vlastnostech odhadovaných vzdáleností mezi LoS a NLoS podmínkami. Využívají metriky jako je čas letu signálu (ToA), časový rozdíl přijetí signálu (TDoA), sílu přijatého signálu (RSSI) a jejich kombinace.
- Metody založené na mapách – jsou velmi vhodné ve statických a dobře zmapovaných vnitřních prostředích. K identifikaci NLoS podmínek využívají známé rozložení prostředí a pozice tagů. Nevýhodou těchto metod je nezbytnost rozsáhlého sběru dat potřebných pro vytvoření mapy a nemožnost identifikovat dynamické překážky. Příkladem takové metody je např. řešení publikované v [5], která zavádí tzv. virtuální setrvačný bod (IMU – Inertial Measurement

Unit), který je vytvořen na základě pohybového trendu tagu a dat z měřicí jednotky. IMU monitoruje pohyb a zrychlení pohybu, na základě čehož vytváří virtuální setrvačný bod, který odráží pohyb objektu. Data z UWB se poté kompenzují pomocí dat z virtuálního setrvačného bodu a dochází k účinné kompenzaci chyby NLOS a zlepšení přesnosti polohování.

- Metody využívající šíření signálu – využívají model šíření signálu nebo impulsní odezvu kanálu (CIR) k identifikaci přítomnosti nejkratší cesty nejčastěji pozorované v LoS podmínkách. Jsou často kombinovány s technikami strojového učení k extrakci rysů ze signálu.
- Strojové učení – využívají extrahované rysy z dat impulsní odezvy kanálu (CIR) a trénují modely pomocí označených datových sad k rozlišení mezi LoS a NLoS scénáři. Extrahované rysy mohou být vybrány přímo tím, kdo řešení vytváří, případně se k extrakci rysů velmi často využívají techniky hlubokého učení.

V rámci evaluace publikovaných metod klasifikace přímého a odraženého signálu byla studována celá řada metod. V této části práce budou představeny právě ty, které nebyly ověřeny v rámci kapitoly 3.2. Každá metoda přináší unikátní řešení přístupem k architektuře modelu a práci s daty, a právě míra popisu řešení může být překážkou pro jejich další studium a podrobnější ověřování.

3.3.1 Řešení využívající konvoluční vrstvy

Společným prvkem většiny publikovaných řešení je využití konvolučních vrstev v architektuře modelu. Konvoluční vrstvy jsou v těchto úlohách využívány zejména z důvodu schopnosti efektivně extrahovat prostorové nebo časové vzory ze vstupních dat, díky čemuž jsou schopny zachytit charakteristické rysy LoS a NLoS podmínek. Tato řešení budou představena v rámci této kapitoly.

CNN GAN

Publikace [65] navrhuje pro klasifikaci použití konvoluční neuronové sítě v kombinaci s generativní sítí (GAN), jejímž cílem je generování dalších dat pro trénink modelu. Architektura modelu je tvořena 2 konvolučními vrstvami (jádra o velikosti 2) následovanými 3 plně propojenými vrstvami (120, 80 a 40 neuronů) a u plně propojených vrstev je použita ReLu jako aktivační funkce. Vstup konvoluční neuronové sítě tvoří pro navržené řešení celkem 8 parametrů (například index první cesty, amplitudy 3 bodů za první cestou, standardní odchylka úrovně šumu, ...) získaných z registrů zařízení DWM 1000.

Pro ověření výkonnosti autoři naměřili malé množství dat nevyváženého charakteru na nspecifikovaném kanálu a porovnávali výkonnost modelu bez a při použití

GAN k dogenerování tréninkových dat pro vyvážení datasetu. Presentovaná přesnost navržené metody při použití GAN je 91,19%. Bez použití GAN je přesnost modelu téměř zachována a klesne pouze v řádu desetin procent, a to v důsledku oslabení schopností modelu pro klasifikaci NLoS vzorků. V nevyváženém datasetu převládají LoS vzorky téměř v poměru 4:1, po dogenerování dat se poměr zmenšuje a je pod hranicí poměru 2:1. Z důvodu chybějícího popisu některých parametrů nezbytných pro replikaci modelu nebylo možné určit hodnoty pro porovnání výpočetní náročnosti. Malý vstupní vektor (1x8), nízký počet konvolučních vrstev i relativně malý počet neuronů v plně propojených vrstvách však působí pozitivně v kontextu výpočetní i paměťové zátěže.

Výhodou modelu je jeho jednoduchost, která jej činí snadno implementovatelným, pravděpodobně nízká paměťová a výpočetní náročnost a očekávat lze i nízké klasifikační časy. Výhodou může být i povaha vstupních dat, kdy jsou všechna data ihned dostupná a nevyžadují další výpočty. Povaha vstupních dat je zároveň i nevýhodou tohoto řešení, neboť činí model závislý na konkrétní platformě, která dokáže vstupní data poskytnout, což může být zásadní problém pro nasazení na data pocházející z jiného zařízení.

CNN-GRU-SE

Řešení navržené v publikaci [35] ke konvoluční neuronové síti přidává jednotku rekurentní brány (GRU – Gated Recurrent Unit) a SE (Squeeze and Excitation – stlačení a vzrušení) bloky. GRU představuje zjednodušenou verzi architektury LSTM a slouží ke zvýšení efektivity tréninku při zachování paměťových schopností. Úkolem SE bloků je zvýraznit důležité a potlačit méně významné rysy. Architektura obsahuje 2 konvoluční vrstvy, 2 SE bloky a 2 GRU jednotky.

Vstupem modelu jsou CIR data o délce 150 ns po indexu první cesty. Navržené řešení bylo kromě simulovaných dat testováno také na datasetu 1, na kterém bylo dosaženo přesnosti 88,45%. Vlastní implementace není z důvodu chybějícího popisu některých částí architektury možná. Pro posouzení výpočetní a časové náročnosti však autoři prezentují počet FLOPs, který přesahuje 950 tisíc, a počet parametrů modelu, kterých je přes 190 tisíc. Výpočetně náročné řešení bylo trénováno na grafické kartě s 11 GB paměti.

Výhodou navrženého řešení je jeho robustnost, neboť dosahuje kvalitních výsledků i na datech z reálného prostředí. Díky GRU je dosaženo kompromisu mezi výkonností a efektivitou modelu, neboť GRU způsobuje nižší výpočetní zátěž systému. Povaha vstupních dat umožňuje snadnou přenositelnost modelu. Přestože omezená délka vstupních dat snižuje nároky na paměť a výpočetní výkon, z uvedených parametrů vyplývá, že nasazení modelu na platformy s omezenými zdroji může být pro

blematické. Komplexita architektury zvyšuje složitost implementačního a ladícího procesu, může způsobovat horší interpretovatelnost rozhodovacího procesu a prodlužuje klasifikační čas, který lze na základě klasifikačních časů dalších ověřovaných modelů odhadovat na desítky ms.

CNN s Fourierovou transformací

Konvoluční neuronovou síť využívá i řešení prezentované v [66]. Vstupní data tvoří CIR, která je pomocí krátkodobé Fourierovy transformace STFT (Short-Time Fourier Transform) transformována na spektrum, díky čemuž je celý problém převeden na problém rozpoznávání obrazu. Navržený model je složen ze 2 konvolučních a 2 plně propojených vrstev, jejichž struktura není v publikaci uvedena. Původ testovacích dat není v publikaci uveden. V rámci publikace je prováděna klasifikace do 4 tříd (LoS a 3 různé typy NLoS) a bylo dosaženo přesnosti 98,24 %.

Odhad paměťové i výpočetní náročnosti této architektury je velmi problematický. Pozitivním vlivem může být menší hloubka architektury, ale zásadní vliv tvoří parametry jednotlivých vrstev. Výpočetně náročné je i předzpracování dat, neboť složitost STFT je pro segment dat o délce N dle [67] $O(N \cdot \log N)$. Výhodou modelu je vysoká přesnost klasifikace a nezávislost na vstupních datech, která jej činí přenositelným i na data pocházející z jiného UWB zařízení. Problém klasifikace LoS a NLoS je převeden na problém klasifikace obrazu, přičemž transformace vstupních dat prodlužuje inferenční čas, snižuje propustnost modelu a zvyšuje nároky na použité fyzické zařízení.

CNN s Morletovou vlnkovou transformací

Řešení [68] využívá před předáním vstupních CIR dat Morletovu vlnkovou transformaci MWT (Morlet Wave Transform). Vstupní CIR data jsou ořezána na délku 512 bodů a následně transformována na černobílý obraz o rozměru 128x128. Obraz tvoří vstup navržené architektury, která se skládá ze 2 konvolučních vrstev (64 a 128 filtrů) a 2 plně propojených vrstev (každá 128 neuronů). Výstupní vrstva má 2 neurony a používá aktivační funkci softmax.

Výkonnost navrženého řešení byla testována několika scénáři. Na simulovaných datech z různých prostředí bylo dosaženo přesnosti 96–100 %. Pro testování na reálných datech byl využit dataset 6, na němž bylo dosaženo přesnosti 98,84 %. Posledním scénářem bylo trénování modelu na simulovaných datech 1 místnosti a ověřování na datech ostatních místností. Během tohoto scénáře se přesnost modelu propadla až na 69,23 % a byl učiněn závěr, že je tato architektura vhodná především na statické scénáře.

V rámci souboru s implementovanými řešeními nese architektura označení *MWT_CNN*. S využitím vlastní implementace byly zjištěny hodnoty následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Počet parametrů modelu: 13 874 946.
- Velikost modelu: 54 229,68 kB.
- FLOPs: 675 637 004.
- Špičková paměťová náročnost metodou PSUtil: 78,30 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 1 789,02 kB.
- Špičková paměťová náročnost metodou Tensorflow: 366 938,75 kB.
- Průměrný čas klasifikace 1 vzorku: 17,942 ms.
- Optimální velikost dávky: 64.
- Propustnost s velikostí dávky 4: 122 S/s.
- Propustnost s velikostí dávky 16: 140 S/s.
- Propustnost s optimální velikostí dávky: 157 S/s.

Pro ověření paměťové a časové náročnosti architektury byla kromě modelu implementována i Morletova vlnková transformace – s využitím knihovny *PyWavelets* ([69]) a s parametry uvedenými v rámci publikace. Transformace 1 náhodného vzorku datasetu 1 trvala 23,91 ms, transformace celého datasetu (42 000 vzorků) pak 1063,3 s.

Z uvedených parametrů vyplývá, že výpočetní i paměťová náročnost modelu je velmi vysoká. Při hledání optimální velikosti dávky došlo při dávce o velikosti 512 k problémům s požadavky na alokaci paměti, které převyšovaly možnosti testovacího prostředí. Do zjištěných hodnot klasifikačního času a propustnosti modelu není započítán čas nezbytný pro transformaci vstupních dat, která v případě samostatného vzorku převyšuje čas následné klasifikace navrženým modelem. Optimální velikost dávky i propustnost modelu jsou velmi nízké a při započítání času nezbytného pro transformaci vstupních dat lze hodnotu propustnosti očekávat výrazně nižší.

Výhodou modelu je vysoká klasifikační přesnost a jednoduchá implementace architektury samotného modelu. Povaha vstupních dat umožňuje práci s daty z různých UWB zařízení. Nevýhodou modelu je velmi vysoká výpočetní i paměťová náročnost, nízká propustnost a delší klasifikační čas. Předzpracování dat je další výpočetně náročnou operací, kterou je pro použitelnost modelu nezbytné implementovat, a která zároveň zvyšuje nároky na použité fyzické zařízení.

CNN LSTM

Kombinace konvoluční neuronové sítě s LSTM pro klasifikaci je představena i v publikaci [70]. Vstupní data architektury jsou tvořena CIR daty o délce 1016. Před předáním dat neuronové síti je pomocí reverzibilní transformace zvýšena hodnota

odstupu signálu od šumu SNR (Signal to Noise Ratio), díky čemuž je dle prezentovaných výsledků navýšena klasifikační schopnost představené architektury o 5 %. Architektura obsahuje 2 konvoluční vrstvy, jejichž detailní struktura není v rámci publikace představena. Data z konvolučních vrstev jsou předána LSTM části s 64 neurony následované 1 plně propojenou vrstvou s neznámým počtem neuronů.

Navržený model byl testován na datech datasetu 1, na kterých bez aplikace transformace pro zvýšení SNR dosahoval přesnosti 80,25 % a při aplikaci transformace 85,67 %.

Z důvodu chybějícího popisu některých parametrů nezbytných pro replikaci modelu nebylo možné určit hodnoty pro porovnání výpočetní náročnosti. Na základě výsledků dalších ověřovaných modelů lze jako výpočetně náročnější část architektury označit LSTM část. Přestože LSTM působí pozitivně na celkovou velikost modelu, bývá výpočetně mnohem náročnější než konvoluční vrstvy, jejichž výpočetní náročnost samozřejmě závisí na jejich parametrech. Klasifikační časy struktur s LSTM, prezentovaných v rámci této práce, se pohybují v řádu desítek ms. Negativní vliv na výpočetní i paměťovou náročnost má také dlouhý vektor vstupních dat.

Výhodou modelu je jeho vysoká přesnost při testování na reálných datech. Navržená metoda vykazuje vysoký nárůst přesnosti při využití reverzibilní transformace. Výhodou navrženého řešení je i povaha vstupních dat, která umožňuje snadnou přenositelnost modelu na jiná UWB zařízení. Nevýhodou modelu jsou předpokládané vyšší výpočetní i paměťové nároky a vyšší klasifikační čas.

CNN

Konvoluční síť využívá i řešení navržené v [34], které cílí na zlepšení lokalizace ve vnitřních prostorech, jež probíhá na zařízeních s omezenými výpočetními schopnostmi. Vstup konvoluční sítě je v tomto případě tvořen impulsní odezvou kanálu ořezanou na délku 152 bodů od indexu první cesty. Podrobná struktura architektury není v publikaci prezentována. Navržený model byl testován na datech datasetu 1, na kterých bylo dosaženo přesnosti 87,4 %.

Autoři publikace provedli testování výpočetní náročnosti navržené architektury na platformách s různými úrovněmi výkonu – od Raspberry Pi s 1 jádrovým procesorem ARM1176JZF-S až po výkonné počítače s 8 jádrovými procesory řady Intel i7. Nejméně výkonná platforma dosahovala propustnosti 212 S/s (velikost dávky 256), což je dle autorů dostatečná hodnota pro klasifikaci na koncovém zařízení. Nejvýkonnější platforma (CPU Intel i7-6700HQ) dosáhla propustnosti 50531 S/s (velikost dávky 2048). Platforma, která se nejvíce přibližuje hardwaru, na kterém jsou testovány modely v rámci této práce, je označena jako „WS1“ a model na ní dosahuje propustnosti 34933 S/s při dávce o velikosti 2048.

Z publikovaných výsledků vyplývá, že při návrhu modelu byl kladen důraz na jeho efektivitu, díky které je možné jeho reálné nasazení i na embedded zařízení. Na základě výsledků lze rovněž konstatovat, že výpočetní a paměťová náročnost modelu je velmi nízká, čehož bylo dosaženo omezenou délkou vstupních dat a kombinací malého počtu konvolučních vrstev a filtrů.

Výhodami modelu jsou jeho jednoduchost, nízká výpočetní i paměťová náročnost a zároveň vysoká přesnost klasifikace na datech z reálného prostředí při velmi nízkých klasifikačních časech. Nevýhoda modelu může spočívat v omezené schopnosti zachytit komplexnější vzory v datech z jiného prostředí způsobené ořezáním vstupních dat a nízkou hloubkou navržené architektury.

FCN, Encoder

Publikace [60] představuje kromě architektury Resnet, která byla ověřována v kapitole 3.2, také plně konvoluční síť (FCN) a model *Encoder* (kombinace konvolučních a plně propojených vrstev). FCN se od standardních konvolučních sítí liší tím, že neobsahuje žádné plně propojené vrstvy, a v tomto případě se skládá ze 3 konvolučních vrstev. *Encoder* je tvořen kombinací více konvolučních vrstev, 1 plně propojené vrstvy a mechanismu soustředění (attention mechanism), jehož úkolem je dynamické váhování významnosti jednotlivých časových kroků. Vstupními daty pro obě sítě jsou CIR data ořezaná na délku 170 bodů v okolí nejvyššího vrcholu. V publikaci bylo rovněž experimentováno s množstvím filtrů konvolučních vrstev. Při nižším počtu parametrů dosahovala vyšší přesnosti FCN, při řádově 10^4 parametrech se přesnosti vyrovnaly a od 10^5 dosahovala větší přesnosti zmíněná síť Resnet. Rozdíl mezi oběma architekturami však činil pouze 1 % – 93 % pro FCN a 94 % pro Resnet.

Navržené metody byly testovány na vlastních datech naměřených v simulované chodbě a v reálném průmyslovém skladu na neuvedeném kanálu s využitím modulu DWM 1000. Prvním scénářem byla klasifikace dat z chodby, na kterých bylo použitím metody *Encoder* dosaženo přesnosti 94 % a s metodou FCN přesnosti 93 %. Cílem druhého scénáře bylo otestovat schopnosti generalizace modelu na datech z průmyslového skladu, kdy byl model trénován pouze na datech z několika přijímačů a ověřován na datech z jiného přijímače. V tomto scénáři průměrná přesnost klasifikace pro FCN dosahuje 92 % a při použití modelu *Encoder* 91 %.

Architektura FCN je tvořena 3 konvolučními vrstvami se 128 filtry a jádry o velikosti 8, 5 a 3. Ve výstupní vrstvě jsou 2 neurony a aktivační funkce softmax. V rámci souboru s implementovanými řešeními nese implementace modelu FCN označení *Stahlke2020a* a byly zjištěny hodnoty následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Počet parametrů modelu: 134 274.

- Velikost modelu: 559,50 kB.
- FLOPs: 43 211 788.
- Špičková paměťová náročnost metodou PSUtil: 25,71 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 1 957,32 kB.
- Špičková paměťová náročnost metodou Tensorflow: 100 179,50 kB, což je 97,83 MB.
- Průměrný čas klasifikace 1 vzorku: 9,850 ms.
- Optimální velikost dávky: 2048.
- Propustnost s velikostí dávky 4: 406 S/s.
- Propustnost s velikostí dávky 16: 956 S/s.
- Propustnost s velikostí dávky 64: 1 705 S/s.
- Propustnost s optimální velikostí dávky: 2 136 S/s.

Model *Encoder* obsahuje 3 konvoluční vrstvy se 128 filtry a jádru o velikosti 5, 11 a 21. Z důvodu chybějícího popisu mechanismu pozornosti nebylo možné určit hodnoty pro porovnání výpočetní náročnosti. V porovnání s architekturou FCN obsahuje *Encoder* stejný počet konvolučních vrstev se stejným počtem filtrů a většími jádru, ale navíc obsahuje ještě mechanismus pozornosti a plně propojenou vrstvu. Větší velikost jader vede ze své podstaty ke zvýšení výpočetních nároků, neboť velikost jádra určuje rozměr matice, která se posouvá po vstupních datech a aplikuje operaci konvoluce. Výpočetní nároky zvyšuje i přítomnost plně propojené vrstvy a mechanismu složitosti. Na základě výše uvedených informací lze model *Encoder* prohlásit za výpočetně i paměťově náročnější než FCN.

Obě architektury pracují pouze s CIR daty, což umožňuje jejich přenositelnost napříč různými UWB zařízeními. Další výhodou obou architektur je vysoká klasifikační přesnost a v případě architektury FCN ještě snadná implementace. Rozdíl mezi architekturami spočívá v hlubší architektuře modelu *Encoder*, která přináší zanedbatelné zvýšení přesnosti při nezanedbatelném navýšení výpočetní i paměťové náročnosti. Klasifikační čas modelu FCN dosahuje téměř 10 ms a propustnost modelu je v porovnání s dalšími řešeními nízká. Velikost samotného modelu vysoká není, ale během svého běhu vyžaduje alokaci většího množství paměti, než většina popisovaných řešení, což je dáno vysokým počtem filtrů v konvolučních vrstvách. Nasazení architektury FCN na zařízení s omezeným výpočetním výkonem může být problematické, architektura *Encoder* je pro nasazení na taková zařízení nevhodná.

3.3.2 Řešení bez konvolučních vrstev

V této kapitole budou představeny modely, které pro klasifikaci nevyužívají konvoluční vrstvy. Jedná se o přístupy využívající plně propojené sítě, rekurentní architektury nebo tradiční metody strojového učení. Jedná se zejména o metody, které

jako vstupní data používají reprezentaci, která vyžaduje extrakci lokálních struktur. Častou motivací pro použití těchto alternativních metod je i minimalizace výpočetní složitosti.

MLP GAN

Publikace [65] navrhuje kromě již popsané metody *CNN GAN* také architekturu MLP (Multilayer Perceptron). Navržený klasifikátor je testován stejným přístupem jako popisované řešení, u kterého bylo při použití GAN dosaženo přesnosti 89,14 %. Na rozdíl od zmíněné CNN metody jsou u této architektury k dispozici všechny nezbytné informace pro implementaci modelu. Architektura obsahuje 2 skryté vrstvy s 16 neurony a použitá aktivační funkce je sigmoid. Vstup architektury je také shodný a tvoří jej celkem 8 parametrů (například index první cesty, amplitudy 3 bodů za první cestou, standardní odchylka úrovně šumu, ...) získaných z registrů zařízení DWM 1000. Obě metody byly testovány na stejných datech malého množství, nevyváženého charakteru a pocházejících z nespécifikovaného kanálu.

V rámci souboru s implementovanými řešeními nese vlastní implementace modelu označení *Zhao2022*. S využitím vlastní implementace byly zjištěny hodnoty následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Počet parametrů modelu: 433.
- Velikost modelu: 17,53 kB.
- FLOPs: 833.
- Špičková paměťová náročnost metodou PSUtil: 19,66 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 1 393,57 kB.
- Špičková paměťová náročnost metodou Tensorflow: 8,25 kB.
- Průměrný čas klasifikace 1 vzorku: 1,863 ms.
- Optimální velikost dávky: 2048.
- Propustnost s velikostí dávky 4: 1 922 S/s.
- Propustnost s velikostí dávky 16: 9 609 S/s.
- Propustnost s velikostí dávky 64: 38 698 S/s.
- Propustnost s optimální velikostí dávky: 942 726 S/s.

Z uvedených parametrů vyplývá, že výpočetní i paměťová náročnost modelu je velmi nízká a model je tak vhodný k nasazení na zařízení s omezeným výkonem. Jednoduchá architektura činí model snadno implementovatelným, interpretovatelným a provozuschopným na zařízeních s omezenými zdroji. Propustnost modelu je rovněž velmi vysoká a klasifikační čas se drží pod 2 ms. Nevýhodou modelu může být původ vstupních dat, který může být překážkou pro přenositelnost modelu na data pocházející z jiného UWB zařízení.

LSTM

Architekturu LSTM představuje publikace [71]. Vstupem modelu je vektor složený ze 4 charakteristik kanálu vypočítaných z CIR (RSS, RMS, šikmost a špičatost) a komplexní frekvenční odezvy kanálu (CSI – Channel State Information) reprezentované 64 vzorky reálné a 64 vzorky imaginární části. Vstupní vektor se tedy celkem skládá ze 132 prvků. Autoři využívají CSI získané vlastním měřením v konferenční místnosti na neznámém modulu a kanálu k výpočtu CIR pomocí inverzní diskrétní Fourierovy transformace (IDFT – Inverse Discrete Fourier Transform).

Model obsahuje pouze 1 vrstvu LSTM s 10 skrytými vrstvami a jeho publikovaná přesnost je 95,47%. V rámci souboru s implementovanými řešeními nese architektura označení *Chang2021*. S využitím vlastní implementace byly zjištěny hodnoty následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Počet parametrů modelu: 491.
- Velikost modelu: 16,08 kB.
- FLOPs: 21.
- Špičková paměťová náročnost metodou PSUtil: 28,23 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 3 267,37 kB.
- Špičková paměťová náročnost metodou Tensorflow: 57,75 kB.
- Průměrný čas klasifikace 1 vzorku: 167,370 ms.
- Optimální velikost dávky: 2048.
- Propustnost s velikostí dávky 4: 24 S/s.
- Propustnost s velikostí dávky 16: 96 S/s.
- Propustnost s velikostí dávky 64: 386 S/s.
- Propustnost s optimální velikostí dávky: 7 500 S/s.

Přestože zjištěný počet parametrů modelu, jeho velikost i počet FLOPs jsou nízké, inferenční čas přesahuje 167 ms a propustnost modelu při různých dávkách je v porovnání s jinými řešeními také velmi nízká. Paměťová náročnost během běhu je v porovnání s ostatními řešeními také velká. Výhodou modelu je vysoká přesnost, malý počet parametrů a jeho malá velikost. Charakter vstupních dat umožňuje přenositelnost modelu na data pocházející z jiného zařízení. Nevýhodou modelu je jeho vysoká výpočetní náročnost a rovněž vyšší paměťová náročnost během běhu. Inferenční čas i nízká propustnost limitují použití modelu pro klasifikaci v reálném čase. Model je použitelný pouze na zařízeních s vyšší úrovní výpočetního výkonu.

SVM

Tradičních metod strojového učení využívá řešení prezentované v [72], které využívá dvoustupňovou metodu identifikace a klasifikace založenou na dynamickém porovnávání prahových hodnot a kombinaci fuzzy logiky s SVM. Zároveň zavádí 3 nové

parametry získané z impulzní odezvy kanálu, které slouží jako vstup SVM. Jedná se o následující parametry:

- Falešné vrcholy (FCN – False Crests Number): Počet falešných vrcholů překračujících prahovou hodnotu.
- Chyba první cesty (FPE – First Path Error): Rozdíl mezi skutečnou první cestou a zařízením detekovanou první cestou.
- Chyba vzdálenosti první cesty (FDE – First Path Distance Error): Rozdíl mezi skutečnou vzdáleností a vzdáleností vypočítanou z úrovně výkonu první cesty.

Pro testování navržené architektury byla naměřena vlastní data v kancelářském a vícepodlažním komerčním prostředí s využitím modulu DWM 1000. Byla použita frekvence 3,5 GHz, což odpovídá kanálu 1. Na těchto datech dosahoval model přesnosti 93,27 %. V rámci publikace není uvedena struktura modelu nezbytná k jeho implementaci pro ověření výpočetní a paměťové náročnosti v rámci této práce. Výpočetní náročnost algoritmů SVM je však obecně spíše nižší, což dokládají i výsledky ověřování jiného SVM algoritmu v rámci kapitoly 3.2.6. Oproti zmíněnému algoritmu pracuje toto řešení pouze se 3 vstupními parametry, což snižuje výpočetní i paměťovou náročnost, negativně však působí výpočet fuzzy věrohodnosti (credibility evaluation).

Výhodou modelu je vysoká přesnost klasifikace a malý rozměr vstupních dat, který pozitivně působí na náročnost modelu. Vstupní data jsou získávána z CIR s použitím informace o indexu první cesty, což model činí snadno přenositelným i na jiná zařízení. Klasifikace je silně závislá na způsobu výpočtu vstupních parametrů modelu a při špatné implementaci může být negativně ovlivněna šumem ve vstupních datech. Další nevýhodou může představovat i náročnější ladění modelu a vyšší složitost implementace fuzzy logiky, která je v publikaci popsána pouze obecně.

DFNN

Architektura dopředné hluboké neuronové sítě (DFNN – Deep Feedforward Neural Network) [73] využívá pro klasifikaci dvojici parametrů získaných z impulzní odezvy kanálu – celkový výkon (TP – Total Power) a výkon první cesty (FPP – First Path Power). Po vstupní vrstvě následují 2 skryté vrstvy s 50 neurony a použitou aktivační funkcí sigmoid. Ve výstupní vrstvě byla jako aktivační funkce použita logaritmická softmax. Cílem autorů bylo vytvořit architekturu optimalizovanou pro jednoduché aplikace a provoz v reálném čase na procesorech s omezeným výpočetním výkonem. Pro testování navržené architektury byla naměřena vlastní data s využitím modulu DWM 1000 na kanálu 5. Navržený model dosáhl přesnosti 98,60 %.

Vlastní implementace modelu nese v rámci souboru s implementovanými řešeními označení *Cwalina2019*. S využitím vlastní implementace byly zjištěny hodnoty

následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Počet parametrů modelu: 2 802.
- Velikost modelu: 28,09 kB.
- FLOPs: 5 512.
- Špičková paměťová náročnost metodou PSUtil: 19,35 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 1 415,74 kB.
- Špičková paměťová náročnost metodou Tensorflow: 34,00 kB.
- Průměrný čas klasifikace 1 vzorku: 2,112 ms.
- Optimální velikost dávky: 2048.
- Propustnost s velikostí dávky 4: 2 738 S/s.
- Propustnost s velikostí dávky 16: 10 243 S/s.
- Propustnost s velikostí dávky 64: 34 274 S/s.
- Propustnost s optimální velikostí dávky: 661 904 S/s.

Na základě uvedených parametrů lze paměťovou a výpočetní náročnost stanovit jako nízkou. Klasifikační čas mírně nad 2 ms a vysoká propustnost činí model vhodný k nasazení i na zařízení s omezeným výkonem. Jednoduchá architektura umožňuje snadnou implementaci, ladění a interpretovatelnost. Vstupní data jsou při znalosti indexu první cesty získatelná z impulsní odezvy kanálu, což činí model použitelný i na datech z jiného UWB zařízení. Pouze 2 extrahované charakteristiky v kombinaci s velmi vysokou přesností klasifikace mohou značit overfitting modelu na konkrétní data a problematickou klasifikaci vzorků z jiného prostředí, u nichž může model vykazovat problém se zachycením komplexních vzorů pomocí pouze 2 parametrů.

RFC

Publikace [74] představuje a experimentuje s 14 příznaky, které mohou být použity pro vylepšení klasifikačních schopností stávajících algoritmů. Většina z nich pochází z registrů DWM 1000 (10) a zbylé (4) jsou dopočítány. Nejlepších výsledků je dosaženo při použití 4 příznaků a algoritmu RFC (Random Forest Classifier), který v tomto případě používá pro klasifikaci 30 rozhodovacích stromů. Použitými příznaky jsou index první cesty, odhadnutá síla přijatého impulsu první cesty, rozdíl mezi odhadnutým RSS a silou impulsu první cesty a amplituda 2. bodu po první cestě.

Metoda byla testována pomocí vlastního datasetu naměřeného v prostoru chodby na kanálu 4 s využitím zařízení DWM 1000. Publikovaná přesnost klasifikace pro algoritmus RFC na tomto datasetu je 97 %.

Na základě publikovaných parametrů algoritmu RFC lze provést jeho implementaci, ale vzhledem k povaze algoritmu nelze stanovit počet parametrů ani počet FLOPs. U algoritmů tohoto typu se místo zmíněných metrik používají parametry

jako počet uzlů a hloubka stromu. Tyto parametry však nelze objektivně určit bez vlastního tréninku algoritmu a jejich počet se může lišit v závislosti na rozložení poskytnutých vstupních dat určených k tréninku modelu. Autoři uvádějí, že klasifikace 1 vzorku uvedeným algoritmem na notebooku s neuvedenými parametry trvá méně než 1 μ s.

Jednoznačnou výhodou modelu je jeho vysoká přesnost. Na základě prezentovaného času inference lze zmínit i nízkou výpočetní náročnost. Paměťová náročnost závisí kromě počtu stromů i na jejich hloubce a počtu uzlů, lze ale očekávat, že paměťová náročnost bude nízká. Navržený algoritmus je také snadno implementovatelný a lehce interpretovatelný. Nevýhoda modelu vyplývá z povahy algoritmu, kdy je struktura jednotlivých stromů přímo závislá na charakteristikách trénovacích dat a model může vykazovat výrazně nižší výkonnost při použití na jiných datech.

MLP

Publikace [75] navrhuje klasifikátor MLP se 3 skrytými vrstvami. Vstupní data jsou tvořena 3 hodnotami z registrů DWM 1000, které určují kvalitu přijatého rámce. Jedná se o standardní odchylku šumu odhadu impulsní odezvy kanálu (Standard Deviation of Channel Impulse Response Estimate Noise), odhad přijímané úrovně výkonu kanálu (Received Channel Power Estimate) a odhad výkonu detekované první cesty (First Path Received Channel Power Estimate). V rámci publikace byly prováděny experimenty se strukturou navrženého MLP klasifikátoru a nejlepšími výsledky bylo dosaženo pro 3 skryté vrstvy s 5, 12 a 3 neurony.

Klasifikátor byl testován na vlastním datasetu naměřeném v kancelářském prostředí. Naměřeno bylo 3600 vzorků na kanálu 2 s využitím DWM 1000. Publikovaná přesnost klasifikace na těchto datech je 89,93 %.

Publikace neobsahuje informace o použitých aktivačních funkcích. Vlastní implementace modelu, při které byly jako aktivační funkce skrytých vrstev použity ReLu, nese v rámci souboru s implementovanými řešeními označení *Bregar2016*. S využitím vlastní implementace byly zjištěny hodnoty následujících parametrů pro odvození výpočetní a paměťové náročnosti modelu:

- Počet parametrů modelu: 135.
- Velikost modelu: 20,21 kB.
- FLOPs: 249.
- Špičková paměťová náročnost metodou PSUtil: 19,29 MB.
- Špičková paměťová náročnost metodou Tracemalloc: 1 416,18 kB.
- Špičková paměťová náročnost metodou Tensorflow: 8,00 kB.
- Průměrný čas klasifikace 1 vzorku: 2,334 ms.
- Optimální velikost dávky: 2048.

- Propustnost s velikostí dávky 4: 1 175 S/s.
- Propustnost s velikostí dávky 16: 7 678 S/s.
- Propustnost s velikostí dávky 64: 21 049 S/s.
- Propustnost s optimální velikostí dávky: 702 727 S/s.

Z uvedených parametrů vyplývá, že výpočetní i paměťová náročnost modelu je velmi nízká a model je tak vhodný k nasazení na zařízení s omezeným výkonem. Výhodami modelu je kromě nízké náročnosti vysoká propustnost, i přestože klasifikace 1 vzorku při jednotkové velikosti dávky přesahuje 2 ms. Navržený model s přesností dosahující téměř 90 % je tedy velmi efektivní. Krátký vektor vstupních dat urychluje klasifikaci a snižuje nároky na fyzická zařízení, ale původ vstupních dat může být překážkou pro přenositelnost modelu na data pocházející z jiného UWB zařízení. Publikovaná přesnost modelu může být zkreslující, neboť byl model testován na datasetu poměrně malého rozsahu.

SVC

Publikace [43] pro klasifikaci navrhuje klasifikátor založený na SVM označený jako SVC (Support Vector Classifier). Klasifikace probíhá na základě vyhodnocení 9 různých příznaků, z nichž jsou některé extrahovány z CIR a zbylé pocházejí z registrů zařízení DWM 1000. Volba jádra navrženého klasifikátoru není explicitně uvedena, ale pravděpodobně se jedná o jádro s funkcí radiální báze RBF (Radial Basis Function), které dokáže pracovat s lineárně neseparovatelnými daty.

Navržené řešení bylo testováno s využitím vlastního datasetu pocházejícího z různých prostor univerzity, který byl autory zveřejněn a v rámci kapitoly 3.1 je popsán jako dataset 3 (DWM 1001, kanál 5). Publikovaná přesnost klasifikace na datech tohoto datasetu je 92 %.

Získání vypovídajících hodnot parametrů pro stanovení výpočetní a paměťové náročnosti modelu není možné, i přesto že by na základě publikované struktury modelu byla možná jeho implementace při použití RBF jádra. Podstata problému spočívá v principu SVM, kdy je struktura konečného modelu (počet podpurných vektorů) závislá na charakteru trénovacích dat. Metriky pro stanovení výpočetní a paměťové náročnosti jsou úzce spjaty s trénovacími daty a pro jiná trénovací data se mohou lišit. Z toho důvodu není možné tyto parametry objektivně stanovit. Navržený model trénovaný na uvedeném datasetu obsahoval 700 podpurných vektorů.

Výpočetní a paměťová náročnost algoritmů SVM je obvykle nižší, stejně jako inferenční čas. Nevýhodou řešení tohoto typu je nejednoznačná struktura modelu, která je přímo závislá na konkrétních trénovacích datech. Nasazení modelu musí tedy vždy předcházet vyhodnocení jeho náročnosti na konkrétních datech. Překážkou pro nasazení může být i povaha vstupních dat, kdy použité parametry z registrů

DWM 1000 nemusí být k dispozici při nasazení na data pocházející z jiného zařízení. Výhodou tohoto řešení je vysoká klasifikační přesnost při testování na reálných datech a očekávaná nižší paměťová i výpočetní náročnost v porovnání s hlubokými neuronovými sítěmi.

LogReg, RBFSVC

Publikace [40] se nezabývá přímo návrhem modelu pro klasifikaci, nýbrž návrhem vstupních příznaků extrahovaných z CIR. Autoři se zabývají problematikou vysoké výpočetní náročnosti stávajících klasifikátorů a navrhují 29 příznaků, které lze extrahovat z CIR. Pro validaci svého řešení používají různé obecné modely, jejichž struktura není v publikaci blíže specifikována. Tyto modely testují na 5 různých datasetech, včetně datasetů 1, 2, 3 a vlastního datasetu, který byl naměřen na UWB kanálu 9. Vlastní dataset obsahuje přes 95 tisíc vzorků z reálného industriálního prostředí a je vyvážený. K měření byl použit modul NXP SR150.

Klasifikátory byly podle své paměťové velikosti rozděleny do 2 skupin na malé a velké ($> 2\text{ kB}$). Následně bylo provedeno 6 scénářů, ve kterých byly jako vstupní data použity různé příznaky extrahované z různých délek okna CIR. Napříč těmito scénáři dosahoval nejlepších výsledků pro dataset naměřený autory této publikace klasifikátor LogReg (Logistic Regression – logistická regrese) a ze skupiny velkých modelů klasifikátor RBFSVC (SVM s RBF jádrem). Zásadním závěrem je, že se nejlepší kombinace použitých příznaků a délky CIR okna liší napříč testovanými datasety.

Klasifikátor LogReg dosahoval napříč scénáři přesnosti 89,65 – 92,80 %. Velikost tohoto klasifikátoru je pro scénář s nejvyšší přesností pouze 0,92 kB a nejlepší přesnosti bylo dosaženo při použití 11–12 příznaků při oknu o délce 140 ns. Nízkou výpočetní náročnost modelu potvrzuje klasifikační čas 882 ns a čas nezbytný pro extrakci příznaků 8,33 ms (na mikrokontroléru ESP8266 80 MHz). Druhý klasifikátor RBFSVC dosahoval napříč scénáři přesnosti 91,87–93,56 %. Nejlepší přesnosti bylo dosaženo při použití 8 příznaků, velikost modelu je v tomto případě 0,95 MB a klasifikační čas pouze 638 μs . Extrakce příznaků z 1 CIR vzorku však trvala 27,7 ms.

Výhodou popsaného přístupu je vysoká efektivita modelů – vysoká klasifikační přesnost a nízká výpočetní i paměťová náročnost. Vysoké množství příznaků umožňuje zachytit komplexní rysy vstupních dat. Významnou nevýhodou přístupu je nutnost realizace experimentů s použitou sadou příznaků a délkou CIR okna. Extrakce příznaků zároveň prodlužuje klasifikační čas a zvyšuje nároky na systémy, na kterých je model nasazen. Tento přístup je vhodný pro statické scénáře, neboť model natrénovaný na určité sadě příznaků nemůže být použit pro klasifikaci jiné sady příznaků, která je vhodná pro konečná data.

3.4 Shrnutí

V této kapitole byly představeny a analyzovány publikované metody strojového učení určené pro klasifikaci LoS a NLoS signálu. První část je zaměřena na detailní ověření vybraných řešení s využitím dostupných datasetů, zatímco ve druhé části byla představena další relevantní řešení. U každého z těchto přístupů byla diskutována jejich výpočetní a paměťová náročnost, stejně jako hlavní výhody a nevýhody.

Komplexní přehled všech popisovaných metod je shrnut v tabulce 3.10, která obsahuje informace o typu řešení, dosažené klasifikační přesnosti a datech, na kterých jí bylo dosaženo. V případech, kdy to povaha a popis řešení umožňovaly, byla implementována samotná architektura modelu a s využitím náhodných vstupních dat byly stanoveny parametry sloužící k vzájemnému porovnání výpočetní a paměťové náročnosti jednotlivých metod. Výčet těchto parametrů je uveden v úvodu kapitoly 3 a zjištěné hodnoty jsou vždy součástí popisu konkrétního řešení. Pro přehlednost tabulka prezentuje pouze vybrané parametry. Některá publikovaná řešení, i přes neúplný popis modelu, uvádějí některé z těchto parametrů a pokud nejsou závislé na vstupních datech, jsou do tabulky rovněž zahrnuty.

Pro účely obecného hodnocení lze popisované metody rozdělit do dvou kategorií na základě charakteru použitých vstupních dat.

První kategorii tvoří **metody využívající průběh impulsní odezvy kanálu**. Řešení v této kategorii jsou nejvíce zastoupena hlubokými neuronovými sítěmi, ve kterých jsou nejvíce využívány konvoluční neuronové sítě. Tyto modely využívají automatickou extrakci charakteristik přímo ze signálu prostřednictvím hlubokých neuronových sítí. Není tak potřeba manuálně definovat rysy, které mají nebo mohou být extrahovány. V porovnání s druhou skupinou dosahují nižší klasifikační přesnosti. Jednoznačnou výhodou těchto řešení je přenositelnost. Díky tomu, že modely jako vstupní data používají CIR, jsou schopny klasifikovat data z různých zdrojů a odpadají problémy s chybějícími hodnotami pocházejícími např. z registrů UWB modulu. Další výhodou je, že po získání CIR lze přejít rovnou ke klasifikaci vzorku a není nutné provádět další výpočty a extrakci charakteristik. Protože se velmi často jedná o hluboké neuronové sítě, jsou tato řešení výpočetně i paměťově náročnější, což je činí méně vhodnými pro zařízení s omezeným výkonem.

Druhou skupinou jsou metody, které pracují s vektorem charakteristik získaných z CIR, hodnotami získanými z registrů UWB zařízení či kombinací obojího. V anglické literatuře by se tato řešení dala označit jako **feature-based** metody. Charakteristiky získané z CIR jsou často vypočítány právě s využitím různých hodnot získaných z registrů zařízení, nejčastěji např. indexu první cesty. Tato řešení častěji využívají jednoduché klasifikační algoritmy a jednoduché neuronové sítě. Jednoduchost těchto metod se pozitivně odráží na nízké výpočetní i paměťové náročnosti,

Typ (Označení), Publikace	Vstup	Dataset	Zařízení	Kanál	Přesnost [%]	Počet Parametrů	FLOPs	Velikost [kB]	Klasifikace [ms]	Propustnost [S/s]
CNN LSTM, [54]	CIR o délce 1016	D1 ([39])	DWM 1000	2	82,14	934 767	20 327	115	Na CPU nebylo možné určit	
CNN (Novel CNN), [55]	CIR o délce 1016	Vlastní	DWM 1000	–	92,67	1,30M	6,9M	5 111	9,326	5 613
Resnet, [60]	CIR o délce 170	Vlastní	DWM 1000	–	92,00	465 282	156,6M	1 930	30,047	471
MLP TL, [61]	CIR o délce 1016 + 1016	Vlastní	EKV1000	–	75,00/90,00	293 377	586 369	1 166	2,451	177 015
SVM LDA, [62]	CIR o délce 100	Vlastní	EVB1000	–	100,00				Závisí na trénovacích datech	
RF, [63]	4 vlastnosti z CIR	Vlastní	N210 (SDR)	–	97,30				Závisí na trénovacích datech	
CNN GAN, [65]	8 hodnot z registrů	Vlastní	DWM 1000	–	91,19				Chybějící informace o modelu	
CNN-GRU-SE, [35]	CIR o délce 150	D1 ([39])	DWM 1000	2	88,45	194 940	958 550		Chybějící informace o modelu	
CNN, [66]	CIR	Vlastní	–	–	98,24				Chybějící informace o modelu	
CNN, [68]	CIR o délce 512	D6 ([51])	Pozyx	5	98,84	13,87M	675,6M	54 230	17,942	157
CNN LSTM, [70]	CIR o délce 1016	D1 ([39])	DWM 1000	2	85,67				Chybějící informace o modelu	
CNN, [34]	CIR o délce 152	D1 ([39])	DWM 1000	2	87,4				Chybějící informace o modelu	
FCN, [60]	CIR o délce 170	Vlastní	DWM 1000	–	92,00	134 274	43,2M	560	9,850	2 136
CNN (Encoder), [60]	CIR o délce 170	Vlastní	DWM 1000	–	91,00				Chybějící informace o modelu	
MLP GAN, [65]	8 hodnot z registrů	Vlastní	DWM 1000	–	89,14	433	833	18	1,863	924 726
LSTM, [71]	4 vlastnosti z CIR, CSI (64, 64)	Vlastní	–	–	95,47	491	21	16	167,370	7 500
SVM, [72]	3 vlastnosti z CIR	Vlastní	DWM 1000	1	93,27				Závisí na trénovacích datech, chybějící informace o modelu	
DFNN, [73]	2 vlastnosti z CIR	Vlastní	DWM 1000	5	98,60	2 802	5 512	28	2,112	661 904
RF (RFC), [74]	4 hodnoty z registrů	Vlastní	DWM 1000	4	97,00				Závisí na trénovacích datech	
MLP, [75]	3 hodnoty z registrů	Vlastní	DWM 1000	2	89,93	135	249	20	2,334	702 727
SVM (SVC), [43]	9 vlastností a hodnot z CIR a registrů	D3 ([42])	DWM 1001	5	92,00				Závisí na trénovacích datech	
LogReg, [40]	11–12 vlastností z CIR	Vlastní	NXP SR150	9	92,80				Závisí na trénovacích datech, chybějící informace o modelu	
SVM (RBFSVC), [40]	8 vlastností z CIR	Vlastní	NXP SR150	9	93,56				Závisí na trénovacích datech, chybějící informace o modelu	

Tab. 3.10: Přehled publikovaných řešení popisovaných v rámci této práce.

díky čemuž jsou vhodnější pro nasazení na zařízení s omezenými zdroji. Metody jsou díky nízkým klasifikačním časům vhodné přímo na koncová zařízení k provádění klasifikace v reálném čase. Nevýhodou těchto metod je omezená přenositelnost, kdy například na UWB zařízení jiného výrobce nemusí být k dispozici hodnoty, které řešení využívá. Výkonnost těchto metod je silně ovlivněna výběrem konkrétních příznaků, což tvoří další překážku přenositelnosti. Publikované výsledky dokazují, že při existenci sady příznaků a dat různého původu dosahuje model na různých datech nejvyšší přesnosti s využitím různých příznaků. Výběr konkrétního souboru příznaků by tedy v ideálním případě měl předcházet nasazení modelu.

Z hlediska výpočetní náročnosti jsou feature-based modely jednoznačně méně náročné a lépe použitelné v systémech s omezenými zdroji. Zatímco metody využívající průběh CIR jsou univerzálnější, druhá skupina metod umožňuje modifikovatelnou adaptaci na cílové prostředí. Pokud bude řešení z první skupiny v cílovém prostředí vykazovat určité výsledky, lze výkon modelu modifikovat pouze zásahem do struktury modelu, modifikací tréninkového procesu a předzpracování dat, přičemž případná velikost změny výkonnosti modelu může být velmi omezená. Naproti tomu feature-based modely mohou díky změně definice vstupních příznaků dosáhnout lepší adaptace, což ale neznamená, že je automaticky zaručena lepší klasifikační přesnost než při použití modelu z 1. skupiny.

Z provedených experimentů v kapitolách 3.2.2– 3.2.8 vyplývá, že výkonnost kaž-

dého řešení závisí na konkrétních datech a napříč různými datasey se liší. Mezi datasey lze najít některé, na jejichž datech je klasifikační přesnost ověřovaných modelů častěji nižší než u ostatních datasetů (např. D1, D7). Součástí ověřování byl i zmíněný neveřejný dataset D7, který obsahuje vzorky UWB komunikace probíhající na kanálu 9 a na zařízení, na jehož datech nebylo testováno žádné z prezentovaných řešení. Výsledky klasifikace pro tento dataset dosahují řádově obdobných hodnot jako výsledky klasifikace pro jiné datasey. Vzorky z kanálu 9 byly rovněž využity pro testování feature-based řešení v publikaci [40]. Řešení testovaná v této publikaci na datech z kanálu 9 dosahují s využitím různých sad příznaků klasifikační přesnosti přes 90 %. Lze tedy konstatovat, že pro kanál 9 je dosahováno obdobných výsledků jako pro ostatní nižší kanály.

4 Stanovení vlivu frekvence

Tato kapitola se zabývá analýzou vlivu frekvence na přenositelnost modelů pro klasifikaci LoS a NLoS. V první části jsou popsána data, která budou využita pro scénáře popsané v následujících částech, jejichž cílem je popsat vliv volby UWB kanálu na schopnosti jednotlivých modelů. Závěrem této kapitoly je shrnutí závěrů vyplývajících z jednotlivých scénářů.

4.1 Metodologie stanovení vlivu frekvence

Pro účely stanovení vlivu frekvence jako parametru na přenositelnost modelů v rámci této práce bylo stanoveno následující kritérium pro použitá data – vliv frekvence na přenositelnost modelů lze posuzovat pouze s využitím dat, u nichž je vliv prostředí, hardwaru, způsobu měření a dalších blíže nespecifikovaných jevů s výjimkou frekvence zanedbatelný.

Primární zaměření většiny ověřovaných řešení v rámci této práce, ale také všech zmíněných řešení v rámci kapitoly 3.2, je vytvořit silný klasifikátor pro konkrétní typ vnitřního prostředí, případně klasifikátor silný nezávisle na typu vnitřního prostředí. Tomu odpovídá i povaha veřejně dostupných datasetů. Pro posouzení vlivu samotné frekvence na přenositelnost modelu je nezbytné vliv posuzovat na datech ze stejného vnitřního prostředí. Nesmí být opomenut ani vliv různých zařízení použitých k měření vzorků, jenž spočívá například v odlišných délkách a charakteristikách CIR a je popsán v [45].

Tabulka 3.1 prezentuje UWB kanály a původ dat jednotlivých datasetů popsaných v kapitole 3.1. Z dokumentace k datasetu 1 je možné určit, která část dat odpovídá konkrétnímu typu vnitřního prostředí, ale nelze určit, zda se jedná o stejné prostředí jako v případě dat z více kanálů 2. datasetu. U druhého datasetu pochází vzorky pro kanál 2 ze 2 sousedících kancelářských místností se spojovací chodbou a pro ostatní kanály (1, 3, 4, 5 a 7) z kancelářského prostředí s více místnostmi. U ostatních veřejně dostupných datasetů není znám klíč k přiřazení konkrétních vzorků ke konkrétnímu místu jejich původu.

Z uvedeného vyplývají jako vhodné pouze vzorky z datasetu 2 pro kanály 1, 3, 4, 5 a 7, které byly naměřeny na stejných místech, stejným způsobem a zařízením. Jedná se vždy o 3600 vzorků, z nichž je 3000 (83,33 %) NLoS. Právě tyto vzorky budou použity ke stanovení vlivu frekvence jako parametru na přenositelnost modelů. Tabulka 4.1 obsahuje přehled frekvencí a šířek pásma odpovídajících jednotlivým HRP kanálům z uvedeného datasetu dle IEEE 802.15.4z–2020.

V následujících kapitolách jsou prezentovány způsoby, pomocí nichž byl stanoven vliv frekvence jako parametru na přenositelnost modelu. V rámci všech scénářů

Kanál	1	3	4	5	7
Frekvence [MHz]	3494,4	4992,8	3993,6	6489,6	6489,6
Šířka pásma [MHz]	499,2	499,2	1331,2	499,2	1081,6

Tab. 4.1: Přehled frekvencí a šířek pásma odpovídajících jednotlivým HRP kanálům z použité části datasetu 2 dle IEEE 802.15.4z–2020.

jsou využívány implementované klasifikátory z části 3.2, které jako vstupní data používají CIR, ze kterého nejsou extrahovány žádné rysy (s výjimkou 3.2.7). Použitý dataset je nevyvážený, a proto jsou pro posuzování výkonnosti modelů na datech jednotlivých kanálů kromě přesnosti využívány také metriky vyvážené přesnosti, F1 skóre a ROC AUC. Zatímco vyvážená přesnost a ROC AUC lépe odráží obecné schopnosti modelu, přesnost a F1 skóre hodnotí model při konkrétním rozhodovacím prahu a přímo neřeší problém nevyváženosti dat. U modelů SVM LDA a RF jsou z důvodu jejich vnitřní struktury hodnoty vyvážené přesnosti a ROC AUC totožné, nicméně pro zachování jednotného formátu výstupů a přehlednost celého řešení jsou prezentovány obě metriky. Během tréninku modelu je vždy zajištěna vyváženost obou tříd pouze v trénovací sadě, rozložení tříd v testovací sadě odpovídá rozložení tříd napříč datasetem.

Součástí vybraných scénářů je i statistické hodnocení. Analyzovány jsou metriky každého z použitých modelů vzhledem ke kanálu, ke kterému náleží data použitá pro trénink. Statistické hodnocení probíhá odděleně pro každou z výkonnostních metrik a sestává se z následujících bodů:

1. Test normality rozdělení. Normalita rozdělení je předpokladem pro použití některých parametrických statistických testů, které předpokládají, že data pocházejí z určitého známého rozdělení a využívají informace o parametrech tohoto rozdělení. Pro každou skupinu metrik (např. soubor vyvážených přesností modelu Y trénovaném na kanálu X) je vyhodnocováno, zda jsou hodnoty metrik rozděleny normálově. Pro hodnocení normality rozdělení je použit Shapiro-Wilkův test, který je vhodný pro malé a střední soubory dat [77], [78]. Použita je standardní hodnota hladiny významnosti $\alpha = 0,05$.
2. Test homogenity rozptylu. Dalším předpokladem pro použití parametrických statistických testů je homogenita rozptylů mezi skupinami. Pro ověření homogenity rozptylu je použit Levenův test, neboť je robustní vůči odchylkám od normality [77], [79].
3. Volba testu pro zjištění statisticky významných rozdílů. Volba vhodného statistického testu závisí na výsledcích testu normality a homogenity rozptylu. Cílem testu je stanovit, zda existují významné odlišnosti mezi výkonnostními

metrikami napříč jednotlivými skupinami (kanály). V případě, že jsou splněny předpoklady normality a homogenity rozptylu, je použitým testem jednofaktrová analýza rozptylu (ANOVA – Analysis of Variance), která testuje nulovou hypotézu, že jsou průměry všech skupin stejné. Pokud předpoklady normality a homogenity rozptylu splněny nejsou, je použit neparametrický Kruskal-Wallisův test, který porovnává mediány skupin a vyhodnocuje, zda se skupiny statisticky významně neliší [77], [80].

4. Následná (Post-Hoc) analýza. Pokud provedený test popsáný v předchozím bodu odhalí statisticky významnou odlišnost mezi skupinami (odmítne nulovou hypotézu), je třeba dále určit konkrétní skupiny, mezi nimiž tento statisticky významný rozdíl existuje. K těmto úlohám slouží právě post-hoc testy, mezi které patří například:

- Tukeyho HSD test pro ANOVA [77], [81].
- Dunnův test s Bonferroniho korekcí pro Kruskal-Wallisův test [77], [82].

Cíle statistických testů mohou být dále upřesněny v rámci popisu konkrétního scénáře. Výsledky nejen statistických testů mohou být kromě tabulek prezentovány i s využitím následujících grafických výstupů ([77], [83], [84]) vytvořených využitím knihovny *seaborn* 0.13.2 [85] pro vizualizaci statistických dat s využitím programovacího jazyka python:

- **Sloupcové grafy** – umožňují porovnání mediánů různých skupin pomocí výšky dílčích sloupců. Vzhledem k tomu, že většina dat nespĺňuje předpoklady normality, jsou výsledky vždy prezentovány jako mediány a 95% percentilový interval. Pomocí tohoto grafického výstupu lze tedy porovnat mediány určité metriky pro všechny modely napříč kanály a sledovat rozsah hodnot, ve kterém leží většina dat. Označení souborů se sloupcovými grafy je v elektronické příloze převzato z anglického označení „Barplots“.
- **Krabicové grafy** – slouží k zobrazení rozdělení dat v různých skupinách a odhaluje jejich variabilitu, symetrii a indikuje přítomnost odlehých hodnot. Krabice představuje mezikvartilové rozpětí a čára uvnitř krabice hodnotu mediánu. Dále zobrazuje rozsah běžných hodnot a hodnoty mimo tento rozsah. Krabicové grafy jsou z důvodu snazší interpretovatelnosti a větší přehlednosti prezentovány v převrácené variantě, kdy na ose *y* jsou vyneseny frekvence s odpovídajícími kanály ve vzestupném pořadí (dle tabulky 4.1) a na ose *x* hodnoty konkrétní metriky. Označení souborů se sloupcovými grafy je v elektronické příloze převzato z anglického označení „Boxplots“.
- **Heatmapy** – reprezentace matice hodnot pomocí barevné škály. Díky tomu je možná rychlá identifikace vzorů, extrémů či podobností mezi hodnotami. Tato forma reprezentace je zvolena také pro prezentaci výsledků post-hoc testů, kdy

heatmapa vždy obsahuje informaci o typu provedeného post-hoc testu, z čehož lze zpětně dovodit, zda byl použit parametrický či neparametrický test.

- **Q-Q graf** – umožňuje posoudit, zda data sledují určité teoretické rozdělení (nejčastěji normálové). Pokud body leží přibližně na přímce, tak data odpovídají zvolenému rozdělení. Odchytky od přímky indikují odklon od teoretického rozdělení.
- **Violin graf** – zobrazuje medián, mezikvartilové rozpětí, ale i hustotu pravděpodobnosti formou tvaru a šířky rozdělení dat. Umožňuje tak porovnat rozdělení dat mezi skupinami.

V každé kapitole popisující konkrétní scénář je vždy uveden způsob a umístění všech výstupů, které mohou být z důvodu vysokého počtu výstupů součástí přílohy nebo elektronické přílohy této práce, a během hodnocení a diskuze nad výsledky jednotlivých scénářů na ně může být v textu odkazováno. V případě prezentace formou tabulky jsou hodnoty sledovaných výkonnostních metrik každého modelu na základě provedených statistických testů, které testují normalitu rozdělení a homogenitu rozptylu, prezentovány buď ve formátu „průměrná hodnota \pm směrodatná odchylka“ přes všechny běhy modelu pro normálově rozložená data splňující podmínku homogenity rozptylu, nebo ve formátu „medián (25. percentil — 75. percentil)“ pro data nesplňující alespoň jednu z těchto podmínek. Hodnoty metrik pro jednotlivé běhy jsou součástí elektronické přílohy – hodnoty metrik získané během tréninku modelu se nacházejí v souboru „Vysledky_train.csv“ a hodnoty metrik získaných během ověřování analogicky v souboru „Vysledky_eval.csv“.

4.2 Scénář 1 – Trénink a testování na stejném kanálu

V rámci prvního scénáře jsou modely trénovány a testovány na stejném kanálu. Cílem je zjistit výkon modelu zvlášť na každém kanálu. Tento výkon slouží rovněž jako referenční pro 2. a 3. scénář. Každý model je trénován a testován na 10 nezávislých náhodných stratifikovaných rozděleních dat (2 opakování pro každé z 5 náhodných rozdělení), čímž je zajištěna robustnost výsledků vůči náhodnosti.

Analyzovány jsou metriky každého z použitých modelů vzhledem ke kanálu, ke kterému náleží data použitá pro trénink i ověřování. Statistické testy testují nulovou hypotézu H_0 , že neexistuje statisticky významný rozdíl mezi výkonností modelu trénovaném na různých kanálech. Testovány jsou vždy hodnoty 1 ze sledovaných metrik napříč všemi kanály.

Získané hodnoty sledovaných metrik a grafické výstupy jsou prezentovány následujícím způsobem:

- Hodnoty sledovaných metrik modelu:
 - V tabulce 4.2 včetně výsledků statistických testů testujících H_0 .

- V souboru „scenar1.csv“, který je součástí elektronické přílohy, včetně výsledků statistických testů testujících H_0 .
- Krabicovým grafem pro metriku vyvážená přesnost prezentovaném na obrázku 4.1. Krabicové grafy pro všechny sledované metriky jsou součástí elektronické přílohy.
- Sloupcovým grafem pro metriku vyvážená přesnost prezentovaném na obrázku 4.2. Krabicové grafy pro všechny sledované metriky jsou součástí elektronické přílohy.
- Výsledky post-hoc testů:
 - Prostřednictvím heatmapy pro metriku vyvážená přesnost prezentované na obrázku 4.3. Heatmapy pro všechny sledované metriky jsou součástí elektronické přílohy.
- Umístění a označení souborů v elektronické příloze:
 - Soubory se nacházejí ve složce „\\Vliv frekvence\Scenar1“.
 - Soubory ve formátu „.webp“ s grafickými výstupy nesou vždy označení ve formátu „**SC_OT_M**“, kde:
 - * **SC** je vždy „Scenar1“.
 - * **OT** představuje typ výstupu, a může nabývat hodnot „Barplot“, „Boxplot“ a „Heatmap“.
 - * **M** představuje typ metriky, jejíž hodnoty grafický výstup prezentuje. Pro označení metrik jsou použity anglické ekvivalenty „Accuracy“ pro přesnost, „BalancedAccuracy“ pro vyváženou přesnost a pro zbývající metriky je použito „F1“ a „RocAuc“.

S využitím výše zmíněných výstupů bude v následujícím textu posouzen vliv frekvence na výkonnost jednotlivých modelů. Součástí tabulky 4.2 jsou rovněž výsledky popsaných statistických testů. Pokud test pro zjištění statisticky významných rozdílů indikuje statistické rozdíly mezi hodnotami konkrétní metriky, je proveden post-hoc test, jehož výsledky jsou prezentovány formou barevné škály prostřednictvím heatmapy. Pro přehlednost je v heatmapě skryt horní trojúhelník, protože matice výsledků je symetrická. Matice se skládá z p-hodnot, které vyjadřují výsledek testu a lze je interpretovat jako pravděpodobnost, že jsou průměry či mediány daných skupin statisticky stejné. Hodnota menší než hladina významnosti ($\alpha = 0,05$) značí významný statistický rozdíl. V heatmapě pro konkrétní metriku jsou obsaženy pouze modely, u nichž byl proveden post-hoc test ke zjištění rozdílů.

Model **CNN LSTM** dosahuje nejvyšší přesnosti i vyvážené přesnosti na kanálech 1 a 4. S postupným navyšováním frekvence dochází k poklesu hodnot obou metrik, přičemž rozdíl mezi mediány činí až 10%. Ze sloupcového grafu na obrázku 4.2 je rovněž patrné, že pro vyšší kanály (4, 5 a 7) je sice medián nižší, ale nižší je i rozptyl hodnot a model vykazoval napříč běhy stabilnější výkonnost. Met-

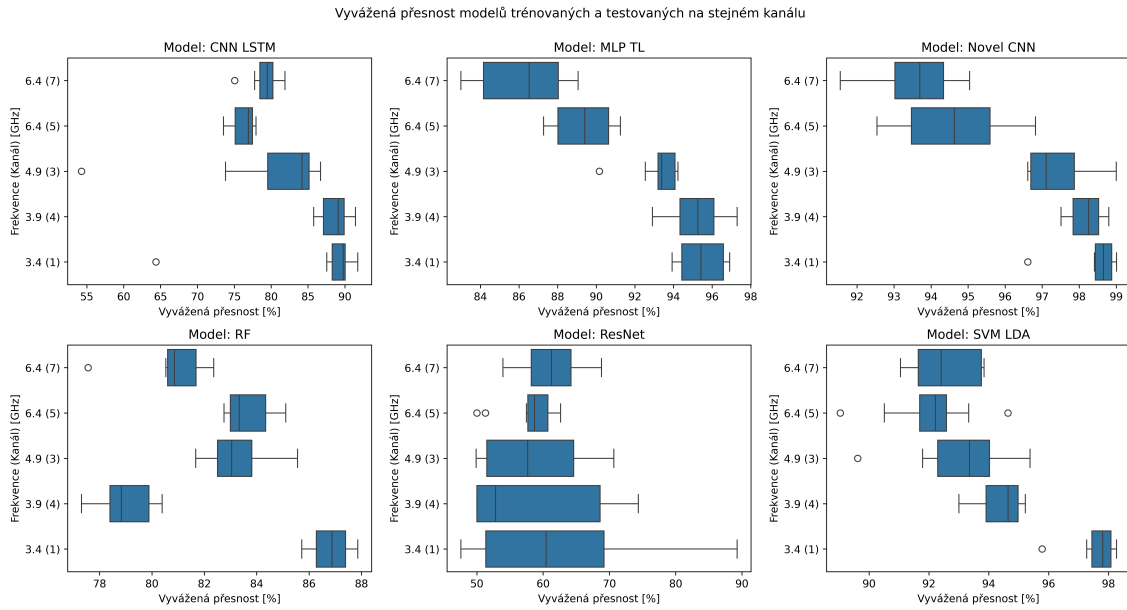
Model, Metriky	Kanáal					Statistické metriky		
	1	3	4	5	7	Normalita	Homogenita	Významný rozdíl
Model CNN LSTM								
Přesnost [%]	85.0 (83.2 – 86.9)	80.9 (79.8 – 82.0)	87.5 (86.2 – 88.8)	75.7 (71.5 – 79.8)	75.7 (73.9 – 77.4)	Ne	Ano	Ano
Vyvážená přesnost [%]	89.8 (88.9 – 90.7)	84.2 (81.4 – 87.0)	89.1 (87.7 – 90.5)	76.9 (75.7 – 78.1)	79.5 (78.6 – 80.4)	Ne	Ano	Ano
F1 skóre [%]	90.2 (88.8 – 91.6)	87.5 (86.6 – 88.4)	92.1 (91.2 – 92.9)	83.6 (80.3 – 87.0)	83.4 (81.9 – 84.9)	Ne	Ano	Ano
ROC AUC [%]	95.8 (95.3 – 96.4)	90.8 (89.4 – 92.1)	95.3 (94.6 – 96.0)	85.1 (84.6 – 85.6)	86.8 (86.3 – 87.3)	Ne	Ano	Ano
Model MLP TL								
Přesnost [%]	94.2 (93.1 – 95.3)	93.5 (92.9 – 94.2)	94.3 (92.5 – 96.0)	87.9 (86.8 – 89.0)	85.5 (84.1 – 86.9)	Ne	Ano	Ano
Vyvážená přesnost [%]	95.4 (94.3 – 96.5)	93.4 (92.9 – 93.8)	95.3 (94.4 – 96.1)	89.4 (88.1 – 90.7)	86.5 (84.6 – 88.5)	Ne	Ne	Ano
F1 skóre [%]	96.4 (95.7 – 97.1)	96.0 (95.6 – 96.4)	96.4 (95.3 – 97.6)	92.3 (91.6 – 93.1)	90.6 (89.7 – 91.6)	Ne	Ano	Ano
ROC AUC [%]	99.3 (98.9 – 99.7)	98.1 (97.9 – 98.3)	99.0 (98.8 – 99.1)	96.0 (95.4 – 96.5)	93.5 (92.3 – 94.7)	Ne	Ne	Ano
Model NovelCNN								
Přesnost [%]	98.2 (97.9 – 98.4)	96.7 (95.7 – 97.7)	97.9 (97.7 – 98.2)	94.0 (92.4 – 95.6)	94.1 (93.4 – 94.8)	Ne	Ne	Ano
Vyvážená přesnost [%]	98.7 (98.4 – 98.9)	97.1 (96.5 – 97.7)	98.3 (97.9 – 98.6)	94.6 (93.6 – 95.7)	93.7 (93.0 – 94.4)	Ne	Ne	Ano
F1 skóre [%]	98.9 (98.7 – 99.1)	98.0 (97.4 – 98.6)	98.7 (98.6 – 98.9)	96.3 (95.3 – 97.3)	96.4 (95.9 – 96.8)	Ne	Ne	Ano
ROC AUC [%]	99.8 (99.6 – 99.9)	99.7 (99.5 – 99.9)	99.7 (99.6 – 99.9)	99.0 (98.8 – 99.3)	97.5 (97.1 – 97.9)	Ano	Ne	Ano
Model RF								
Přesnost [%]	84.6 ± 1.4	79.1 ± 1.1	75.6 ± 1.1	80.1 ± 0.8	77.8 ± 1.5	Ano	Ano	Ano
Vyvážená přesnost [%]	86.9 (86.3 – 87.4)	83.0 (82.4 – 83.7)	78.8 (78.1 – 79.6)	83.3 (82.7 – 84.0)	80.9 (80.3 – 81.4)	Ne	Ano	Ano
F1 skóre [%]	90.0 ± 1.0	86.0 ± 0.8	83.5 ± 0.9	86.8 ± 0.6	85.1 ± 1.2	Ano	Ano	Ano
ROC AUC [%]	86.9 (86.3 – 87.4)	83.0 (82.4 – 83.7)	78.8 (78.1 – 79.6)	83.3 (82.7 – 84.0)	80.9 (80.3 – 81.4)	Ne	Ano	Ano
Model Resnet								
Přesnost [%]	81.6 (77.8 – 85.5)	76.0 (60.4 – 91.6)	74.5 (60.8 – 88.3)	45.2 (27.7 – 62.7)	56.5 (33.7 – 79.3)	Ne	Ne	Ano
Vyvážená přesnost [%]	60.4 (51.5 – 69.4)	57.7 (51.1 – 64.2)	52.8 (43.5 – 62.1)	58.7 (57.2 – 60.2)	61.3 (58.3 – 64.3)	Ne	Ano	Ne
F1 skóre [%]	89.6 (86.8 – 92.4)	84.6 (69.8 – 99.4)	82.9 (69.6 – 96.2)	52.4 (29.5 – 75.3)	66.9 (41.4 – 92.5)	Ne	Ne	Ano
ROC AUC [%]	80.7 ± 8.2	74.4 ± 7.6	75.1 ± 8.7	73.0 ± 7.1	73.0 ± 5.3	Ano	Ano	Ne
Model SVM LDA								
Přesnost [%]	97.8 (97.4 – 98.2)	95.0 (94.5 – 95.4)	95.1 (94.7 – 95.5)	93.7 (93.0 – 94.4)	94.2 (93.6 – 94.7)	Ne	Ano	Ano
Vyvážená přesnost [%]	97.8 (97.5 – 98.1)	93.4 (92.5 – 94.2)	94.6 (94.1 – 95.2)	92.2 (91.8 – 92.7)	92.4 (91.4 – 93.5)	Ne	Ano	Ano
F1 skóre [%]	98.7 (98.5 – 98.9)	97.0 (96.7 – 97.2)	97.0 (96.8 – 97.3)	96.2 (95.7 – 96.7)	96.5 (96.1 – 96.8)	Ne	Ano	Ano
ROC AUC [%]	97.8 (97.5 – 98.1)	93.4 (92.5 – 94.2)	94.6 (94.1 – 95.2)	92.2 (91.8 – 92.7)	92.4 (91.4 – 93.5)	Ne	Ano	Ano

Tab. 4.2: Přehled výkonnostních metrik modelů trénovaných a ověřovaných na stejném kanálu.

riky ROC AUC a F1 skóre vykazují stejný průběh jako oba typy přesností – nejvyšší jsou pro kanály 1 a 4 a se zvyšující se frekvencí klesají až o 10 %. Rozptyl metrik klesá stejně jako u přesností. Rozptyl všech metrik je největší na kanálu 3. Statistické testy detekují pro všechny metriky významný statistický rozdíl. Z heatmapy na obrázku 4.3 pro vyváženou přesnost vyplývá, že největší rozdíl je mezi spodními a horními frekvencemi. To platí i pro ostatní metriky, přičemž významný statistický rozdíl je detekován mezi kanálem 1 a kanály 5, 7, mezi kanálem 4 a kanály 5, 7. U přesnosti a F1 skóre se významný rozdíl objevuje také mezi kanály 3 a 4.

Model **Novel CNN** dosahuje velmi konzistentních výsledků napříč všemi kanály, kdy rozdíl mezi mediány napříč metrikami nepřesahuje 5 %. Nejvyšších přesností obou typů je dosaženo na kanálech 1 a 4, s narůstající frekvencí dochází opět k poklesu výkonnosti modelu, což je v tomto scénáři typické pro všechny modely. Ze sloupcového grafu je patrný nízký rozptyl všech sledovaných metrik napříč kanály, což značí konzistentní výkonnost modelu napříč běhy. Výsledkem statistických testů pro každou metriku je vždy detekce významného rozdílu. Rozdělíme-li kanály na 2 skupiny (1, 4 a 5, 7), k detekci rozdílu dochází ve všech případech právě mezi těmito 2 skupinami. Pro metriky přesnost a F1 skóre je významný rozdíl detekován také u dvojic kanálů 3, 5 a 3, 7.

Velmi kolísavých výsledků dosahuje model **Resnet**. Při sledování metrik přesnost a F1 skóre je patrný skokový pokles o přibližně 30 % při přechodu na kanál

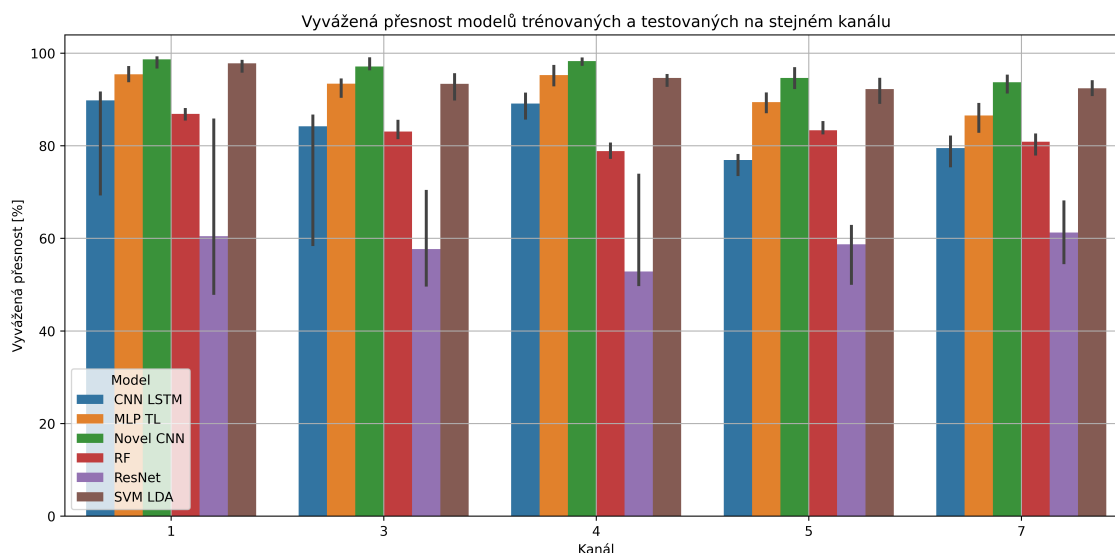


Obr. 4.1: Krabicový graf vyvážené přesnosti modelů trénovaných a testovaných na stejném kanálu.

5 a 7. U metriky ROC AUC lze pozorovat vyrovnané výsledky s výjimkou kanálu 1, u kterého lze pozorovat naopak nárůst. Vyvážená přesnost dosahuje vyrovnaných mediánů napříč kanály s minimem na kanálu 4. Na rozdíl od ostatních modelů je přítomen velmi vysoký rozptyl hodnot všech metrik napříč kanály. Schopnosti modelu jsou tedy velmi nekonzistentní. Možnou příčinou může být malá velikost trénovací sady v poměru k hloubce modelu. Statistické testy odhalily statisticky významné rozdíly pouze mezi hodnotami přesností a F1 skóre. Pro přesnost byl rozdíl detekován pouze mezi kanály 1 a 5, pro F1 skóre pouze mezi kanálem 1 a 7.

Pro model **MLP TL** nebylo v tomto scénáři využito přenesené učení. Stejně jako u výše popisovaných modelů dosahuje obecně nejvyšší výkonnosti na kanálech 1 a 4, přičemž se vzrůstající frekvencí výkonnost modelu klesá. Rozdíl mezi mediány nepřesahuje 9%. Stejně jako u modelu *CNN LSTM* lze pozorovat skokové snížení schopností modelu při přechodu na kanál 5 a 7. Naopak rozdílem proti zmíněnému modelu je nízký rozptyl hodnot sledovaných metrik napříč kanály. Výkonnost modelu je i přes zmíněný skok velmi konzistentní. Statistickými testy byl detekován významný rozdíl u všech sledovaných metrik. Z heatmap vyplývá, že rozdíly jsou napříč metrikami mezi stejnými kanály, které odpovídají popisovanému skokovému snížení schopností modelů. Rozdíly byly detekovány mezi kanálem 1, 3, 4 a kanály 5 a 7.

Model **SVM LDA** dosahuje velmi konzistentní výkonnosti se stejným průběhem, který byl již zmíněn např. u modelů *Novel CNN* a *MLP TL*. Rozdíl mezi mediány

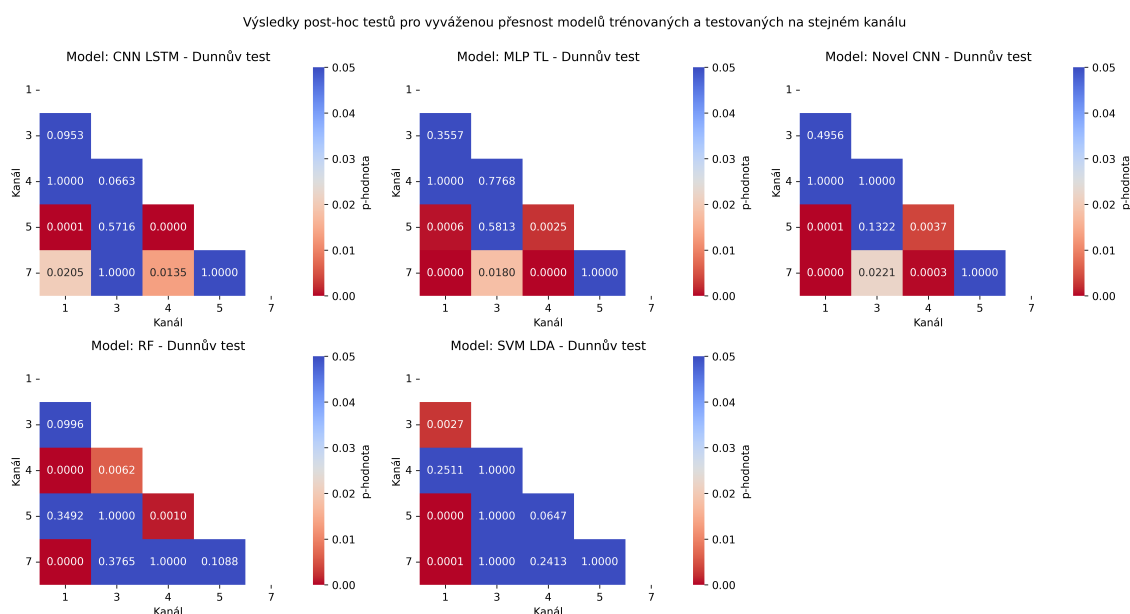


Obr. 4.2: Sloupcový graf vyvážené přesnosti modelů trénovaných a testovaných na stejném kanálu.

se pohybuje maximálně v blízkosti 5 %. Nejvyšší výkonnosti je dosaženo pro kanál 1. Po kanálu 1 následuje pro některé metriky výraznější propad, po kterém se zvyšující se frekvencí dochází k postupnému poklesu výkonnosti bez dalšího výrazného propadu, což potvrzují i krabicové grafy. Rozptyl hodnot jednotlivých metrik je napříč kanály shodně velmi malý, výjimku představuje pouze metrika přesnost, u níž je pro kanály 3 a 7 rozptyl v porovnání s ostatními vyšší. Výsledkem statistických testů pro každou metriku je vždy detekce významného rozdílu. Rozdělíme-li kanály na 2 skupiny (1 a 3, 5, 7), k detekci rozdílu dochází ve všech případech právě mezi těmito 2 skupinami.

Skokové rozdíly mezi hodnotami jednotlivých metrik vykazuje model **RF**. Přestože rozdíl mezi mediány nepřesahuje 7 %, nelze v posloupnosti sledovat trend přítomný u předchozích modelů. Nejvyšší výkonnosti je dosaženo pro kanál 1, následují skokové poklesy výkonnosti pro kanály 3 a 4. Výkonnost kanálu 5 je přibližně srovnatelná s výkonností modelu na kanálu 3 a následuje opět skokový pokles pro kanál 7. Nejhorších výsledků model dosahuje pro kanál 4. S výjimkou metrik přesnost a F1 skóre jsou rozptýleny hodnoty jednotlivých metrik napříč kanály velmi nízké a model tak vykazuje konzistentní výsledky napříč běhy. U zmíněných metrik je rozptyl mírně vyšší pro kanály 3 a 7. Pro hodnoty všech metrik byl statistickými testy detekován významný rozdíl. Pro metriky vyvážená přesnost a ROC AUC se rozdíl nachází mezi kanálem 1 a kanály 4, 7, mezi kanálem 4 a kanály 3 a 5. Pro zbylé metriky byl rozdíl detekován mezi kanálem 1 a kanály 3, 5 a 7.

V rámci tohoto scénáře byl tedy model vždy trénován a testován na stejném



Obr. 4.3: Heatmapa s výsledky post-hoc testů znázorňující rozdíly v hodnotách metriky „vyvážená přesnost“ napříč kanály.

datasetu. Sledovaným trendem je pokles výkonnosti modelu se zvyšující se frekvencí s maximální výkonností na kanálu 1 a lze tedy pozorovat rozdíly ve výkonnostech modelů na datech různých kanálů.

4.3 Scénář 2 – Trénink a testování na různých kanálech

Ve druhém scénáři jsou využívány modely z 1. scénáře, jejichž výkonnost je ale v tomto případě testována na všech kanálech různých od trénovacího. Cílem scénáře je přímo ověřit přenositelnost modelu natrénovaném na určitém kanálu a testovaném na kanálu jiném. Každý model byl ověřován – shodně jako v 1. scénáři – celkem 10x, vždy vůči modelu natrénovanému na jiném rozdělení trénovacích dat.

K vyjádření rozdílu ve výkonnosti je prezentován rozdíl mezi trénovacími a testovacími (validačními) hodnotami sledovaných metrik. Trénovací hodnoty odpovídají hodnotám metrik získaným v 1. scénáři a testovací hodnoty odpovídají hodnotám získaným v rámci tohoto scénáře při ověřování na jiném než trénovacím kanálu. Například pro ověřování výkonnosti modelu na kanálu 1 odpovídají trénovací (referenční) hodnoty metrik hodnotám získaným tréninkem a testováním modelu na tomto kanálu a testovací hodnoty odpovídají hodnotám metrik získaným testováním modelů na kanálu 1, přičemž tyto modely byly natrénovány na ostatních kanálech

(3, 4, 5 a 7). Očekávány jsou kladné hodnoty rozdílů, které znamenají pokles výkonnosti modelu při změně frekvenční domény. Čím nižší uvedená hodnota je, tím menší je pokles výkonnosti.

Provedeny byly stejné statistické testy, jako v předchozím scénáři 4.2. Testy jsou vždy provedeny pro každý model, metriku a daný trénovací kanál. Testují tedy nulovou hypotézu H_0 , která spočívá v neexistenci významného statistického rozdílu mezi sledovanými metrikami napříč testovacími kanály modelu trénovaného na konkrétním jiném kanálu. Stejným způsobem jsou prezentovány i grafické výstupy – v rámci výstupu je vždy konstantní sledovaná metrika a trénovací kanál. Získané hodnoty sledovaných metrik a grafické výstupy jsou prezentovány následujícím způsobem:

- Hodnoty sledovaných metrik modelu:
 - V souboru „scenar2.csv“, který je součástí elektronické přílohy, včetně výsledků statistických testů testujících H_0 .
 - Heatmapami, které prezentují hodnoty sledovaných metrik pro všechny kombinace kanálů. Tento způsob grafické reprezentace byl zvolen pro svou přehlednost pro tento typ závislosti. Heatmapa pro 1 ze sledovaných metrik je součástí této kapitoly (obrázek 4.4) a soubor všech heatmap pro sledované metriky je součástí přílohy této práce – obrázky A.1, A.2, A.3 a A.4.
 - Krabicovými grafy, které jsou součástí elektronické přílohy.
 - Sloupcovými grafy, které jsou součástí elektronické přílohy.
- Rozdíly ve výkonnosti:
 - V tabulce 4.3, která je součástí této kapitoly, jsou prezentovány rozdíly pro metriku vyvážená přesnost. V tabulce je uveden rozdíl metrik pro každou kombinaci trénovacího a testovacího kanálu. Uveden je rovněž aritmetický průměr těchto hodnot pro účely posouzení možného nejuniverzálnějšího tréninkového kanálu či nejhůře přenositelného kanálu. Rozdíly ve všech sledovaných metrikách jsou prezentovány v tabulce A.1, která je součástí přílohy této práce.
 - Prostřednictvím heatmapy pro každou metriku. Heatmapa pro metriku přesnost je součástí této kapitoly a nachází se na obrázku 4.5. Soubor všech heatmap je součástí přílohy této práce v podobě obrázků A.5, A.6, A.7 a A.8.
- Výsledky post-hoc testů:
 - Prostřednictvím heatmap, jenž jsou součástí elektronické přílohy, pro všechny sledované metriky a konkrétní trénovací kanály.
- Umístění a označení souborů v elektronické příloze:
 - Soubory se nacházejí ve složce „\\Vliv frekvence\Scenar2“.
 - Soubory ve formátu „.webp“ s grafickými výstupy nesou vždy označení

Testovací kanál	Trénovací kanál - Model CNN LSTM						Testovací kanál	Trénovací kanál - Model MLP TL						Testovací kanál	Trénovací kanál - Model Novel CNN					
	1	3	4	5	7	Ø		1	3	4	5	7	Ø		1	3	4	5	7	Ø
1	-	12,3	15,4	33,1	24,7	21,4	1	-	9,4	9,3	16,4	12,1	11,8	1	-	28,7	19,2	43,5	31,3	30,7
3	8,1	-	13,3	32,2	30,4	21,0	3	11,0	-	14,8	16,5	17,9	15,1	3	18,7	-	24,1	32,7	31,0	26,6
4	16,4	18,5	-	21,8	14,7	17,9	4	13,1	11,8	-	13,9	13,8	13,1	4	18,8	28,9	-	33,2	24,8	26,4
5	25,0	26,1	12,3	-	4,1	16,9	5	20,6	17,2	16,0	-	11,0	16,2	5	45,7	32,3	23,9	-	18,0	30,0
7	26,8	27,8	13,0	11,2	-	19,7	7	24,7	19,7	19,0	11,4	-	18,7	7	34,7	32,7	24,0	17,2	-	27,1
Ø	19,1	21,2	13,5	24,6	18,5	-	Ø	17,4	14,5	14,8	14,6	13,7	-	Ø	29,5	30,7	22,8	31,7	26,3	-
Testovací kanál	Trénovací kanál - Model RF						Testovací kanál	Trénovací kanál - Model Resnet						Testovací kanál	Trénovací kanál - Model SVM					
	1	3	4	5	7	Ø		1	3	4	5	7	Ø		1	3	4	5	7	Ø
1	-	16,5	24,6	25,1	26,5	23,2	1	-	2,6	9,5	12,4	16,8	10,3	1	-	37,8	30,6	54,1	40,0	40,6
3	20,9	-	12,9	18,9	18,9	17,9	3	6,3	-	2,8	8,2	14,9	8,1	3	37,4	-	29,1	31,4	33,7	32,9
4	23,4	9,3	-	13,4	18,2	16,1	4	1,1	0,3	-	2,6	3,9	2,0	4	36,1	29,7	-	38,9	32,5	34,3
5	31,9	20,1	19,1	-	10,1	20,3	5	16,8	10,4	8,7	-	0,9	9,2	5	41,3	36,0	20,0	-	10,7	27,0
7	31,2	21,9	24,5	13,4	-	22,7	7	17,6	15,4	11,3	4,8	-	12,2	7	38,7	33,1	19,0	18,7	-	27,4
Ø	26,8	17,0	20,3	17,7	18,4	-	Ø	10,5	7,2	8,1	7,0	9,1	-	Ø	38,4	34,2	24,7	35,8	29,2	-

Tab. 4.3: Rozdíly ve sledované metrice „vyvážená přesnost“ pro modely trénované na jiném než testovacím kanále,

ve formátu „*SC_TP_OT_M_CH*“, kde:

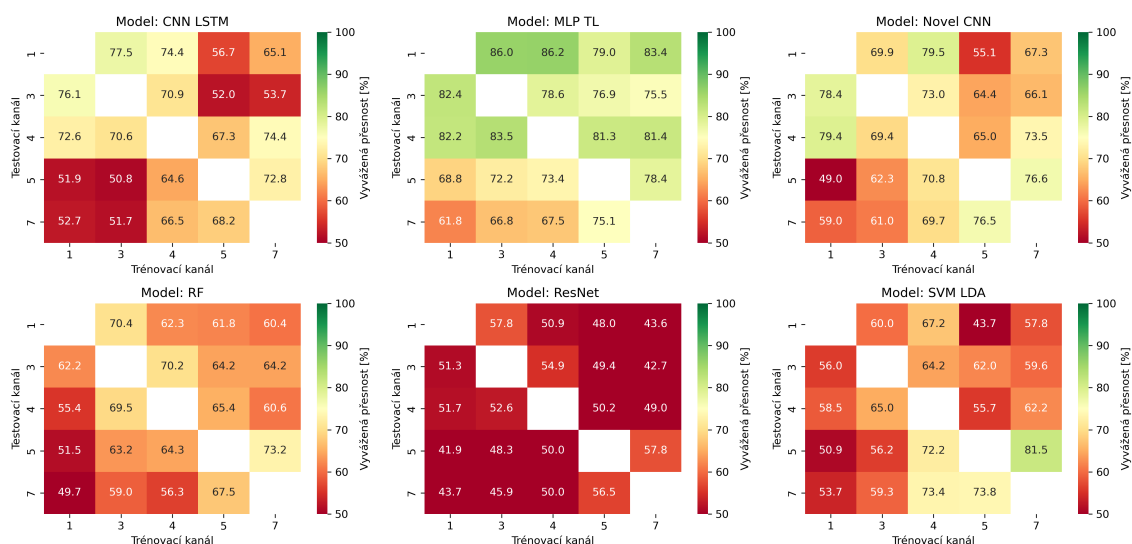
- * **SC** je vždy „Scenar2“.
- * **TP** udává, zda se jedná o hodnoty sledovaných metrik („Abs“), rozdíly v hodnotách sledovaných metrik („Diff“) či o výsledky post-hoc testů („PostHoc“).
- * **OT** představuje typ výstupu, a může nabývat hodnot „Barplot“, „Boxplot“ a „Heatmap“.
- * **M** představuje typ metriky, jejíž hodnoty grafický výstup prezentuje. Pro označení metrik jsou použity anglické ekvivalenty „Accuracy“ pro přesnost, „BalancedAccuracy“ pro vyváženou přesnost a pro zbývající metriky je použito „F1“ a „RocAuc“.
- * **CH** představuje trénovací kanál modelu, který byl ověřován s využitím kanálů vyobrazených v grafickém výstupu. Kanál nemusí být uveden v případech, kdy grafický výstup není závislý na použitém trénovacím kanálu.

S využitím výše zmíněných výstupů bude v následujícím textu posouzen vliv frekvence na přenositelnost jednotlivých modelů. Využity budou zejména tabulky a heatmapy s rozdíly mezi trénovacími a testovacími hodnotami a heatmapy s hodnotami testovacích metrik.

Z rozdílů hodnot metriky vyvážená přesnost pro model **CNN LSTM** vyplývá, že má model největší problémy s přenosem mezi nižšími a vyššími kanály. Rozdíl v rámci skupiny kanálů 1, 3 a v rámci skupiny kanálů 5 a 7 je přibližně poloviční v porovnání s rozdílem mezi těmito skupinami. Obdobná situace je pozorovatelná i u ROC AUC, což značí celkový pokles schopností modelů při přenosu mezi těmito 2 skupinami. Průměrné hodnoty rozdílů metrik dosahují maximální hodnoty 22,9 % a ve všech případech je tomu tak u modelu ověřovaného na kanálu 1, který lze tak označit za nejobtížnější doménu pro nasazení modelu trénovaného na jiné doméně.

Z hlediska skutečných schopností modelu lze za nejlepší tréninkovou doménu označit kanál 4, na kterém je dosaženo nejmenšího průměru rozdílu vyvážené přesnosti a ROC AUC. Na konkrétních testovacích a trénovacích datech je však nejlepší trénovací doménou kanál 1, na kterém je dosaženo nejmenšího rozdílu přesnosti a také F1 skóre. Výsledky statistických testů potvrzují, že statisticky významný není napříč různými trénovacími kanály rozdíl pouze mezi kanály 3, 4 a 5 a 7. Absolutní hodnoty ROC AUC indikují možnou přenositelnost po úpravě rozhodovací úrovně mezi kanály 1, 3 a 4 a mezi kanály 5 a 7. V ostatních případech lze očekávat klasifikační přesnost pod úrovní 75 %. Pro tento konkrétní (nevyvážený) dataset a stávající nastavení modelu lze o přenositelnosti uvažovat pouze z kanálu 1 na kanály 3 a 4, kdy přesnost v obou případech dosahuje téměř 80 %, a z kanálu 7 na kanál 5, při kterém je přesnost modelu 76,1 %. Při přenosu modelu na jinou frekvenční doménu byla v testovaných případech zachována preciznost modelu, došlo však k výraznému poklesu jeho citlivosti.

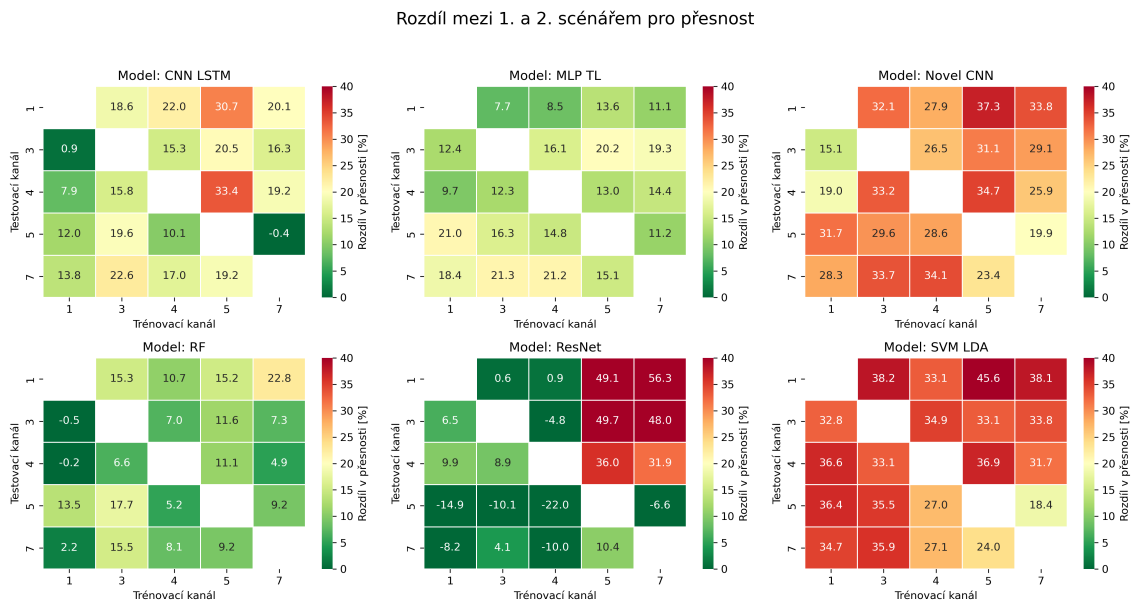
Vyvážená přesnost modelů testovaných a trénovaných na různých kanálech



Obr. 4.4: Heatmapa s hodnotami metriky „vyvážená přesnost“ pro modely trénované a testované na různých kanálech.

Model **Novel CNN** přenosem na jinou frekvenční doménu ztrácí velkou část svých schopností. Z rozdílů hodnot jednotlivých metrik je patrné, že model vykazuje overfitting na původní doménu. Z hlediska rozdílů v přesnostech platí, že menšího rozdílu je dosaženo v rámci skupin vyšší (kanály 5 a 7) a nižší frekvence (kanály 1, 3, 4), což potvrzují i výsledky statistických testů. I ty nejmenší rozdíly v přesnostech se však přibližují 20 %. Nejmenších rozdílů v porovnání s trénovacími hodnotami je dosaženo mezi kanály 1 a 3, 1 a 4 a kanály 5 a 7. Průměrné hodnoty rozdílů

metrik dosahují maximální hodnoty 32,8%, ve 3 případech je maximálních hodnot dosaženo u modelu testovaném na kanálu 1 a ve zbývajícím případě u modelu testovaném na kanálu 4. Rozdíl průměrných rozdílů ve vyvážené přesnosti činí 0,7% v neprospěch kanálu 1, zatímco rozdíl v ROC AUC je 4,5% v neprospěch kanálu 4. Oba tyto kanály lze tak označit za nejobtížnější doménu pro validaci schopností modelu. Z hlediska obecných schopností modelu lze za nejlepší tréninkovou doménu označit kanál 4, na konkrétních testovacích a trénovacích datech a použitím nastavení modelu je však nejlepší trénovací doménou kanál 1, na kterém je dosaženo nejmenšího rozdílu přesnosti a také F1 skóre. Přestože jsou rozdíly oproti tréninkovým hodnotám ve všech případech poměrně výrazné, existují případy, ve kterých je absolutní výkonnost modelu v porovnání s ostatními řešeními stále ucházející. Při úpravě rozhodovací úrovně modelu by přenositelnost mohla být možná mezi kanály 1, 3 a 4, z kanálu 4 na kanál 7 a mezi kanály 5 a 7. Pro tento konkrétní dataset je při přenosu z kanálu 1 na kanály 3 a 4 dosaženo přesnosti alespoň 79% a při přenosu z kanálů 4 a 5 na kanál 7 přesnosti alespoň 72%, což potvrzuje kanál 1 jako nejuniverzálnější tréninkovou doménu pro tento dataset. Při přenosu modelu na jinou frekvenční doménu byla v testovaných případech zachována preciznost modelu, došlo však k výraznému poklesu jeho citlivosti.



Obr. 4.5: Heatmapa s rozdíly ve sledované metrice „přesnost“ pro modely trénované na jiném než testovacím kanálu vůči hodnotám metriky získaných tréninkem a ověřováním na stejném kanálu.

U modelu **Resnet** lze pozorovat zanedbatelný, v případě metrik přesnost a F1 skóre dokonce i záporný rozdíl mezi hodnotami sledovaných metrik. U metrik po-

pisujících obecné schopnosti modelu je velmi nízkého rozdílu dosaženo při přenosu mezi kanály 1, 3 a 4 a při přenosu mezi kanály 5 a 7, zatímco u metrik reflektujících schopnosti modelu na použitém datasetu je nízkých, až vysoce záporných hodnot dosaženo při přenosech z kanálů 1, 3 a 4. V obou případech je třeba uvést, že referenční hodnoty jsou velmi nízké a v případě záporných hodnot jsou hodnoty těchto metrik doprovázeny extrémními rozptyly. Průměrné hodnoty rozdílů metrik dosahují maximální hodnoty 27,0 %, ve 3 případech je maximálních hodnot dosaženo u modelu testovaném na kanálu 1 a ve zbývajícím případě u modelu testovaném na kanálu 7. Rozdíl průměrných rozdílů ve vyvážené přesnosti činí 1,9 % v neprospěch kanálu 7 a rozdíl v ROC AUC 1,8 % naopak v neprospěch kanálu 1. Oba tyto kanály lze tak označit za nejobtížnější doménu pro validaci schopností modelu. Jako nejlepší trénovací doménu lze pro tento model označit kanál 4, na kterém je dosaženo nejmenších rozdílů v hodnotách sledovaných metrik. Přestože je vliv frekvence na přenositelnost tohoto modelu v porovnání s ostatními menší, je reálné nasazení stávajícího modelu na jinou frekvenční doménu vyloučeno. Z hodnot ROC AUC vychází pouze možná přenositelnost modelu z kanálů 3 a 4 na kanál 1. Pro dataset použitý v této části práce je přenositelnost realizovatelná z kanálů 3 a 4 na kanály 1 a 3, kdy je ve všech případech dosaženo přesnosti přes 80 %. Při přenosu modelu na jinou frekvenční doménu došlo k souměrnému poklesu preciznosti i citlivosti modelu. Je nezbytné brát v úvahu hloubku modelu Resnet a velmi malé množství trénovacích dat, díky čemuž model dosahuje velmi variabilních výsledků, což vede k ovlivnění výsledků tohoto scénáře.

Model **MLP TL** dosahuje menších rozdílů mezi metrikami zejména při přenosu v rámci skupin nižších a vyšších frekvencí. V tomto scénáři bylo na rozdíl od předchozího scénáře využito přenesené učení k dotrénování modelu na cílovou doménu. Nejmenší rozdíly v přesnostech jsou pod hranicí 10 % a průměrné hodnoty rozdílů všech sledovaných metrik dosahují maximální hodnoty 19,9 %. Průměrné hodnoty rozdílů metrik se navyšují spolu se zvyšující se frekvencí testovacího kanálu – nejmenší jsou pro kanál 1, následují kanály 4, 3 a 5 a nejvyšší jsou pro kanál 7. Za nejlepší cílovou doménu lze tedy jednoznačně považovat kanál 1, zatímco nejlepší doménou pro trénink, a zároveň nejobtížnější cílovou doménou, je kanál 7. V heatmapách pro výsledky post-hoc testů lze se stoupajícím tréninkovým kanálem pozorovat modraní mapy, které představuje neexistenci významných statistických rozdílů. To potvrzuje zjištění, že kanály 1, 3 a 4 představují pro tento model jednoduchou cílovou doménu. Matice se zvyšujícím se tréninkovým kanálem pak modrá z důvodu snižujícího se rozdílu mezi kanály 5 a 7. Hodnoty vyvážené přesnosti a ROC AUC indikují vysoké schopnosti modelu při přenosu z libovolného kanálu z testované množiny kanálů na kanály 1, 3 a 4 a při přenosu mezi kanály 5 a 7. Pro tento konkrétní dataset je dosaženo přesnosti alespoň 80 % při přenosu

na kanály 1 a 4 a při přenosu z kanálu 1 na kanál 3. Nižšího vlivu frekvence na přenositelnost modelu je v případě tohoto modelu docíleno zejména díky využití přeneseného učení. Při přenosu modelu na jinou frekvenční doménu došlo v testovaných případech k poklesu jeho citlivosti.

U modelu **RF**, který jako jediný z testovaných modelů využívá charakteristiky extrahované z impulsní odezvy, je rozdíl různých metrik napříč doménami velmi variabilní. U vyvážené přesnosti převažují vyšší rozdíly, což značí celkový pokles schopností modelu. Pro přesnost a F1 skóre jsou naopak rozdíly v hodnotách metrik nižší, což ukazuje na nižší vliv frekvence na přenositelnost modelu pro konkrétní (nevyvážený) dataset. Nejmenších rozdílů je dosaženo při přenosu mezi kanály 3 a 4 a mezi kanály 5 a 7. Průměrné hodnoty rozdílů metrik dosahují maximální hodnoty 23,2 % a ve všech případech je maximální průměrné hodnoty rozdílů dosaženo u modelu ověřovaném na kanálu 1, který lze tak označit za nejobtížnější doménu pro nasazení tohoto modelu trénovaného na jiné doméně. Z hlediska skutečných schopností modelu lze za nejlepší tréninkovou doménu označit kanál 3, pro použitá testovací a trénovací data a nastavení modelu je však nejlepší trénovací doménou kanál 1, na kterém je dosaženo nejmenšího rozdílu přesnosti a také F1 skóre. Na základě získaných hodnot vyvážené přesnosti lze při použití přeneseného modelu očekávat přesnost klasifikace v nejlepších případech v okolí 70 %. Při přenosu modelu na jiný kanál v rámci testovacího datasetu přesahuje hodnota vyvážené přesnosti 70 % pouze v případech přenosu z kanálu 3 na kanál 1, z kanálu 4 na kanál 3 a z kanálu 7 na kanál 5. Přesnosti nad 75 % dosahuje model přenesený z kanálu 1 na kanály 3, 4 a 7 a této hodnotě se blíží také pro přenos z kanálu 4 na kanál 5. Obecně lze konstatovat, že vliv frekvence na přenositelnost tohoto modelu je značný. Příčinou může být kromě jednoduchosti modelu i povaha vstupních dat, jejichž frekvenční závislost je diskutována v kapitole 4.5. Při přenosu modelu na jinou frekvenční doménu byla v testovaných případech zachována citlivost modelu, došlo však k poklesu jeho preciznosti.

Model **SVM LDA** v konceptu přenositelnosti v tomto scénáři naprosto selhává. Pokles hodnot sledovaných metrik je v případě přenosu na jinou frekvenční doménu výrazný napříč testovanými kanály. Výjimku tvoří pouze přenos z kanálů 4, 5 a 7 na kanály 5 a 7, přičemž nejnižších rozdílů napříč metrikami je dosaženo pro přenos z kanálu 5 na kanál 7. Z rozdílů hodnot sledovaných metrik vyplývá, že model vykazuje overfitting na trénovací doménu. Maximální průměrná hodnota rozdílů metrik dosahuje hranice 40,6 %. Z hodnot obou přesností na datech tohoto nevyváženého datasetu vyplývá, že je diskutovatelná přenositelnost pouze mezi kanály 5 a 7, u které lze dosáhnout přesnosti klasifikace nad 75 %. Při přenosu modelu na jinou frekvenční doménu došlo k poklesu citlivosti modelu a výraznému poklesu jeho preciznosti.

Z provedených experimentů v tomto scénáři jednoznačně vyplývá vliv frekvence jako parametru na přenositelnost modelů pro klasifikaci LOS a NLoS. Všechny modely vykazují jednoznačný pokles výkonnosti při přenosu na jinou frekvenční doménu. Tento závěr potvrzují i provedené statistické testy, které indikují významný statistický vliv napříč hodnotami sledovaných metrik. Vliv frekvence na přenositelnost se napříč modely liší, obecně však bývá menší mezi kanály 1 a 3 a mezi kanály 5 a 7. Mezi nejobtížnější validační domény patří kanály 1 a 7, nejuniverzálnější tréninkovou doménou jsou nejčastěji kanály 4 a 1.

4.4 Scénář 3 – Testování na kanálu vyřazeném z tréninku

Stejně jako ve 2. scénáři jsou i zde využívány stejné modely z 1. scénáře. V tomto scénáři jsou však modely trénovány na všech kanálech vyjma jednoho, na kterém je následně ověřována jejich výkonnost. Cílem scénáře je ověřit, zda lze tréninkem na širší frekvenční sadě zmírnit či dokonce eliminovat vliv frekvence na přenositelnost modelu. Vzhledem k povaze použitého datasetu lze zejména u hlubokých modelů očekávat zvýšení schopností modelu, neboť tréninková sada bude v porovnání s 1. scénářem 4x větší.

Hlavním cílem scénáře je prezentovat rozdíl vůči 2. scénáři. K vyjádření rozdílu bude stejně jako ve 2. scénáři prezentován rozdíl v hodnotách sledovaných metrik. Referenční hodnotou v tomto případě budou hodnoty z tohoto scénáře, od kterých budou odečteny hodnoty sledovaných metrik ze druhého scénáře. Rozdíly lze tedy interpretovat jako míru rozdílu ve schopnostech modelu mezi 3. a 2. scénářem, kdy kladné hodnoty znamenají nárůst schopností modelu v porovnání s 2. scénářem a zmírnění vlivu frekvence na přenositelnost modelu. Stejně jako ve zmiňovaném 2. scénáři je prezentován i rozdíl vůči referenční hodnotě z 1. scénáře, pro který platí, že čím nižší je jeho hodnota, tím menší je pokles výkonnosti modelu.

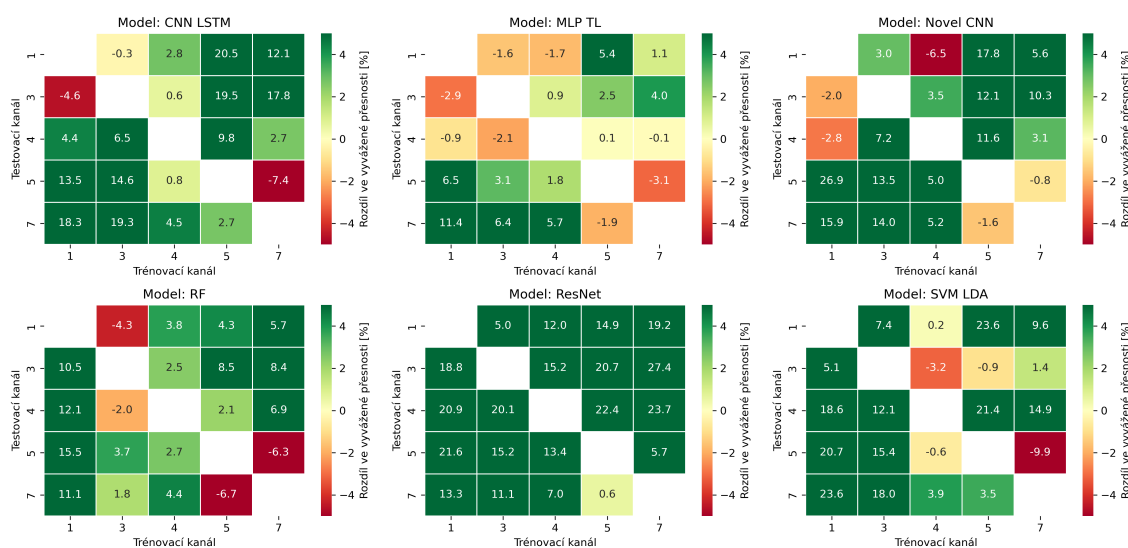
Analyzovány jsou metriky každého z použitých modelů vzhledem ke kanálu, ke kterému náleží data použitá pro trénink i ověřování. Statistické testy testují nulovou hypotézu H_0 , že neexistuje statisticky významný rozdíl mezi výkonností modelu trénovaném na různých kanálech. Testovány jsou vždy hodnoty 1 ze sledovaných metrik napříč všemi kanály.

Provedeny byly stejné statistické testy jako v 1. scénáři. Analyzovány jsou metriky každého z použitých modelů vzhledem ke kanálu, ke kterému náleží data použitá pro ověřování. Testují nulovou hypotézu H_0 , že neexistuje statisticky významný rozdíl mezi sledovanými metrikami napříč ověřovanými kanály. Získané hodnoty sledovaných metrik a grafické výstupy jsou prezentovány následujícím způsobem:

- Hodnoty sledovaných metrik modelu:
 - V tabulce 4.4 včetně výsledků statistických testů testujících H_0 .
 - V souboru „scenar3.csv“, který je součástí elektronické přílohy, včetně výsledků statistických testů testujících H_0 .
 - Krabicovými grafy, které jsou součástí elektronické přílohy.
 - Sloupcovými grafy, které jsou součástí elektronické přílohy.
- Rozdíly ve výkonnosti mezi 3. a 2. scénářem:
 - Prostřednictvím heatmapy pro každou metriku. Heatmapa pro metriku vyvážená přesnost je součástí této kapitoly a nachází se na obrázku 4.6. Soubor všech heatmap je součástí přílohy této práce v podobě obrázků A.10, A.9, A.11 a A.12. Heatmapa má stejný formát jako ve 2. scénáři a znázorňuje rozdíl mezi výkonností modelu ve 3. a 2. scénáři pro všechny kombinace trénovacích a testovacích kanálů ze 2. scénáře. Mapu je tedy vhodné číst způsobem, kdy je sledován konkrétní testovací kanál a mapa ve vodorovném směru prezentuje rozdíly ve výkonnosti modelu dosažené ve 2. scénáři tréninkem na různých trénovacích kanálech. Detailní interpretace hodnot jedné z heatmap je uvedena v následujícím odstavci.
 - V tabulce A.2 pro zachování způsobu interpretace rozdílů vůči 1. scénáři použitým ve 2. scénáři.
- Výsledky post-hoc testů:
 - Prostřednictvím heatmap pro všechny sledované metriky. Heatmapy jsou součástí elektronické přílohy.
- Umístění a označení souborů v elektronické příloze:
 - Soubory se nacházejí ve složce „\\Vliv frekvence\Scenar3“
 - Soubory ve formátu „.webp“ s grafickými výstupy nesou vždy označení ve formátu „**SC_OT_M**“, kde:
 - * **SC** je vždy „Scenar3“.
 - * **OT** představuje typ výstupu, a může nabývat hodnot „Barplot“, „Boxplot“ a „Heatmap“.
 - * **M** představuje typ metriky, jejíž hodnoty grafický výstup prezentuje. Pro označení metrik jsou použity anglické ekvivalenty „Accuracy“ pro přesnost, „BalancedAccuracy“ pro vyváženou přesnost a pro zbývající metriky je použito „F1“ a „RocAuc“.

Pro úplnost bude v rámci tohoto odstavce vysvětlen způsob interpretace hodnot z heatmapy 4.6 pro model *CNN LSTM*. Jak už bylo zmíněno, heatmapa znázorňuje rozdíl mezi výkonností modelu ve 3. a 2. scénáři pro všechny kombinace trénovacích a testovacích kanálů ze 2. scénáře. Na heatmapě pro vyváženou přesnost a model *CNN LSTM* jsou pro testovací kanál 7 uvedeny 4 hodnoty. Každá hodnota odpovídá jinému trénovacímu kanálu (1, 3, 4 a 5). Hodnoty vyjadřují rozdíl vyvážené

Rozdíl mezi 3. a 2. scénářem pro vyváženou přesnost



Obr. 4.6: Heatmapa s rozdíly ve sledované metrice „vyvážená přesnost“ mezi 3. a 2. scénářem.

přesnosti modelu trénovaného na všech těchto kanálech zároveň vůči vyvážené přesnosti modelu trénovaného vždy na 1 z kanálů. Pro testovací kanál 7 tedy hodnota 18,3% vyjadřuje, že model trénovaný pouze na kanálu 1 dosahuje z pohledu vyvážené přesnosti horších výsledků, přičemž rozdíl ve vyvážených přesnostech činí 18,3%. Protože tento scénář k tréninku využívá všechny kanály, je nezbytné porovnávat pouze nejlepší případy. Na uvedeném příkladu to lze vysvětlit tak, že při tréninku modelu pouze na kanálu 1 je rozdíl ve vyvážených přesnostech 18,3%, ale při tréninku modelu pouze na kanálu 5 činí rozdíl pouze 2,7%. Protože je v tomto scénáři předpoklad dostupnosti dat ze všech kanálů, je nezbytné porovnávat schopnosti modelu ověřovaného na kanálu 7 a trénovaného na všech ostatních kanálech právě vůči schopnostem modelu trénovaného na kanálu 5 a ověřovaného na kanálu 7.

V následujícím textu je posouzen vliv širší tréninkové datové (frekvenční) sady na přenositelnost jednotlivých modelů, a to zejména s využitím výstupů pro rozdíly ve výkonnosti modelů mezi 3. a 2. scénářem. Další výstupy mohou zůstat bez komentáře a mohou sloužit pro účely budoucího porovnávání modelů či jako základ pro ověření dosažených výsledků.

Model **CNN LSTM** vykazuje zlepšení vyvážené přesnosti pro ověřování na kanálech 4 a 7, na kterých je v porovnání s nejlepšími případy 2. scénáře dosaženo rozdílu alespoň 2,7%. Pro ROC AUC je přítomno zlepšení o 5,7%, ale pouze pro kanál 4. U metrik „přesnost“ a „F1 skóre“ reflektujících schopnosti modelu na použitém datasetu je přítomno pouze malé zlepšení a pouze pro kanál 7, které pro

přesnost dosahuje hodnoty 3,9 %.

Model **Novel CNN** vykazuje pouze zanedbatelné zlepšení F1 skóre v porovnání s nejlepšími případy 2. scénáře, v případě ostatních metrik zlepšení proti nejlepším výsledkům ze 2. scénáře přítomno není. Napříč všemi ověřovanými modely dosahuje spolu s modelem *MLP TL* nejlepších hodnot vyvážené přesnosti.

Model **MLP TL** nevykazuje zlepšení hodnot sledovaných metrik v porovnání s nejlepšími případy 2. scénáře. Napříč všemi ověřovanými řešeními dosahuje v tomto scénáři nejlepších hodnot obou přesností i ROC AUC, a to zejména z důvodu využití přeneseného učení.

Model **Resnet** vykazuje výrazné zlepšení vyvážené přesnosti pro ověřování na všech kanálech kromě kanálu 7, v případě ROC AUC na všech kanálech kromě kanálu 1. U vyvážené přesnosti se v případě kanálů 1 a 5 jedná o zlepšení o 5 %, v případě kanálu 3 o 15 %, a v případě kanálu 4 dokonce o 20 %. U ROC AUC je situace podobná, pro kanál 1 se jedná o zlepšení o 3,8 %, pro kanály 3 a 5 o 10 % a pro kanál 4 opět o 20 %. U metrik reflektujících přesnost modelu na použitém datasetu je přítomno zlepšení u pouze kanálů 4, 5 a 7, které ve všech případech činí alespoň 10 %. Napříč všemi ověřovanými modely dosahuje spolu s modelem *MLP TL* nejlepších hodnot přesnosti. Stejně jako u 2. scénáře i v tomto případě došlo ke zlepšení schopnosti modelu vůči prvnímu scénáři, ve kterém byl model trénován a testován na stejném kanálu. Na rozdíl od 1. i 2. scénáře je v tomto případě pozorovatelný stabilizační jev větší datové sady na tento hluboký model, díky čemuž byly minimalizovány rozptyly hodnot jednotlivých metrik, které napříč všemi modely patří k nižším.

Model **RF** vykazuje pouze malé zlepšení (2,5 %) vyvážené přesnosti v porovnání s nejlepšími případy 2. scénáře pro ověřování dat kanálu 3, v případě ostatních metrik zlepšení proti nejlepším výsledkům ze 2. scénáře přítomno není.

Model **SVM LDA** vykazuje výrazné zlepšení vyvážené přesnosti pro ověřování kanálu 4 (14,9 %) a mírné zlepšení pro ověřování na kanálu 7 (3,5 %). U metrik reflektujících přesnost modelu na použitém datasetu je přítomno zlepšení u kanálů 1, 4 a 7. Zatímco pro kanál 4 dochází ke zlepšení přesnosti o 9,4 % a F1 skóre o 7,2 %, u kanálů 1 a 7 je přítomno pouze malé zlepšení nepřesahující 3,5 %. Na schopnosti modelu má pozitivní vliv i eliminace overfittingu modelu na trénovací doménu v důsledku širší tréninkové sady.

Navzdory očekávání nedošlo k výraznému zmírnění vlivu frekvence u hlubokých modelů. Výrazné zmírnění vlivu frekvence na přenositelnost modelů proti 2. scénáři je pozorováno zejména u modelů *Resnet* a *SVM LDA*. V obou případech však na zmírnění vlivu může mimo širšího souboru trénovacích domén působit také větší velikost datové sady. V případě velmi hlubokého modelu *Resnet* došlo kromě zmírnění vlivu frekvence na přenositelnost i k celkové stabilizaci modelu, která je jednoznačně

Model, Metriky	Ověřovaný kanál					Statistické metriky		
	1	3	4	5	7	Normalita	Homogenita	Významný rozdíl
Model CNN LSTM								
Přesnost [%]	64.5 (61.5 – 67.4)	59.5 (54.8 – 64.2)	64.2 (56.5 – 71.9)	71.3 (67.2 – 75.5)	65.7 (60.6 – 70.8)	Ne	Ano	Ne
Vyvážená Přesnost [%]	77.2 (74.7 – 79.7)	71.5 (69.4 – 73.6)	77.1 (73.4 – 80.7)	65.4 (62.7 – 68.1)	71.0 (69.1 – 72.8)	Ne	Ano	Ano
F1 skóre [%]	73.2 (70.6 – 75.9)	68.6 (63.2 – 73.9)	72.7 (65.3 – 80.1)	81.1 (77.3 – 84.8)	75.3 (70.3 – 80.4)	Ne	Ano	Ne
ROC AUC [%]	89.1 (86.4 – 91.8)	77.4 (74.3 – 80.5)	86.8 (84.0 – 89.6)	74.4 (72.8 – 76.0)	75.8 (73.7 – 77.9)	Ne	Ano	Ano
Model MLP TL								
Přesnost [%]	84.7 (83.7 – 85.6)	78.9 (78.0 – 79.8)	79.5 (78.1 – 81.0)	74.7 (71.8 – 77.6)	68.0 (65.6 – 70.3)	Ne	Ano	Ano
Vyvážená Přesnost [%]	84.4 ± 2.0	79.4 ± 2.0	81.4 ± 2.4	75.3 ± 2.3	73.2 ± 2.2	Ano	Ano	Ano
F1 skóre [%]	90.2 (89.5 – 90.9)	86.2 (85.5 – 86.9)	86.5 (85.2 – 87.8)	83.1 (80.6 – 85.6)	77.2 (74.8 – 79.6)	Ne	Ano	Ano
ROC AUC [%]	91.6 ± 2.1	86.9 ± 1.4	89.8 ± 1.7	83.6 ± 2.3	80.3 ± 2.6	Ano	Ano	Ano
Model Novel CNN								
Přesnost [%]	69.9 ± 3.3	76.0 ± 1.5	73.8 ± 2.0	70.9 ± 3.2	67.8 ± 3.2	Ano	Ano	Ano
Vyvážená Přesnost [%]	72.9 ± 1.9	76.5 ± 1.7	76.6 ± 2.6	75.8 ± 1.5	75.0 ± 2.0	Ano	Ano	Ano
F1 skóre [%]	79.0 ± 3.1	84.0 ± 1.3	82.1 ± 1.7	79.6 ± 2.8	76.8 ± 3.0	Ano	Ano	Ano
ROC AUC [%]	80.4 ± 1.9	82.2 ± 1.3	84.4 ± 1.9	81.4 ± 1.6	82.4 ± 1.5	Ano	Ano	Ano
Model RF								
Přesnost [%]	68.7 ± 1.6	69.9 ± 0.9	68.1 ± 1.6	63.9 ± 1.1	69.7 ± 1.8	Ano	Ano	Ano
Vyvážená Přesnost [%]	66.1 (65.0 – 67.2)	72.6 (71.9 – 73.4)	67.5 (66.4 – 68.7)	66.9 (66.2 – 67.7)	60.8 (59.7 – 61.8)	Ne	Ano	Ano
F1 skóre [%]	78.9 ± 1.3	79.1 ± 0.7	78.2 ± 1.4	74.2 ± 1.0	80.3 ± 1.4	Ano	Ano	Ano
ROC AUC [%]	66.1 (65.0 – 67.2)	72.6 (71.9 – 73.4)	67.5 (66.4 – 68.7)	66.9 (66.2 – 67.7)	60.8 (59.7 – 61.8)	Ne	Ano	Ano
Model Resnet								
Přesnost [%]	80.5 ± 1.4	77.4 ± 1.8	77.6 ± 1.6	79.5 ± 1.3	76.4 ± 1.2	Ano	Ano	Ano
Vyvážená Přesnost [%]	62.9 ± 3.5	70.2 ± 2.5	72.6 ± 1.7	63.4 ± 2.7	57.0 ± 2.0	Ano	Ano	Ano
F1 skóre [%]	88.4 ± 0.9	85.6 ± 1.4	85.6 ± 1.3	87.7 ± 0.8	85.9 ± 0.8	Ano	Ano	Ano
ROC AUC [%]	78.8 ± 3.0	80.1 ± 2.2	82.6 ± 1.4	76.0 ± 2.1	67.5 ± 2.4	Ano	Ano	Ano
Model SVM								
Přesnost [%]	66.6 (66.5 – 66.8)	59.1 (57.9 – 60.3)	72.8 (72.1 – 73.4)	69.1 (68.5 – 69.8)	73.7 (73.1 – 74.3)	Ne	Ano	Ano
Vyvážená Přesnost [%]	67.4 (67.0 – 67.7)	61.1 (59.9 – 62.3)	77.1 (75.7 – 78.5)	71.6 (68.7 – 74.6)	77.3 (75.1 – 79.5)	Ne	Ano	Ano
F1 skóre [%]	76.8 (76.6 – 77.0)	70.3 (69.3 – 71.3)	81.1 (80.7 – 81.6)	78.8 (78.6 – 78.9)	82.0 (81.6 – 82.4)	Ne	Ano	Ano
ROC AUC [%]	67.4 (67.0 – 67.7)	61.1 (59.9 – 62.3)	77.1 (75.7 – 78.5)	71.6 (68.7 – 74.6)	77.3 (75.1 – 79.5)	Ne	Ano	Ano

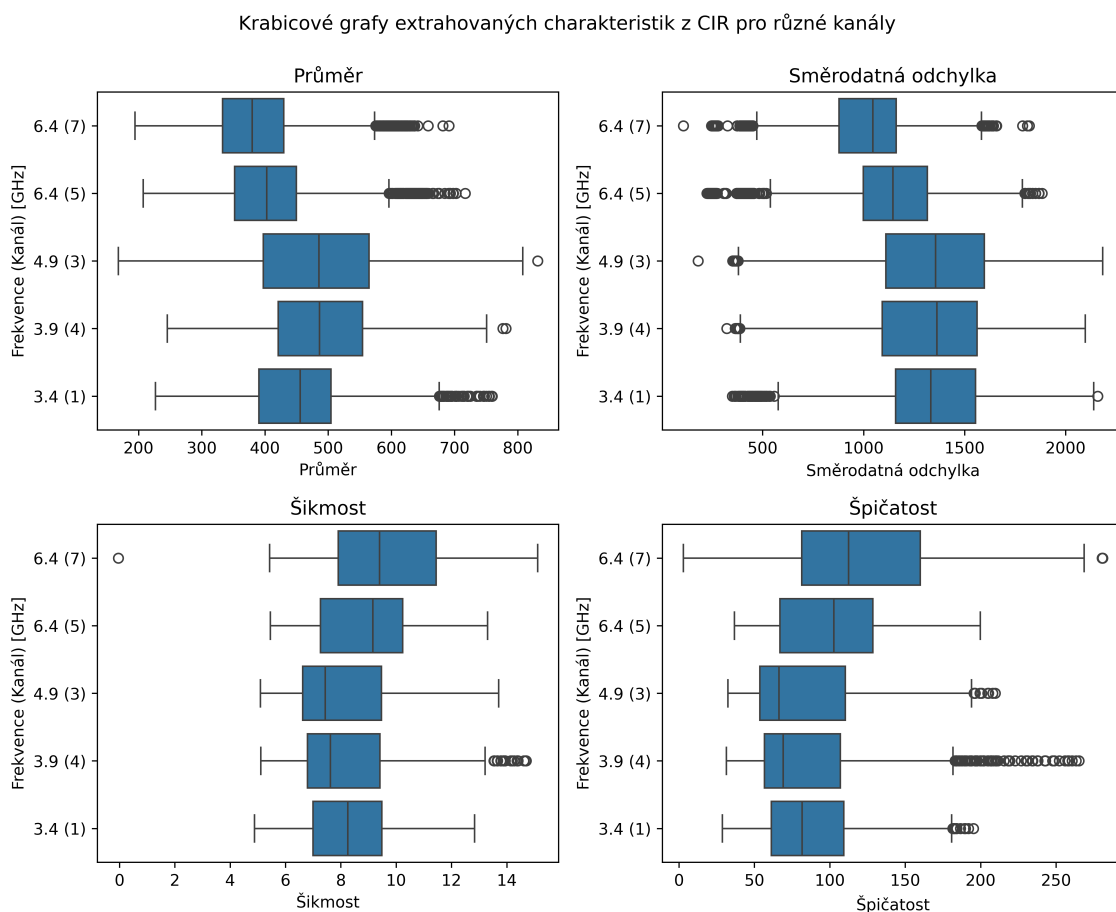
Tab. 4.4: Přehled výkonnostních metrik modelů trénovaných na datech všech kanálů současně vyjma kanálu, na kterém byl model ověřován.

důsledkem větší datové sady. U modelu *SVM LDA* nezbytné uvést citlivost LDA na nízké množství vzorků a overfitting modelu v 1. a 2. scénáři. Pro vyloučení vlivu LDA a nízkého množství vzorků na výsledky 2. scénáře by však bylo nezbytné scénáře replikovat s *SVM LDA* modelem bez LDA. Zmírnění vlivu frekvence na přenositelnost modelů je kromě modelu *Resnet* pozorováno u modelu *SVM LDA* (kanály 4 a 7), *CNN LSTM* (kanály 4 a 7) a u modelu *RF* (kanál 3). S výjimkou modelu *Resnet* je tedy případné zmírnění vlivu frekvence přítomno pouze pro vybranou menšinu kanálů, která se mezi modely různí.

4.5 Scénář 4 – Porovnání extrahovaných charakteristik modelu RF

Cílem posledního scénáře je porovnání extrahovaných charakteristik, které využívá model *RF*, pro všechny kanály v datasetu. Všechny scénáře se dosud zabývaly rozdílem ve výkonnostech jednotlivých modelů, ale žádný se nezabýval rozdíly v samotných datech. Jedná se o extrahované charakteristiky z impulsní odezvy kanálu představené v rámci kapitoly 3.2.7 – průměr, směrodatná odchylka, šikmost a špičatost.

Výstupy tohoto scénáře jsou pro snadnou interpretaci pouze grafické a jedná se o krabicový graf 4.7, violin graf 4.8 a Q-Q graf 4.9. Statistické testy nebyly z důvodu vysokého množství vzorků realizovány, neboť dle [83] mohou být na velký rozsah dat velmi citlivé a často detekovat i nevýznamné rozdíly. Výstupy tohoto scénáře jsou součástí této kapitoly a nejsou zahrnuty v příloze ani elektronické příloze.



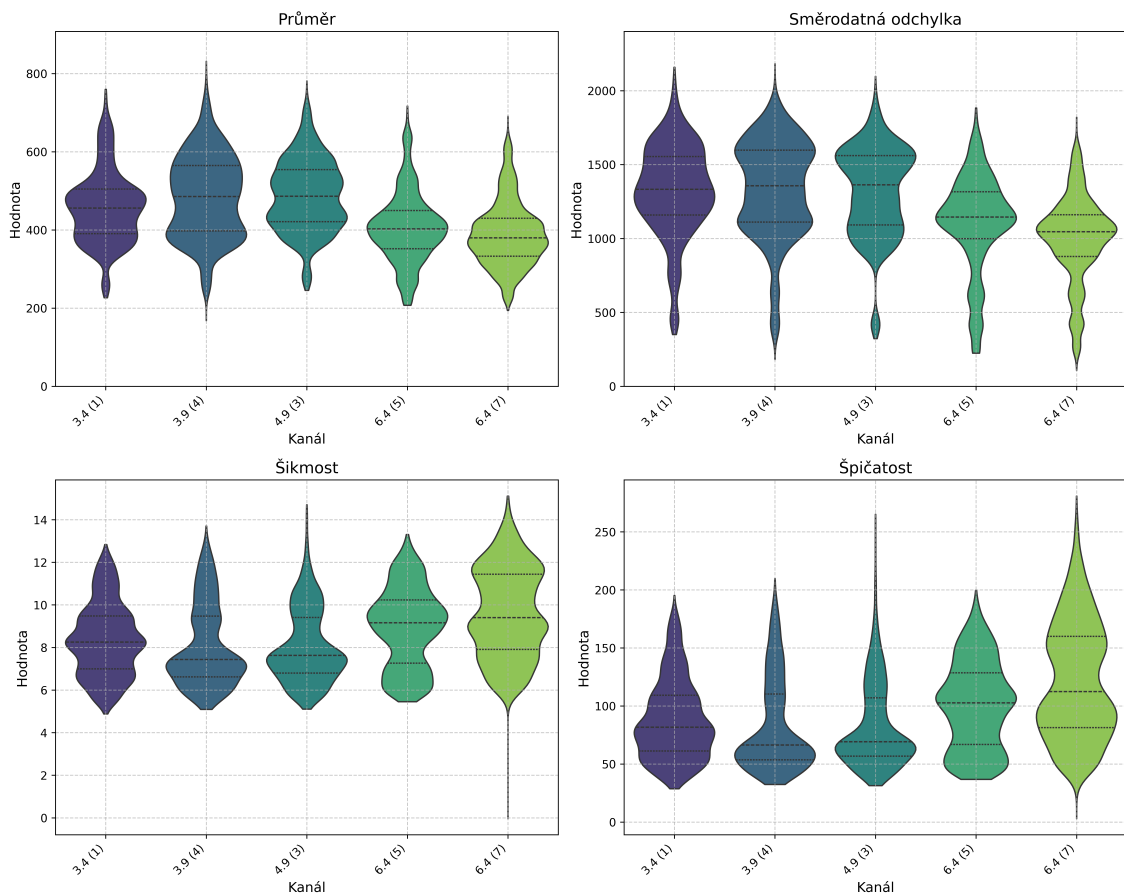
Obr. 4.7: Krabicový graf znázorňující rozdíly v extrahovaných vlastnostech napříč jednotlivými kanály.

Z krabicového grafu 4.7 byly identifikovány výrazné rozdíly mezi skupinami nižších (1, 3, 4) a vyšších kanálů (5, 7). Napříč všemi metrikami jsou hodnoty mediánů v rámci každé skupiny velmi blízké, avšak mezi skupinami byl patrný posun krabic v horizontálním směru, což indikuje systematické rozdíly v hodnotách těchto charakteristik mezi skupinami kanálů. Průměr a směrodatná odchylka vykazují menší variabilitu u kanálů 1, 5 a 7, zatímco u šikmosti a špičatosti je pozorována větší variabilita, zejména u kanálů 4, 5 a nejvýrazněji u kanálu 7, kde byly zaznamenány nejširší krabice. Rozdíly v polohách mediánů v rámci jednotlivých krabic jsou obecně malé a ve většině případů se medián nachází blízko středu krabice, což svědčí o symetrickém rozdělení. Výraznější asymetrie v polohách mediánů je pozorována u šikmosti a špičatosti – u kanálů 3 a 4 je medián výrazně blíže dolnímu kvartilu, zatímco u kanálu 5 naopak blíže hornímu kvartilu. U kanálů 1 a 7 je medián rovněž blíže dolnímu kvartilu, avšak jeho vzdálenost od středu krabice je podstatně nižší než u ostatních kanálů. Tyto rozdíly naznačují, že v rámci vyšších kanálů dochází k větší variabilitě a asymetrii v rozdělení extrahovaných charakteristik, především u šikmosti a špičatosti.

Violin graf 4.8 poskytuje detailní pohled na hustotu a tvar rozdělení jednotlivých charakteristik napříč kanály. U průměru a směrodatné odchylky lze pozorovat menší koncentraci hodnot s ostrým vrcholem, které odpovídají symetrickému rozložení, u kanálů 5 a 7. Naopak pro kanály 3 a 4 je pozorovatelná vysoká koncentrace hodnot v delším intervalu s plošším vrcholem, což značí vyšší variabilitu hodnot. Průměr pro kanál 1 se svým tvarem blíží spíše vyšším kanálům (5 a 7), zatímco směrodatná odchylka vykazuje podobnost spíše s kanály 3 a 4. U šikmosti a špičatosti je u kanálů 3 a 4 pozorována užší koncentrace hodnot s velmi ostrými vrcholy, což značí nižší variabilitu a vyšší četnost hodnot kolem mediánu. Naproti tomu u kanálů 5 a 7 je distribuce těchto charakteristik mnohem širší a vrcholy jsou plošší, což svědčí o vyšší variabilitě a přítomnosti vyšších hodnot šikmosti a špičatosti. Kanál 1 v těchto charakteristikách opět vykazuje tvar podobný spíše nižším kanálům, avšak s mírně širší koncentrací hodnot a plošší špičkou. Obecně lze konstatovat, že skupina kanálů 1, 3 a 4 vykazuje koncentraci u nižších hodnot šikmosti a špičatosti, zatímco kanály 5 a 7 mají silné zastoupení i u vyšších hodnot těchto charakteristik.

Porovnání rozdělení extrahovaných charakteristik s normálním rozdělením bylo realizováno pomocí Q-Q grafu 4.9. Z dílčích grafů vyplývá, že k normálovému rozložení se blíží pouze průměr u kanálů 3 a 4, kde body leží těsně podél diagonály. Ostatní kanály a všechny další charakteristiky vykazují charakteristický *S*-tvar, což indikuje přítomnost chvostů a odchylek od normality. Na základě polohy tohoto zakřivení vůči diagonále lze dle [83], [84] dovodit, že data mají nižší špičatost než normálové rozdělení, případně jsou přítomny odchylky v symetrii rozdělení (šikmost).

Distribuce extrahovaných charakteristik z CIR pro různé kanály



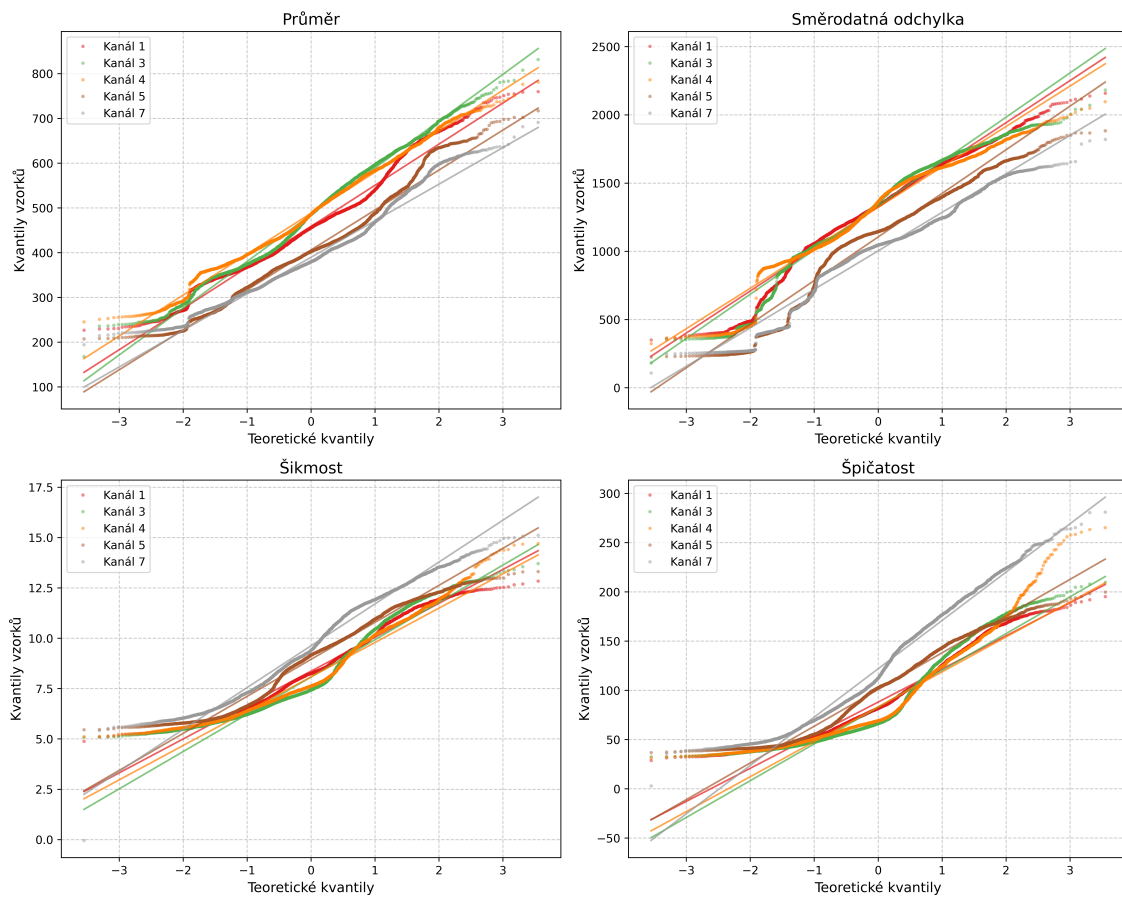
Obr. 4.8: Violin graf znázorňující rozdíly v extrahovaných vlastnostech napříč jednotlivými kanály.

Na základě grafických výstupů lze konstatovat, že mezi skupinami kanálů existují systematické rozdíly v rozdělení všech sledovaných charakteristik. Kanály 1, 3 a 4 vykazují stabilnější a symetričtější rozdělení s nižší variabilitou, zejména u šířkosti a špičatosti. Naopak kanály 5 a 7 jsou charakteristické vyšší variabilitou a častějším výskytem vyšších hodnot šířkosti a špičatosti.

4.6 Shrnutí

Z provedených experimentů vyplývá jednoznačná závislost použitého UWB kanálu na výkonnost jednotlivých klasifikátorů. Výsledky 1. scénáře potvrzují, že model není schopen stejně efektivně klasifikovat data z různých UWB kanálů. Vliv frekvence na přenositelnost modelu natrénovaného na jiné frekvenční doméně není zanedbatelný a nejvýraznější je zejména při přenosu mezi skupinami nižších (1, 3) a vyšších ka-

Q-Q grafy extrahovaných charakteristik z CIR pro různé kanály



Obr. 4.9: Q-Q graf znázorňující rozdíly v rozdělení extrahovaných vlastností napříč jednotlivými kanály.

nálů (5, 7). Nejobtížnější cílovou doménou jsou pro většinu modelů kanály 1 a 7, tedy nejnižší a nejvyšší kanál v dostupném datasetu. Nejuniverzálnějším tréninkovou doménou jsou kanály 1 a 4, tedy obecně kanály s nízkou frekvencí. Přítomnost širší sady frekvenčních domén v tréninku ve většině případů výrazně nesnižuje vliv frekvence na přenositelnost modelů, ale může mít vliv zejména na stabilitu modelu a prevenci přeučení. V posledním scénáři byly vizualizovány rozdíly mezi extrahovanými charakteristikami z CIR, které potvrzují rozdíl v distribuci dat mezi nižšími a vyššími kanály. Učiněné závěry jsou vztaženy k původu a povaze dostupných dat a pro data pocházející z jiného prostředí a zařízení se mohou lišit.

Závěr

V teoretické části byla analyzována problematika UWB komunikace, její potenciál v oblasti přesného měření vzdálenosti a lokalizace. Byly popsány základní lokalizační techniky, výhody a omezení UWB technologie, související standardy, důraz byl kladen na bezpečnostní aspekty a význam rozlišování mezi LoS a NLoS scénáři, včetně vlivu odrazů signálu na přesnost měření a důležitost detekce takového signálu. Dále byly představeny metody strojového učení vhodné pro klasifikaci signálů, včetně požadavků na vstupní data.

V praktické části byla provedena rešerše dostupných datových sad vhodných pro klasifikaci LoS a NLoS signálů na základě dat impulsní odezvy kanálu. Součástí práce bylo rovněž získání a zpracování neveřejné datové sady z UWB kanálu 9, která byla následně anotována dle poskytnutého klíče. Na těchto datech bylo realizováno podrobné ověření 6 publikovaných metod pro klasifikaci LoS a NLoS, a to prostřednictvím 4 experimentálních scénářů zaměřených na posouzení schopností modelů v různých aplikačních situacích. V posledním scénáři byla hodnocena generalizace modelu na různá vnitřní prostředí, včetně měření tréninkového času a paměťových nároků jednotlivých modelů. Výsledky ukazují, že metody založené na umělých neuronových sítích dosahují v tomto scénáři přesnosti nad 81 %, zatímco tradiční metody strojového učení vykazují přesnost kolem 74 %.

Bez detailního experimentálního ověření bylo dále popsáno 17 dalších publikovaných metod využívajících různé techniky strojového učení pro klasifikaci LoS a NLoS. U každé metody byla na základě dostupných informací popsána struktura, požadavky na vstupní data a schopnosti daného řešení. V případech, kdy to bylo možné, byla publikovaná architektura implementována a byly získány parametry charakterizující výpočetní a paměťovou náročnost, jako je počet parametrů modelu či inferenční čas. Bylo zjištěno, že metody založené na neuronových sítích obecně dosahují vyšší přesnosti, avšak za cenu vyšší výpočetní a paměťové náročnosti. Tradiční metody strojového učení jsou naopak často efektivnější z hlediska rychlosti a velikosti modelu. Významný vliv na výkonnost i přenositelnost modelů má rovněž povaha vstupních dat.

Analýza publikovaných řešení ukázala, že většina metod byla navržena pro data z konkrétního, nejčastěji laboratorního prostředí, a téměř všechny využívaly data získaná na různých kanálech prostřednictvím UWB modulu DMW 1000. Z výsledků experimentálního ověření nevyplývá, že by výkonnost klasifikátorů byla systematicky snížena na datech z určitého zařízení či kanálu.

Poslední část práce se věnuje analýze vlivu frekvence na přenositelnost modelů pro klasifikaci LoS a NLoS. Byly navrženy 4 experimentální scénáře, které byly realizovány s využitím 6 podrobně ověřovaných modelů. Experimenty prokázaly, že

výkonnost i přenositelnost klasifikátorů je závislá na použitém UWB kanálu. Vliv frekvence se mezi kanály liší, přičemž obecně platí, že se zvyšující se frekvencí klesá výkonnost modelu. Nejvýraznější pokles byl pozorován při přenosu mezi skupinami nízkofrekvenčních a vysokofrekvenčních kanálů, což souvisí i s rozdíly v distribuci extrahovaných charakteristik impulsní odezvy. Tyto poznatky představují základ pro formulaci budoucích hypotéz, avšak jejich zobecnění na jiná prostředí či zařízení vyžaduje další ověření, neboť v rámci experimentální části byl potvrzen významný vliv prostředí na výkonnost jednotlivých modelů. Pro data z jiného prostředí, získaná odlišným zařízením nebo metodikou, se mohou závěry o vlivu frekvence na přenositelnost lišit. Pro komplexní posouzení tohoto vlivu je nezbytné využít datové sady, které obsahují data z více frekvencí měřených ve stejném vnitřním prostředí. Dalším rozvojem výzkumu může být experimentování s vlivem různých tréninkových domén na přenositelnost modelů, aplikace generalizačních metod pro minimalizaci vlivu frekvence či realizace experimentů zaměřených na výběr souboru extrahovaných vlastností impulsní odezvy, které povedou ke zlepšení výkonnosti modelů napříč UWB kanály.

Literatura

- [1] COPPENS, D.; SHAHID, A.; LEMEY, S.; VAN HERBRUGGEN, B.; MARSHALL, C. et al. An Overview of UWB Standards and Organizations (IEEE 802.15.4, FiRa, Apple): Interoperability Aspects and Future Research Directions. Online. *IEEE Access*. 2022, č. 10, s. 70219-70241. Dostupné z: <https://doi.org/10.1109/ACCESS.2022.3187410>. [cit. 2024-10-02].
- [2] QU, Junwei. A review of UWB indoor positioning. Online. *Journal of Physics: Conference Series*. 2023, č. 2669, s. 012003. Dostupné z: <https://doi.org/10.1088/1742-6596/2669/1/012003>. [cit. 2024-10-06].
- [3] SINGH, M.; ROESCHLIN, M.; ZALZALA, E.; CAPKUN, C. a CAPKUN, S. Security Analysis of IEEE 802.15.4z/HRP UWB Time-of-Flight Distance Measurement. Online. *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 2021, č. WiSec '21, s. 227-237. Dostupné z: <https://doi.org/10.1145/3448300.3467831>. [cit. 2024-09-26].
- [4] KIM, Dae-Ho; FARHAD, Arshad a PYUN, Jae-Young. UWB Positioning System Based on LSTM Classification With Mitigated NLOS Effects. Online. *IEEE Internet of Things Journal*. 2023, roč. 10, č. 2, s. 1822-1835. ISSN 2327-4662. Dostupné z: <https://doi.org/10.1109/JIOT.2022.3209735>. [cit. 2024-11-10].
- [5] YANG, Xiaofei; WANG, Jun; SONG, Dapeng; FENG, Beizhen a YE, Hui. A Novel NLOS Error Compensation Method Based IMU for UWB Indoor Positioning System. Online. *IEEE Sensors Journal*. 2021, roč. 21, č. 9, s. 11203-11212. ISSN 1530-437X. Dostupné z: <https://doi.org/10.1109/JSEN.2021.3061468>. [cit. 2024-11-10].
- [6] JEŘÁBEK, Jan. Komunikační technologie. Brno: Vysoké učení technické v Brně, 2023. s. 1-175. ISBN: 978-80-214-4713-4. [cit. 2024-11-30].
- [7] SEDLÁČEK, P.; SLANINA, M. a MAŠEK, P. An Overview of the IEEE 802.15.4z Standard its Comparison and to the Existing UWB Standards. Online. *2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA)*. 2019, s. 1-6. Dostupné z: <https://doi.org/10.1109/RADIOELEK.2019.8733537>. [cit. 2024-10-08].
- [8] IEEE. 802.15.4z-2020 – IEEE Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques. Online. *IEEE Std 802.15.4z-2020 (Amendment*

- to *IEEE Std 802.15.4-2020*). 2020. Dostupné z: <https://doi.org/10.1109/IEEESTD.2020.9179124>. [cit. 2024-10-08].
- [9] UO, Xiliang; KALKANLI, Cem; ZHOU, Hao; ZHAN, Pengcheng a COHEN, Moche. Secure Ranging with IEEE 802.15.4z HRP UWB. Online. In: *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, s. 2794-2811. ISBN 979-8-3503-3130-1. Dostupné z: <https://doi.org/10.1109/SP54263.2024.00238>. [cit. 2025-02-09].
- [10] TM FiRa a Consortium. UWB Secure Ranging in FiRa. Online. In: *FiRa Consortium August 2022*. 2022. Dostupné z: https://www.firaconsortium.org/sites/default/files/2022-09/FIRA-Whitepaper-UWB-Secure-Ranging-August-2022_0.pdf. [cit. 2025-02-09].
- [11] ANLIKER, Claudio; CAMURATI, Giovanni a ČAPKUN, Srdjan. Time for Change: How Clocks Break UWB Secure Ranging. Online. In: *SEC '23: Proceedings of the 32nd USENIX Conference on Security Symposium*. Anaheim, CA, USA: USENIX Association, 2023. ISBN 978-1-939133-37-3. Dostupné z: <https://www.usenix.org/system/files/usenixsecurity23-anliker.pdf>. [cit. 2025-02-02].
- [12] WANG, Qiu; CHEN, Mingsong; LIU, Jiajie; LIN, Yongcheng; LI, Kai et al. 1D-CLANet: A Novel Network for NLoS Classification in UWB Indoor Positioning System. Online. *Applied Sciences*. 2024, roč. 14, č. 17. ISSN 2076-3417. Dostupné z: <https://doi.org/10.3390/app14177609>. [cit. 2024-11-10].
- [13] KAREDAL, Johan; WYNE, Shurjeel; ALMERS, Peter; TUFVESSION, Fredrik a MOLISH, Andreas F. Statistical analysis of the UWB channel in an industrial environment. Online. *EEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall*. 2004, roč. 2004, č. 1, s. 81-85. Dostupné z: <https://doi.org/10.1109/VETECF.2004.1399930>. [cit. 2025-03-21].
- [14] CHOLLET, Francois. *Deep Learning with Python*. Second Edition. Manning Publications, 2021. ISBN 9781617296864. [cit. 2024-10-15].
- [15] MUKHAMEDIEV, R. I.; POPOVA, Y.; KUCHYN, Y.; ZAITSEVA, E.; KALIMOLDAYEV, A. et al. Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges. *Mathematics*. 2022, č. 2552. Dostupné z: <https://doi.org/10.3390/math10152552>. [cit. 2024-10-17]

- [16] ZHANG, J. a YU, P. S. *Machine Learning Overview*. Online. Springer, Cham, 2019. ISBN 978-3-030-12528-8 Dostupné z: https://doi.org/10.1007/978-3-030-12528-8_2. [cit. 2024-10-12]
- [17] LEE, J. H.; SHIN, J. a REALFF, M. J. Machine learning: Overview of the recent progresses and implications for the process systems engineering field. Online. In: *Computers & Chemical Engineering 2018*, s. 111-121. Dostupné z: <https://doi.org/10.1016/j.compchemeng.2017.10.008>. [cit. 2024-10-12]
- [18] BLACK, Jason E.; KUEPER, J. K. a WILLIAMSON, Tyler S. An introduction to machine learning for classification and prediction. Online. *Family Practice*. 2022, č. 40, s. 200-204. Dostupné z: <https://doi.org/10.1093/fampra/cmab104>. [cit. 2024-10-17].
- [19] BADILLO, S.; BÁNFAI, B.; DAVYDOV, I. I.; HUTCHINSON, L.; KAMTHONG, T. et al. An Introduction to Machine Learning. Online. *Clinical Pharmacology and Therapeutics*. 2020, č. 107, s. 871-885. Dostupné z: <https://doi.org/10.1002/cpt.1796>. [cit. 2024-10-15].
- [20] CERVANTES, Jair; GARCIA-LAMONT, Farid; RODRÍGUEZ-MAZAHUA, Lisbeth a LOPEZ, Asdrubal. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Online. *Neurocomputing*. Č. 408, s. 189-215. Dostupné z: <https://doi.org/10.1016/j.neucom.2019.10.118>. [cit. 2024-10-20].
- [21] MOHAIMINUL, Islam; GUORONG, Chen a JIN, Shangzhu. An overview of neural network. Online. *American Journal of Neural Networks and Applications*. 2019, č. 5, s. 7-11. Dostupné z: <https://www.sciencepublishinggroup.com/article/10.11648/j.ajnna.20190501.12>. [cit. 2024-10-20].
- [22] SHARMA, Siddharth; SHARMA, Simone a ATHAIYA, Anidhya. Activation functions in neural networks. Online. *International Journal of Engineering Applied Sciences and Technology*. 2020. Dostupné z: <https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf>. [cit. 2024-10-25].
- [23] ZOU, Yiling. Vanishing Gradient Problem: An Activation Function that Seeks to Improve VGP. Online. *Advances in Engineering Technology Research*. 2024, roč. 11, č. 1. ISSN 2790-1688. Dostupné z: <https://doi.org/10.56028/aetr.11.1.501.2024>. [cit. 2024-11-30].
- [24] ALBAWI, Saad; MOHAMMED, Tareq Abed a AL-ZAWI, Saad. Understanding of a convolutional neural network. Online. *2017 International Conference on*

- Engineering and Technology (ICET)*. 2017, s. 1-6. Dostupné z: <https://doi.org/10.1109/ICEngTechnol.2017.8308186>. [cit. 2024-10-25].
- [25] *The Importance of Balanced Accuracy in Imbalanced Class Classification Models*. Online. In: *Adventures in Machine Learning*. Dostupné z: <http://bit.ly/3H1GfGp>. [cit. 2025-05-05].
- [26] CAI, Dongting. Enhancing capabilities of generative models through VAE-GAN integration: A review. Online. *Applied and Computational Engineering*. 2024, roč. 67, č. 1, s. 239-246. ISSN 2755-2721. Dostupné z: <https://doi.org/10.54254/2755-2721/67/2024MA0070>. [cit. 2025-04-06].
- [27] SINGH, Aman a OGUNFUNMI, Tokunbo. An Overview of Variational Auto-encoders for Source Separation, Finance, and Bio-Signal Applications. Online. *Entropy*. 2022, roč. 24, č. 1. ISSN 1099-4300. Dostupné z: <https://doi.org/10.3390/e24010055>. [cit. 2025-04-06].
- [28] KARAHODA, Bertan. Generating Time Series Data With Real-Valued DC-GAN From Complex Time-Frequency Domain: Application to ECG Synthesis. Online. *IEEE Access*. 2024, roč. 12, s. 143215-143225. ISSN 2169-3536. Dostupné z: <https://doi.org/10.1109/ACCESS.2024.3469541>. [cit. 2025-04-06].
- [29] AKROUT, Mohamed; FERIANI, Amal; BELLILI, Faouzi; MEZGHANI, Amine a HOSSAIN, Ekram. Domain Generalization in Machine Learning Models for Wireless Communications: Concepts, State-of-the-Art, and Open Issues. Online. *IEEE Communications Surveys & Tutorials*. 2023, roč. 25, č. 4, s. 3014-3037. ISSN 1553-877X. Dostupné z: <https://doi.org/10.1109/COMST.2023.3326399>. [cit. 2025-02-21].
- [30] ZHOU, Kaiyang; LIU, Ziwei; QIAO, Yu; XIANG, Tao a LOY, Chen Change. Domain Generalization: A Survey. Online. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. S. 1-20. ISSN 0162-8828. Dostupné z: <https://doi.org/10.1109/TPAMI.2022.3195549>. [cit. 2025-02-21].
- [31] LI, Da; YANG, Yongxin; SONG, Yi-Zhe a HOSPEDALES, Timothy. Learning to Generalize: Meta-Learning for Domain Generalization. Online. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1). 2018. Dostupné z: <https://doi.org/https://doi.org/10.1609/aaai.v32i1.11596>. [cit. 2025-02-21].
- [32] Tensorflow 2.10.1. Tensorflow. Online. Dostupné z: <https://doi.org/10.5281/zenodo.7604132>. [cit. 2025-04-28].

- [33] PEDREGOSA, Fabian; VAROQUAUX, Gaël; GRAMFORT, Alexandre; MICHEL, Vincent; THIRION, Bertrand et al. Scikit-learn: Machine Learning in Python. Online. *Journal of Machine Learning Research*. 2011, č. 12, s. 2825-2830. Dostupné z: <https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>. [cit. 2025-05-04].
- [34] BREGAR, Klemen a MOHORČIČ, Mihael. Improving Indoor Localization Using Convolutional Neural Networks on Computationally Restricted Devices. Online. *IEEE Access*. 2018, č. 6, s. 17429-17441. Dostupné z: <https://doi.org/10.1109/ACCESS.2018.2817800>. [cit. 2024-10-25].
- [35] LIU, Qingzhi; YIN, Zhendong; ZHAO, Yanlong; WU, Zhilu a WU, Mingyang. UWB LOS/NLOS identification in multiple indoor environments using deep learning methods. Online. *Physical Communication*. 2022, roč. 52. ISSN 18744907. Dostupné z: <https://doi.org/10.1016/j.phycom.2022.101695>. [cit. 2024-11-15].
- [36] TENGLI, Nandini. *All About Efficiency Metrics in ML*. Online. In: MEDIUM. Towards AI. 2024. Dostupné z: <https://pub.towardsai.net/all-about-the-efficiency-metrics-for-ml-e7fb956ade77>. [cit. 2025-05-04].
- [37] Pstutil 7.0.0. Python community. Online. 2025. Dostupné z: <https://pypi.org/project/psutil/>. [cit. 2025-04-28].
- [38] Tracemalloc — Trace memory allocations. Python documentation. Online. Dostupné z: <https://docs.python.org/3/library/tracemalloc.html>. [cit. 2025-04-28].
- [39] BREGAR, Klemen a MOHORČIČ, Mihael. UWB LOS and NLOS Data Set. Online. 2018. Dostupné z: <https://github.com/ewine-project/UWB-localization/tree/master/data/NLOSClassification>. [cit. 2024-10-27].
- [40] EMBER, Elisei; PESTANA, Jesus; KRISPER, Michael; STOCKER, Michael; RÖMER, Kay et al. Impact of Feature Selection and CIR Window Length on NLoS Classification for UWB Systems. Online. *2023 19th International Conference on Mobility, Sensing and Networking (MSN)*. 2023, s. 72-80. Dostupné z: <https://doi.org/10.1109/MSN60784.2023.00024>. [cit. 2024-10-25].
- [41] BREGAR, Klemen a MOHORČIČ, Mihael. UWB Localization Data Set. Online. 2018. Dostupné z: <https://github.com/ewine-project/UWB-localization/tree/master/data/localization>. [cit. 2024-10-27].

- [42] STOCKER, Michael; GALLACHER, Markus; BOANO, Carlo Alberto a RÖMER, Kay. Dataset UWB-NLoS cpsiot 2021 Stocker. Online. 2021. Dostupné z: <https://cloud.tugraz.at/index.php/s/QwCfmaYBrf55Ksg>. [cit. 2024-10-30].
- [43] STOCKER, Michael; GALLACHER, Markus; BOANO, Carlo Alberto a RÖMER, Kay. Performance of support vector regression in correcting UWB ranging measurements under LOS/NLOS conditions. Online. *Proceedings of the Workshop on Benchmarking Cyber-Physical Systems and Internet of Things. 2021, č. CPS-IoTBench '21, s. 6-11*. Dostupné z: <https://doi.org/10.1145/3458473.3458820>. [cit. 2024-10-25].
- [44] FLUERATORU, Laura; LOHAN, Elena Simona a NICULESCU, Dragos. Dataset: Ultra-Wideband Ranging Measurements Acquired With Three Different Platforms (Qorvo, TDSR, 3db Access). Online. 2022. Dostupné z: <https://doi.org/10.5281/zenodo.6984698>. [cit. 2024-10-30].
- [45] FLUERATORU, Laura; LOHAN, Elena Simona a NICULESCU, Dragos. Challenges in platform-independent UWB ranging and localization systems. Online. *Proceedings of the 16th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization. 2022, č. WiNTECH '22, s. 9-15*. Dostupné z: <https://doi.org/10.1145/3556564.3558238>. [cit. 2024-10-27].
- [46] *DWM3000: 6.5 & 8.0 GHz Ultra-Wideband (UWB) Module*. Online. In: Qorvo. Dostupné z: <https://www.qorvo.com/products/p/DWM3000>. [cit. 2024-10-30].
- [47] *An Overview of the IEEE 802.15.4 HRP UWB Standard*. Online. In: KEYSIGHT. RF + Microwave. 2021, 2021-07-28. Dostupné z: <https://www.keysight.com/blogs/en/tech/rfmw/2021/07/28/an-overview-of-the-ieee-802154-hrp-uwband-standard>. [cit. 2024-10-30].
- [48] *Technology: 3DB6830*. Online. In: 3db-access. Dostupné z: <https://www.3db-access.com/technology>. [cit. 2024-10-30].
- [49] FONTAINE, Jaron and SHAHID, Adnan and VAN HERBRUGGEN, Ben. Industrial UWB localization dataset containing Channel Impulse Response data (CIRs). Online. *IEEE Dataport*. 2023. Dostupné z: <https://doi.org/10.21227/w0v9-5e35>. [cit. 2024-10-30].
- [50] FONTAINE, Jaron; RIDOLFI, Matteo; VAN HERBRUGGEN, Ben; SHAHID, Adnad a DE POORTER, Elie. Edge Inference for UWB Ranging Error

- Correction Using Autoencoders. Online. *IEEE Access*. 2020, č. 8, s. 139143-139155. Dostupné z: <https://doi.org/10.1109/ACCESS.2020.3012822>. [cit. 2024-10-27].
- [51] BARRAL, Valentín. Pozyx CIR and range with LOS and NLOS. Online. *IEEE Dataport*. 2020. Dostupné z: <https://doi.org/10.21227/sr92-6s06>. [cit. 2024-10-30].
- [52] BARRAL, Valentín; ESCUDERO, Carlos J.; GARCÍA-NAYA, José A. a MANEIRO-CATOIRA, Roberto. NLOS Identification and Mitigation Using Low-Cost UWB Devices. Online. *Sensors*. 2019, č. 19. Dostupné z: <https://doi.org/10.3390/s19163464>. [cit. 2024-10-27].
- [53] *UWB Settings*. Online. In: Pozyx Knowledge Center. Dostupné z: <https://docs.pozyx.io/enterprise/uwb>. [cit. 2024-10-27].
- [54] JIANG, Changhui; SHEN, Jichun; CHEN, Shuai; CHEN, Yuwei; LIU, Di et al. UWB NLOS/LOS Classification Using Deep Learning Method. Online. *IEEE Communications Letters*. 2020, roč. 24, č. 10, s. 2226-2230. ISSN 1089-7798. Dostupné z: <https://doi.org/10.1109/LCOMM.2020.2999904>. [cit. 2024-11-10].
- [55] GHAEMIFAR, Mohammadali; EBADOLLAHI, Saeed; GHASEMZADEH, Milad a PIRAHMADIAN, Saba. A Novel CNN Model for NLoS Classification in UWB Indoor Positioning System. Online. In: *2024 10th International Conference on Web Research (ICWR)*. IEEE, 2024, s. 182-186. ISBN 979-8-3503-9498-6. Dostupné z: <https://doi.org/10.1109/ICWR61162.2024.10533361>. [cit. 2024-11-15].
- [56] *Epoch in Machine Learning* Online. In: Geeks for Geeks. 2024-11-02. Dostupné z: <https://www.geeksforgeeks.org/epoch-in-machine-learning/>. [cit. 2024-11-29].
- [57] IOFFE, Sergey a SZEGEDY, Christian. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Online. 2015. Dostupné z: <https://doi.org/10.48550/arXiv.1502.03167>. [cit. 2024-11-29].
- [58] *Optimization Algorithms in Machine Learning* Online. In: Geeks for Geeks. 2024-05-28. Dostupné z: <https://www.geeksforgeeks.org/optimization-algorithms-in-machine-learning/>. [cit. 2024-11-29].
- [59] *What is Adam Optimizer?* Online. In: Geeks for Geeks. 2024-04-20. Dostupné z: <https://www.geeksforgeeks.org/adam-optimizer/>. [cit. 2024-11-29].

- [60] STAHLKE, Maximilian; KRAM, Sebastian; MUTSCHLER, Christopher a MAHR, Thomas. NLOS Detection using UWB Channel Impulse Responses and Convolutional Neural Networks. Online. In: *2020 International Conference on Localization and GNSS (ICL-GNSS)*. IEEE, 2020, s. 1-6. ISBN 978-1-7281-6455-7. Dostupné z: <https://doi.org/10.1109/ICL-GNSS49876.2020.9115498>. [cit. 2025-04-11].
- [61] PARK, JiWoong; NAM, SungChan; CHOI, HongBeom; KO, YoungEun a KO, Young-Bae. Improving Deep Learning-Based UWB LOS/NLOS Identification with Transfer Learning: An Empirical Approach. Online. *Electronics*. 2020, roč. 9, č. 10. ISSN 2079-9292. Dostupné z: <https://doi.org/10.3390/electronics9101714>. [cit. 2024-11-15].
- [62] KRISTENSEN, Jeppe Bro; MASSANET GINARD, Michel; JENSEN, Ole Kiel a SHEN, Ming. Non-Line-of-Sight Identification for UWB Indoor Positioning Systems using Support Vector Machines. Online. In: *2019 IEEE MTT-S International Wireless Symposium (IWS)*. IEEE, 2019, s. 1-3. ISBN 978-1-7281-0716-5. Dostupné z: <https://doi.org/10.1109/IEEE-IWS.2019.8804072>. [cit. 2024-11-15].
- [63] RAMADAN, Mohammed and SARK, Vladica and GUTIERREZ, Jesus and GRASS, Eckhard. NLOS Identification for Indoor Localization using Random Forest Algorithm. Online. In: *WSA 2018; 22nd International ITG Workshop on Smart Antennas*. 2018, s. 1-5. Dostupné z: <https://ieeexplore.ieee.org/document/8385494>. [cit. 2025-04-21].
- [64] NKROW, Raphael Elikplim; SILVA, Bruno; BOSHOFF, Dutliff; HANCKE, Gerhard; GIDLUND, Mikael et al. NLOS Identification and Mitigation for Time-based Indoor Localization Systems: Survey and Future Research Directions. Online. *ACM Computing Surveys*. 2024, roč. 56, č. 12, s. 1-41. ISSN 0360-0300. Dostupné z: <https://doi.org/10.1145/3663473>. [cit. 2024-11-10].
- [65] ZHAO, Yuxuan a WANG, Manyi. The LOS/NLOS Classification Method Based on Deep Learning for the UWB Localization System in Coal Mines. Online. *Applied Sciences*. 2022, roč. 12, č. 13. ISSN 2076-3417. Dostupné z: <https://doi.org/10.3390/app12136484>. [cit. 2024-11-10].
- [66] WANG, Fei; XU, Zhan; ZHI, Ruxin; CHEN, Jinhui a ZHANG, Peiyue. LOS/NLOS Channel Identification Technology Based on CNN. Online. In: *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*. IEEE,

- 2019, s. 200-203. ISBN 978-1-7281-5163-2. Dostupné z: <https://doi.org/10.1109/NICS48868.2019.9023805>. [cit. 2025-04-11].
- [67] LEIBER, Maxime; MARNISSI, Yosra a BARRAU, Axel. Differentiable Short-Time Fourier Transform: A Time-Frequency Layer with Learnable Parameters. Online. *ICML 2024 Workshop, Vienna, Austria*. 2024, roč. 2024. Dostupné z: https://differentiable.xyz/papers-2024/paper_02.pdf. [cit. 2025-04-28].
- [68] CUI, Zhichao; GAO, Yufang; HU, Jing; TIAN, Shiwei a CHENG, Jian. LOS/NLOS Identification for Indoor UWB Positioning Based on Morlet Wavelet Transform and Convolutional Neural Networks. Online. *IEEE Communications Letters*. 2021, roč. 25, č. 3, s. 879-882. ISSN 1089-7798. Dostupné z: <https://doi.org/10.1109/LCOMM.2020.3039251>. [cit. 2024-11-10].
- [69] LEE, Gregory; GOMMERS, Ralf; WASELEWSKI, Filip; WOHLFAHRT, Kai a O'LEARY, Aaron. PyWavelets: A Python package for wavelet analysis. Online. *Journal of Open Source Software*. 2019, roč. 4, č. 36. ISSN 2475-9066. Dostupné z: <https://doi.org/10.21105/joss.01237>. [cit. 2025-04-28].
- [70] ZHANG, Kexin; XI, Zhihong a ZHENG, Fenghao. UWB Indoor Positioning System Based on NLOS Recognition. Online. In: *2023 IEEE 6th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*. IEEE, 2023, s. 376-379. ISBN 979-8-3503-0562-3. Dostupné z: <https://doi.org/10.1109/AUTEEE60196.2023.10407534>. [cit. 2024-11-28].
- [71] CHANG, Tiantian; JIANG, Suying; SUN, Yuzhe; JIA, Ailin a WANG, Wei. Multi-bandwidth NLOS Identification Based on Deep Learning Method. Online. In: *2021 15th European Conference on Antennas and Propagation (EuCAP)*. IEEE, 2021, s. 1-5. ISBN 978-88-31299-02-2. Dostupné z: <https://doi.org/10.23919/EuCAP51087.2021.9411236>. [cit. 2025-04-13].
- [72] YANG, Hongchao; WANG, Yunjia; SEOW, Chee Kiat; SUN, Meng; SI, Minghao et al. UWB Sensor-Based Indoor LOS/NLOS Localization With Support Vector Machine Learning. Online. *IEEE Sensors Journal*. 2023, roč. 23, č. 3, s. 2988-3004. ISSN 1530-437X. Dostupné z: <https://doi.org/10.1109/JSEN.2022.3232479>. [cit. 2024-11-28].
- [73] CWALINA, Krzysztof K.; RAJCHOWSKI, Piotr; BLASZKIEWICZ, Olga; OLEJNICZAK, Alicja a SADOWSKI, Jaroslaw. Deep Learning-Based LOS and NLOS Identification in Wireless Body Area Networks. Online. *Sensors*.

2019, roč. 19, č. 19. ISSN 1424-8220. Dostupné z: <https://doi.org/10.3390/s19194229>. [cit. 2025-04-11].

- [74] FERREIRA, Andre G.; FERNANDES, Duarte; BRANCO, Sergio; CATA-RINO, Andre Paulo a MONTEIRO, Joao L. Feature Selection for Real-Time NLOS Identification and Mitigation for Body-Mounted UWB Transceivers. Online. *IEEE Transactions on Instrumentation and Measurement*. 2021, roč. 70, s. 1-10. ISSN 0018-9456. Dostupné z: <https://doi.org/10.1109/TIM.2021.3070619>. [cit. 2025-04-20].
- [75] BREGAR, Klemen and HROVAT, Andrej and MOHORCIC, Mihael. NLOS Channel Detection with Multilayer Perceptron in Low-Rate Personal Area Networks for Indoor Localization Accuracy Improvement. Online. 2016. Dostupné z: https://www.researchgate.net/publication/308986067_NLOS_Channel_Detection_with_Multilayer_Perceptron_in_Low-Rate_Personal_Area_Networks_for_Indoor_Localization_Accuracy_Improvement. [cit. 2025-04-20].
- [76] POULOSE, Alwin a HAN, Dong Seog. UWB Indoor Localization Using Deep Learning LSTM Networks. Online. *Applied Sciences*. 2020, roč. 10, č. 18. ISSN 2076-3417. Dostupné z: <https://doi.org/10.3390/app10186290>. [cit. 2024-11-28].
- [77] BUDÍKOVÁ, Marie; KRÁLOVÁ, Maria a MAROŠ, Bohumil. *Průvodce základními statistickými metodami*. Expert. Praha: Grada Publishing, 2010. ISBN 978-80-247-3243-5.
- [78] SHAPIRO, S. S. a WILK, M. B. An Analysis of Variance Test for Normality (Complete Samples). Online. *Biometrika*. 1965, roč. 52, č. 3/4, s. 591-611. ISSN 00063444. Dostupné z: <https://doi.org/10.2307/2333709>. [cit. 2025-05-05].
- [79] *Levene Test for Equality of Variances*. Online. In: NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. NIST. Dostupné z: <https://www.itl.nist.gov/div898/handbook/eda/section3/eda35a.htm>. [cit. 2025-05-05].
- [80] DALGAARD, Peter. Analysis of variance and the Kruskal–Wallis test. Online. In: *Introductory Statistics with R. Statistics and Computing*. New York, NY: Springer New York, 2008, s. 127-143. ISBN 978-0-387-79053-4. Dostupné z: https://doi.org/10.1007/978-0-387-79054-1_7. [cit. 2025-05-05].

- [81] TUKEY, John W. Comparing Individual Means in the Analysis of Variance. Online. *Biometrics*. 1949, roč. 5, č. 2, s. 99–114. ISSN 0006341X. Dostupné z: <https://doi.org/10.2307/3001913>. [cit. 2025-05-05].
- [82] DINNO, Alexis. Nonparametric Pairwise Multiple Comparisons in Independent Groups using Dunn’s Test. Online. *The Stata Journal: Promoting communications on statistics and Stata*. 2015, roč. 15, č. 1, s. 292-300. ISSN 1536-867X. Dostupné z: <https://doi.org/10.1177/1536867X1501500117>. [cit. 2025-05-05].
- [83] FIELD, Andy P. *Discovering statistics using SPSS: (and sex and drugs and rock ’n’ roll)*. 3rd ed. Los Angeles: Sage, 2009. ISBN 978-1-4129-7752-4.
- [84] *Jak se pozná normalita pomocí grafů?* 2013. Online. In: STATSOFT. StatSoft. Dostupné z: http://www.statsoft.cz/file1/PDF/newsletter/2013_10_09_StatSoft_Jak_se_pozna_normalita_pomoci_grafu.pdf. [cit. 2025-05-13].
- [85] WASKOM, Michael. Seaborn: statistical data visualization. Online. *Journal of Open Source Software*. 2021, roč. 6, č. 60. ISSN 2475-9066. Dostupné z: <https://doi.org/10.21105/joss.03021>. [cit. 2025-05-17].

Seznam symbolů a zkratek

Adam	Adaptive Moment Estimation – optimalizační algoritmus
AI	Artificial Intelligence – umělá inteligence
AoA	Angle of Arrival – úhel příchodu signálu
BPM	Burst Position Modulation – modulace polohy shluku pulzů
BPSK	Binary Phase Shift Keying – binární fázové klíčování
CCC	Car Connectivity Consortium
CIR	Channel Impulse Response – impulsní odezva kanálu
cm	Centimetr – jednotka délky
CNN	Convolutional Neural Network – konvoluční neuronová síť
CSML	Common Service Management Layer – společná vrstva pro služby a správu
CSI	Channel State Information – komplexní frekvenční odezva kanálu
DA	Domain Adaptation – adaptace domény
dBm	Decibel miliWatt – jednotka pro vyjádření výkonu
dBm/MHz	Decibely vzhledem k miliwattu na megahertz – jednotka pro vyjádření spektrální hustoty výkonu signálu
DFNN	Deep Feedforward Neural Network – dopředná hluboká neuronová síť
DS-TWR	Double-Sided Two Way Ranging – oboustranný dvoucestný systém lokalizace
FDE	First Path Distance Error – chyba vzdálenosti první cesty
FN	False Negative – počet nesprávně klasifikovaných pozitivních vzorků
FP	False Positive – počet správně klasifikovaných negativních vzorků
FPE	First Path Error – chyba první cesty
FPP	First Path Power – výkon první cesty
FPR	False Positive Rate – poměr správně předpovězených negativních hodnot vůči celkovému počtu negativních hodnot

FCN	False Crests Number – počet falešných vrcholů
GAN	Generative Adversarial Network – generativní adversariální síť
GHz	GigaHertz — jednotka frekvence
GRU	Gated Recurrent Unit – jednotka rekurentní brány
HRP	High Rate Pulse – vysokorychlostní režim fyzické vrstvy UWB
CSI	Inverse Discrete Fourier Transform – inverzní diskrétní Fourierova transformace
IEEE	Institute of Electrical and Electronics Engineers
IR-UWB	Impulse Radio UWB – technologie vycházející z UWB
ISO/OSI	Open Systems Interconnection mode – výchozí model komunikační architektury pro realizaci veřejných datových sítí
LDA	Linear Discriminant Analysis – lineární diskriminační analýza
LogReg	Logistic Regression – logistická regrese
LoS	Line of Sight – přímá viditelnost
LSTM	Long Short-Term Memory
LRP	Low Rate Pulse – nízkorychlostní režim fyzické vrstvy UWB
MAC	Medium Access Control Layer – Linková vrstva
MAML	Model Agnostic Meta Learning – jedna z technik meta učení
MHz	MegaHertz — jednotka frekvence
MFi	Made For iPhone – program společnosti Apple pro výrobce, kteří chtějí vyvíjet zařízení kompatibilní s Apple UWB čipem
ML	Machine Learning – strojové učení
MLP	Multilayer Perceptron – vícevrstvý perceptron
mm	Milimetr – jednotka délky
MPEM	Maximal Peak to Earlier Peak ratio
MWT	Morlet Wave Transform – Morletova vlnková transformace

NLoS	Non Line of Sight – nepřímá viditelnost
ns	Nanosekunda – jednotka času
OOK	On Off Keying – binární modulace
OWR	One Way Ranging – jednocestný systém lokalizace
PAPR	Peak to Average Power Ratio – poměr špičkového výkonu vůči průměrnému výkonu
PBFSK	Pulsed Binary Frequency Shift Keying – pulzní binární kmitočtové klíčování
PHY	Physical Layer – Fyzická vrstva
PPM	Pulse Position Modulation – pulzně polohová modulace
PRF	Pulse Repetition Frequency – opakovací frekvence pulsů
ReLu	REctified Linear Unit – usměrněná lineární funkce
RF	Random Forest – algoritmus náhodného lesa
RNN	Recurrent Neural Network – rekurentní neuronová síť
ROC	Receiver Operating Characteristic – křivka znázorňující závislost TPR na FPR
RSSI	Received Signal Strength Indicator – síla přijímaného signálu
RTLS	Real-Time Location System – systém pro lokalizaci v reálném čase
RTT	Round Trip Time – čas, který uplyne od odeslání signálu do přijetí odpovědi
SE	Squeeze and Excitation – bloky stlačení a vzrušení
SNR	Signal to Noise Ratio – odstup signálu od šumu
STS	Secure Training Sequences – bezpečné trénovací sekvence
STFT	Short-Time Fourier Transform – krátkodobá Fourierova transformace
SVM	Support Vector Machine – metoda podpůrných vektorů
TDoA	Time Difference of Arrival – časový rozdíl příchodu signálu
TL	Transfer Learning – přenesené učení

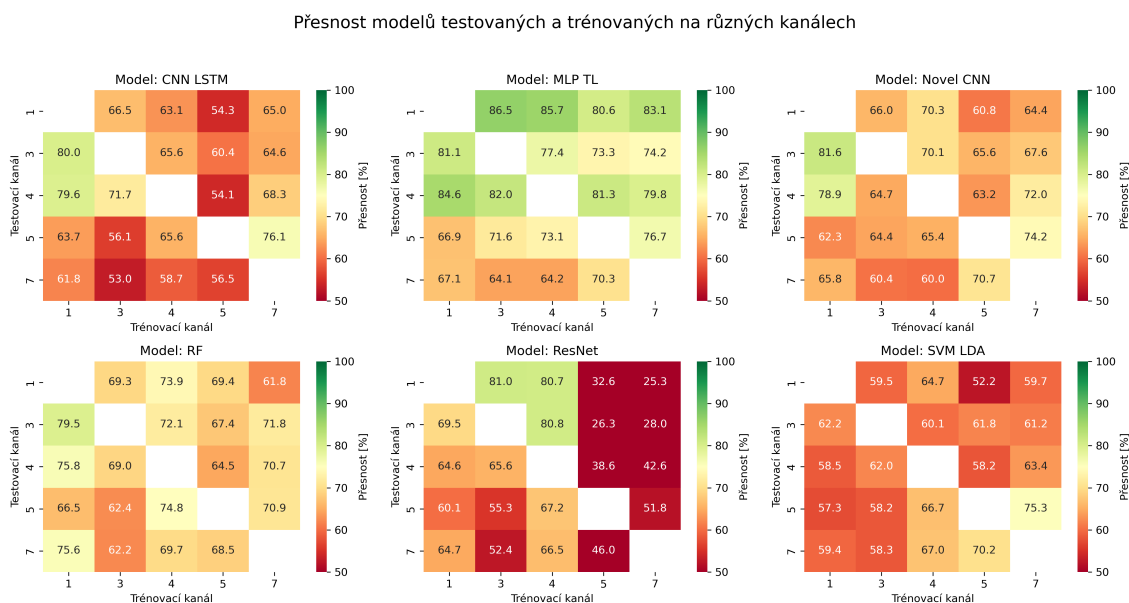
TN	True Negative – počet správně klasifikovaných negativních vzorků
ToA	Time of Arrival – čas přijetí signálu
ToF	Time of Flight – doba letu signálu
TP	True Positive – počet správně klasifikovaných pozitivních vzorků
TP	Total Power – celkový výkon
TPR	True Positive Rate – poměr správně předpovězených pozitivních hodnot vůči celkovému počtu pozitivních hodnot
TWR	Two Way Ranging – dvoucestný systém lokalizace
UWB	Ultra-WideBand – rádiová komunikace s velkou šířkou pásma
VAE	Variational Autoencoder – variační autoenkodéry
WL	Window Length – velikost (časového) okna
ZL	Zero-shot Learning – učení s nulovým počtem snímků

A Vybrané výstupy scénářů z kapitoly 4

V této části dokumentu jsou prezentovány vybrané výstupy scénářů z kapitoly 4, která se zabývá vlivem frekvence na přenositelnost modelů pro klasifikaci LOS a NLoS. Výčet výstupů obsažených v této kapitole je vždy prezentován v rámci konkrétního scénáře. Jedná se vždy o získané hodnoty sledovaných metrik ve formě tabulky či heatmapy a rozdíly v těchto metrikách v kontextu jednotlivých scénářů. Další typy výstupů jsou součástí elektronické přílohy této práce, která je popsána v části B.

A.1 Scénář 4.3

Pro 2. scénář, popsáný v kapitole 4.3 a v rámci kterého je testována výkonnost modelu na všech kanálech různých od trénovacího kanálu, jsou součástí přílohy heatmapy s hodnotami sledovaných metrik a tabulka i heatmapy vyjadřující rozdíl ve výkonnosti v porovnání s 1. scénářem, v rámci kterého byl model trénován a testován na stejném kanálu.

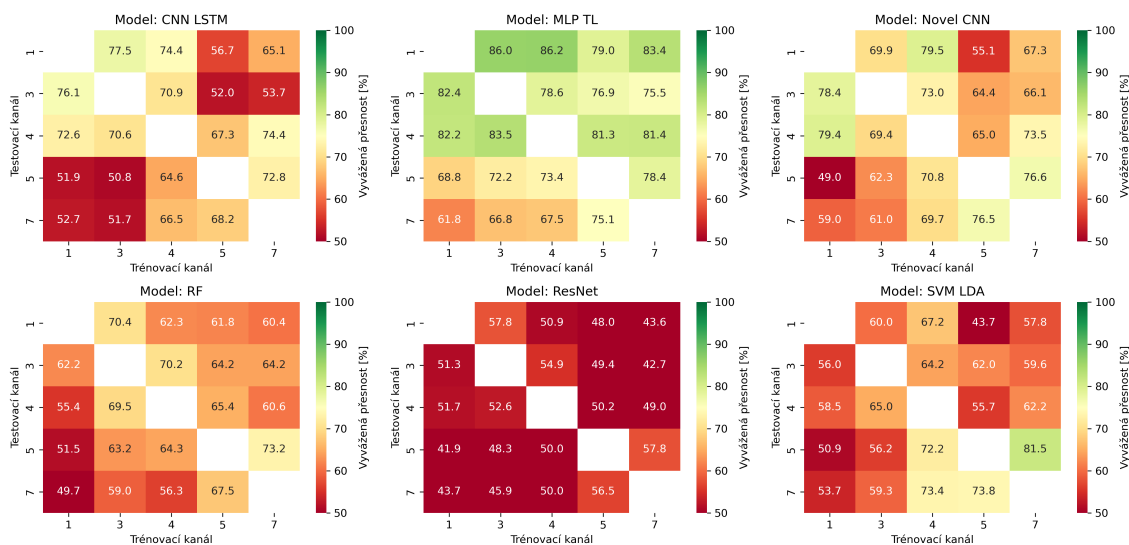


Obr. A.1: Heatmapa s hodnotami metriky „přesnost“ pro modely trénované a testované na různých kanálech.

		Přesnost [%]																		
Testovací kanál	Trénovací kanál - Model CNN LSTM						Testovací kanál	Trénovací kanál - Model MLP TL						Testovací kanál	Trénovací kanál - Model Novel CNN					
	1	3	4	5	7	Ø		1	3	4	5	7	Ø		1	3	4	5	7	Ø
1	-	18,6	22,0	30,7	20,1	22,9	1	-	7,7	8,5	13,6	11,1	10,2	1	-	32,1	27,9	37,3	33,8	32,8
3	0,9	-	15,3	20,5	16,3	13,3	3	12,4	-	16,1	20,2	19,3	17,0	3	15,1	-	26,5	31,1	29,1	25,4
4	7,9	15,8	-	33,4	19,2	19,1	4	9,7	12,3	-	13,0	14,4	12,3	4	19,0	33,2	-	34,7	25,9	28,2
5	12,0	19,6	10,1	-	-0,4	10,3	5	21,0	16,3	14,8	-	11,2	15,8	5	31,7	29,6	28,6	-	19,9	27,5
7	13,8	22,6	17,0	19,2	-	18,1	7	18,4	21,3	21,2	15,1	-	19,0	7	28,3	33,7	34,1	23,4	-	29,9
Ø	8,6	19,1	16,1	26,0	13,8	-	Ø	15,4	14,4	15,2	15,5	14,0	-	Ø	23,5	32,2	29,3	31,6	27,2	-
Testovací kanál	Trénovací kanál - Model RF						Testovací kanál	Trénovací kanál - Model Resnet						Testovací kanál	Trénovací kanál - Model SVM					
	1	3	4	5	7	Ø		1	3	4	5	7	Ø		1	3	4	5	7	Ø
1	-	15,3	10,7	15,2	22,8	16,0	1	-	0,6	0,9	49,1	56,3	26,7	1	-	38,2	33,1	45,6	38,1	38,8
3	-0,5	-	7,0	11,6	7,3	6,3	3	6,5	-	-4,8	49,7	48,0	24,9	3	32,8	-	34,9	33,1	33,8	33,7
4	-0,2	6,6	-	11,1	4,9	5,6	4	9,9	8,9	-	36,0	31,9	21,7	4	36,6	33,1	-	36,9	31,7	34,6
5	13,5	17,7	5,2	-	9,2	11,4	5	-14,9	-10,1	-22,0	-	-6,6	-13,4	5	36,4	35,5	27,0	-	18,4	29,3
7	2,2	15,5	8,1	9,2	-	8,7	7	-8,2	4,1	-10,0	10,4	-	-0,9	7	34,7	35,9	27,1	24,0	-	30,4
Ø	3,7	13,8	7,7	11,8	11,0	-	Ø	-1,6	0,9	- 9,0	36,3	32,4	-	Ø	35,1	35,7	30,5	34,9	30,5	-
		Vyvážená přesnost [%]																		
Testovací kanál	Trénovací kanál - Model CNN LSTM						Testovací kanál	Trénovací kanál - Model MLP TL						Testovací kanál	Trénovací kanál - Model Novel CNN					
	1	3	4	5	7	Ø		1	3	4	5	7	Ø		1	3	4	5	7	Ø
1	-	12,3	15,4	33,1	24,7	21,4	1	-	9,4	9,3	16,4	12,1	11,8	1	-	28,7	19,2	43,5	31,3	30,7
3	8,1	-	13,3	32,2	30,4	21,0	3	11,0	-	14,8	16,5	17,9	15,1	3	18,7	-	24,1	32,7	31,0	26,6
4	16,4	18,5	-	21,8	14,7	17,9	4	13,1	11,8	-	13,9	13,8	13,1	4	18,8	28,9	-	33,2	24,8	26,4
5	25,0	26,1	12,3	-	4,1	16,9	5	20,6	17,2	16,0	-	11,0	16,2	5	45,7	32,3	23,9	-	18,0	30,0
7	26,8	27,8	13,0	11,2	-	19,7	7	24,7	19,7	19,0	11,4	-	18,7	7	34,7	32,7	24,0	17,2	-	27,1
Ø	19,1	21,2	13,5	24,6	18,5	-	Ø	17,4	14,5	14,8	14,6	13,7	-	Ø	29,5	30,7	22,8	31,7	26,3	-
Testovací kanál	Trénovací kanál - Model RF						Testovací kanál	Trénovací kanál - Model Resnet						Testovací kanál	Trénovací kanál - Model SVM					
	1	3	4	5	7	Ø		1	3	4	5	7	Ø		1	3	4	5	7	Ø
1	-	16,5	24,6	25,1	26,5	23,2	1	-	2,6	9,5	12,4	16,8	10,3	1	-	37,8	30,6	54,1	40,0	40,6
3	20,9	-	12,9	18,9	18,9	17,9	3	6,3	-	2,8	8,2	14,9	8,1	3	37,4	-	29,1	31,4	33,7	32,9
4	23,4	9,3	-	13,4	18,2	16,1	4	1,1	0,3	-	2,6	3,9	2,0	4	36,1	29,7	-	38,9	32,5	34,3
5	31,9	20,1	19,1	-	10,1	20,3	5	16,8	10,4	8,7	-	0,9	9,2	5	41,3	36,0	20,0	-	10,7	27,0
7	31,2	21,9	24,5	13,4	-	22,7	7	17,6	15,4	11,3	4,8	-	12,2	7	38,7	33,1	19,0	18,7	-	27,4
Ø	26,8	17,0	20,3	17,7	18,4	-	Ø	10,5	7,2	8,1	7,0	9,1	-	Ø	38,4	34,2	24,7	35,8	29,2	-
		F1 skóre [%]																		
Testovací kanál	Trénovací kanál - Model CNN LSTM						Testovací kanál	Trénovací kanál - Model MLP TL						Testovací kanál	Trénovací kanál - Model Novel CNN					
	1	3	4	5	7	Ø		1	3	4	5	7	Ø		1	3	4	5	7	Ø
1	-	15,2	17,7	24,1	14,9	18,0	1	-	5,0	5,5	8,9	7,4	6,7	1	-	22,6	20,4	25,8	24,4	23,3
3	0,3	-	12,3	14,4	10,4	9,4	3	8,3	-	11,1	14,4	13,2	11,7	3	9,7	-	18,8	21,7	20,3	17,6
4	5,1	11,3	-	28,8	14,7	15,0	4	6,2	8,2	-	8,6	9,7	8,2	4	12,6	23,9	-	24,8	17,9	19,8
5	7,5	14,1	7,6	-	-0,9	7,1	5	15,6	11,1	10,3	-	8,0	11,2	5	21,2	20,9	21,0	-	14,0	19,3
7	9,1	17,8	14,9	17,5	-	14,8	7	12,8	16,0	16,1	11,2	-	14,0	7	19,4	24,6	26,7	17,0	-	21,9
Ø	5,5	14,6	13,1	21,2	9,8	-	Ø	10,7	10,1	10,7	10,8	9,6	-	Ø	15,7	23,0	21,7	22,3	19,1	-
Testovací kanál	Trénovací kanál - Model RF						Testovací kanál	Trénovací kanál - Model Resnet						Testovací kanál	Trénovací kanál - Model SVM					
	1	3	4	5	7	Ø		1	3	4	5	7	Ø		1	3	4	5	7	Ø
1	-	10,9	6,4	10,6	16,9	11,2	1	-	0,6	0,4	51,4	60,6	28,3	1	-	27,5	23,5	32,0	27,2	27,6
3	-1,8	-	4,7	8,4	4,3	3,9	3	3,6	-	-4,4	60,2	50,1	27,4	3	22,8	-	25,9	24,4	24,3	24,4
4	-2,0	4,8	-	8,6	2,5	3,5	4	5,7	7,3	-	39,7	30,7	20,9	4	26,9	24,0	-	26,5	23,1	25,1
5	8,2	13,4	2,7	-	6,7	7,8	5	-22,0	-15,9	-25,0	-	-8,8	-17,9	5	26,2	25,9	19,9	-	13,2	21,3
7	-0,8	11,2	4,3	6,6	-	5,3	7	-11,2	1,3	-10,6	11,9	-	-2,2	7	24,6	26,4	20,0	17,1	-	22,0
Ø	0,9	10,1	4,5	8,5	7,6	-	Ø	-6,0	-1,7	- 9,9	40,8	33,2	-	Ø	25,1	25,9	22,3	25,0	21,9	-
		ROC AUC [%]																		
Testovací kanál	Trénovací kanál - Model CNN LSTM						Testovací kanál	Trénovací kanál - Model MLP TL						Testovací kanál	Trénovací kanál - Model Novel CNN					
	1	3	4	5	7	Ø		1	3	4	5	7	Ø		1	3	4	5	7	Ø
1	-	11,5	12,9	35,0	24,4	21,0	1	-	6,0	5,8	13,0	7,8	8,1	1	-	20,5	12,8	42,4	25,4	25,3
3	8,0	-	13,0	36,6	32,4	22,5	3	9,4	-	11,5	14,2	15,7	12,7	3	14,7	-	21,9	30,0	26,4	23,3
4	14,3	16,4	-	22,1	14,3	16,8	4	7,5	7,8	-	8,8	10,1	8,6	4	11,9	24,5	-	29,2	17,8	20,9
5	28,6	33,3	14,6	-	2,4	19,7	5	21,8	17,9	16,2	-	9,4	16,3	5	48,6	32,7	23,4	-	14,5	29,8
7	32,1	36,2	17,4	9,8	-	23,9	7	26,8	21,8	20,3	10,7	-	19,9	7	36,5	34,5	20,8	13,8	-	26,4
Ø	20,8	24,3	14,5	25,9	18,4	-	Ø	16,4	13,4	13,5	11,7	10,8	-	Ø	27,9	28,1	19,7	28,8	21,0	-
Testovací kanál	Trénovací kanál - Model RF						Testovací kanál	Trénovací kanál - Model Resnet						Testovací kanál	Trénovací kanál - Model SVM					
	1	3	4	5	7	Ø		1	3	4	5	7	Ø		1	3	4	5	7	Ø
1	-	16,5	24,6	25,1	26,5	23,2	1	-	1,5	5,6	51,2	49,5	27,0	1	-	37,8	30,6	54,1	40,0	40,6
3	20,9	-	12,9	18,9	18,9	17,9	3	12,7	-	4,6	34,0	43,2	23,6	3	37,4	-	29,1	31,4	33,7	32,9
4	23,4	9,3	-	13,4	18,2	16,1	4	20,6	13,2	-	18,5	30,5	20,7	4	36,1	29,7	-	38,9	32,5	34,3
5	31,9	20,1	19,1	-	10,1	20,3	5	32,9	28,3	22,4	-	6,9	22,6	5	41,3	36,0	20,0	-	10,7	27,0
7	31,2	21,9	24,5	13,4	-	22,7	7	31,2	33,1	27,1	9,2	-	25,2	7	38,7	33,1	19,0	18,7	-	27,4
Ø	26,8	17,0	20,3	17,7	18,4	-	Ø	24,3	19,0	14,9	28,2	32,5	-	Ø	38,4	34,2	24,7	35,8	29,2	-

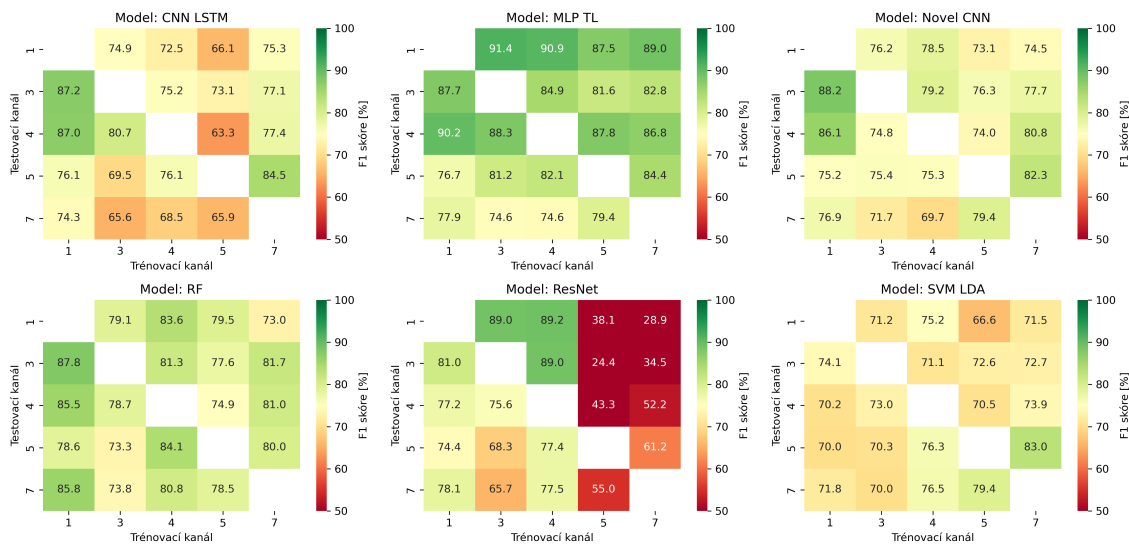
Tab. A.1: Rozdíly ve sledovaných metrikách pro modely trénované na jiném než testovacím kanálu vůči metrikám získaných tréninkem a ověřováním na stejném kanálu.

Vyvážená přesnost modelů testovaných a trénovaných na různých kanálech



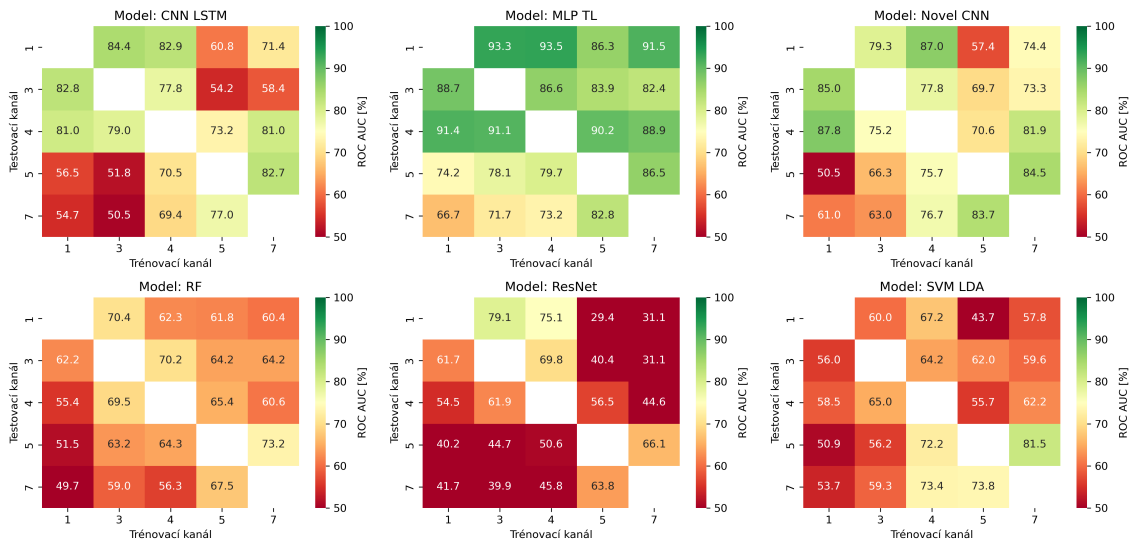
Obr. A.2: Heatmapa s hodnotami metriky „vyvážená přesnost“ pro modely trénované a testované na různých kanálech.

F1 modelů testovaných a trénovaných na různých kanálech



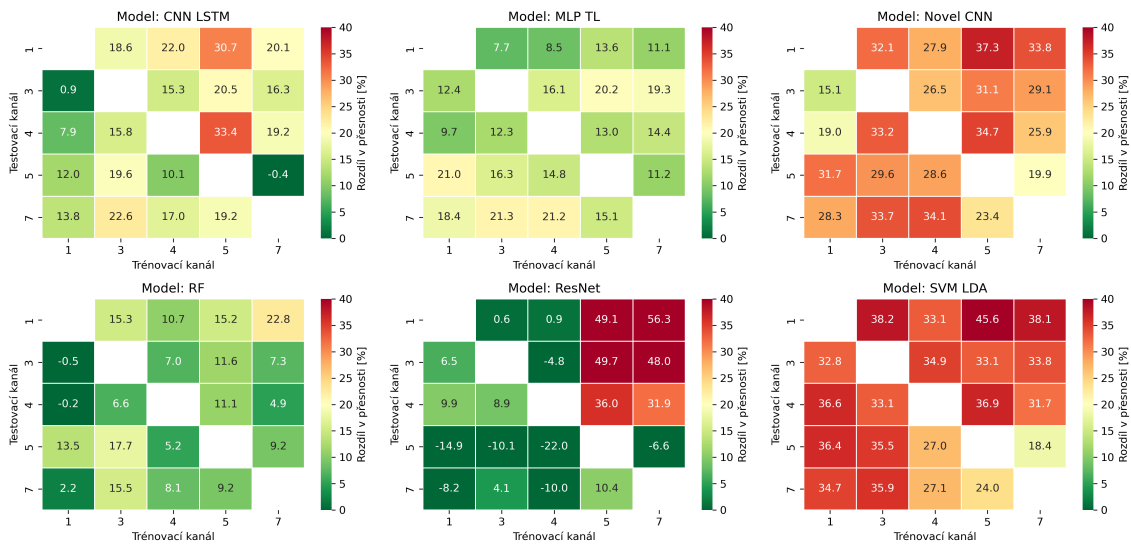
Obr. A.3: Heatmapa s hodnotami metriky „F1 skóre“ pro modely trénované a testované na různých kanálech.

RocAuc modelů testovaných a trénovaných na různých kanálech



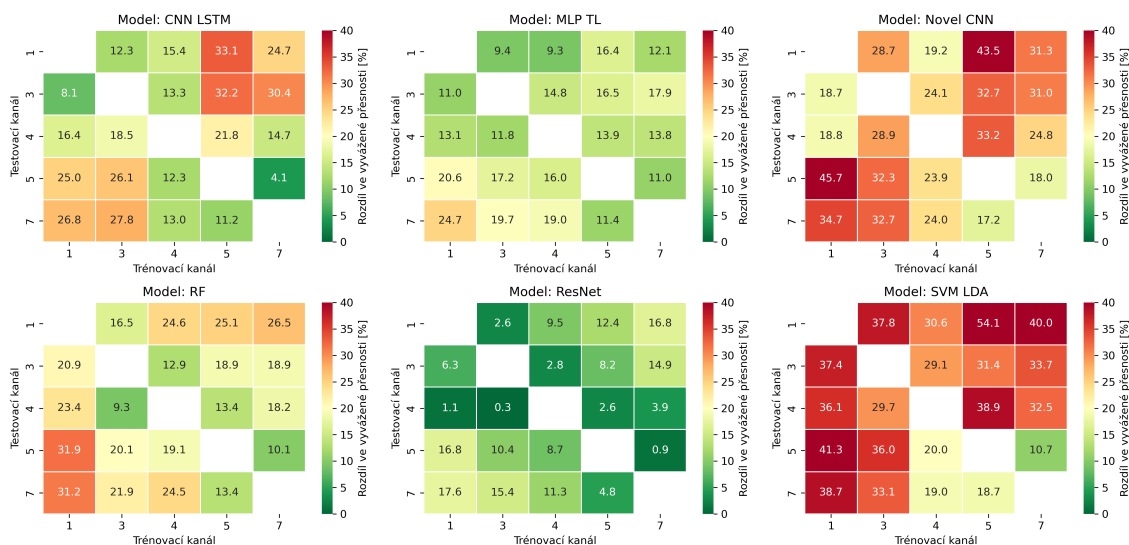
Obr. A.4: Heatmapa s hodnotami metriky „ROC AUC“ pro modely trénované a testované na různých kanálech.

Rozdíl mezi 1. a 2. scénářem pro přesnost



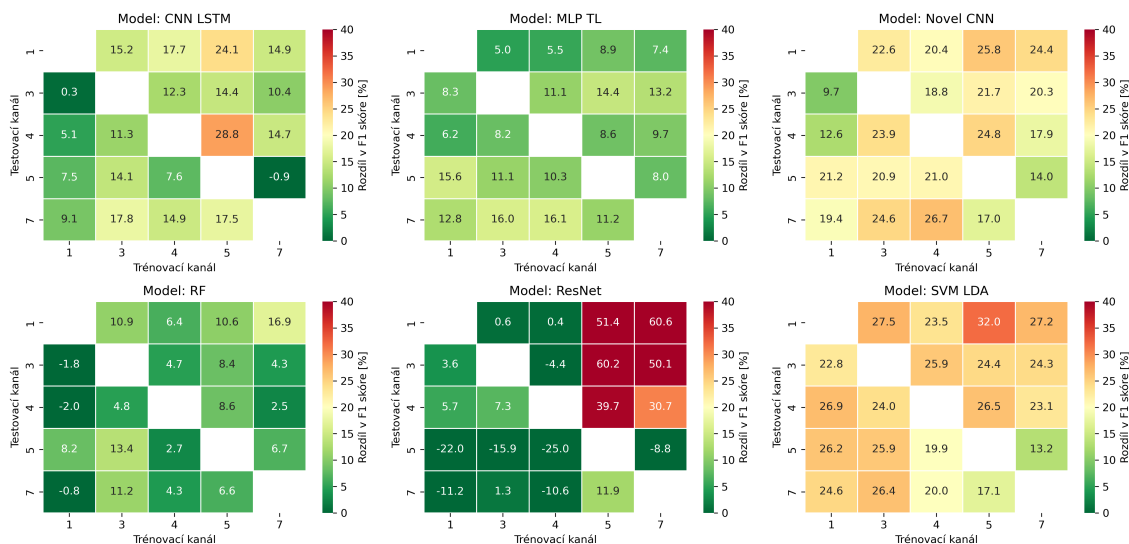
Obr. A.5: Heatmapa s rozdíly ve sledované metrice „přesnost“ pro modely trénované na jiném než testovacím kanálu vůči hodnotám metriky získaných tréninkem a ověřováním na stejném kanálu.

Rozdíl mezi 1. a 2. scénářem pro vyváženou přesnost



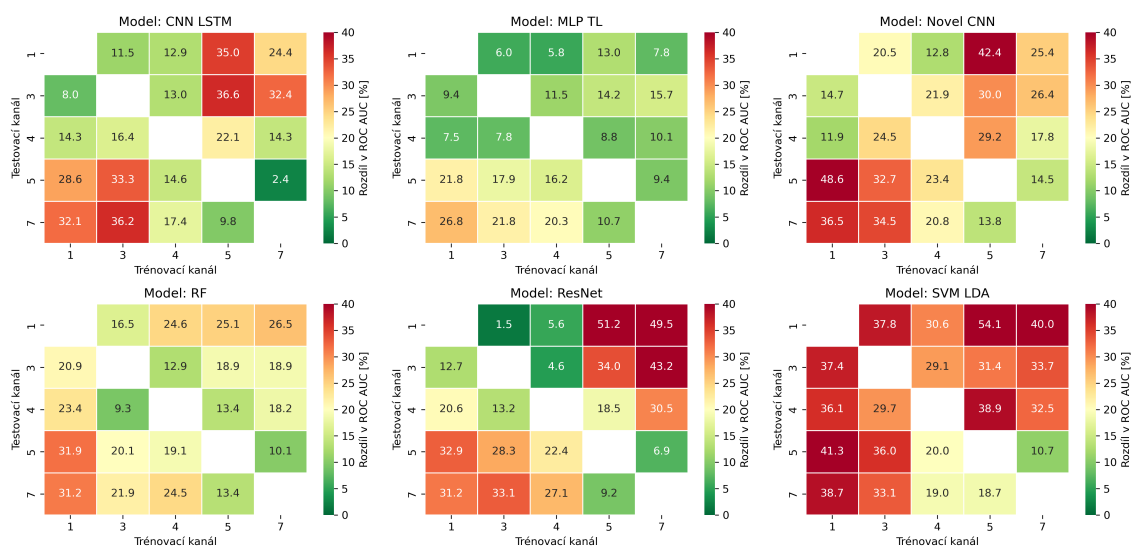
Obr. A.6: Heatmapa s rozdíly ve sledované metrice „vyvážená přesnost“ pro modely trénované na jiném než testovacím kanálu vůči hodnotám metriky získaných tréninkem a ověřováním na stejném kanálu.

Rozdíl mezi 1. a 2. scénářem pro F1 skóre



Obr. A.7: Heatmapa s rozdíly ve sledované metrice „F1 skóre“ pro modely trénované na jiném než testovacím kanálu vůči hodnotám metriky získaných tréninkem a ověřováním na stejném kanálu.

Rozdíl mezi 1. a 2. scénářem pro ROC AUC



Obr. A.8: Heatmapa s rozdíly ve sledované metrice „ROC AUC“ pro modely trénované na jiném než testovacím kanálu vůči hodnotám metriky získaných tréninkem a ověřováním na stejném kanálu.

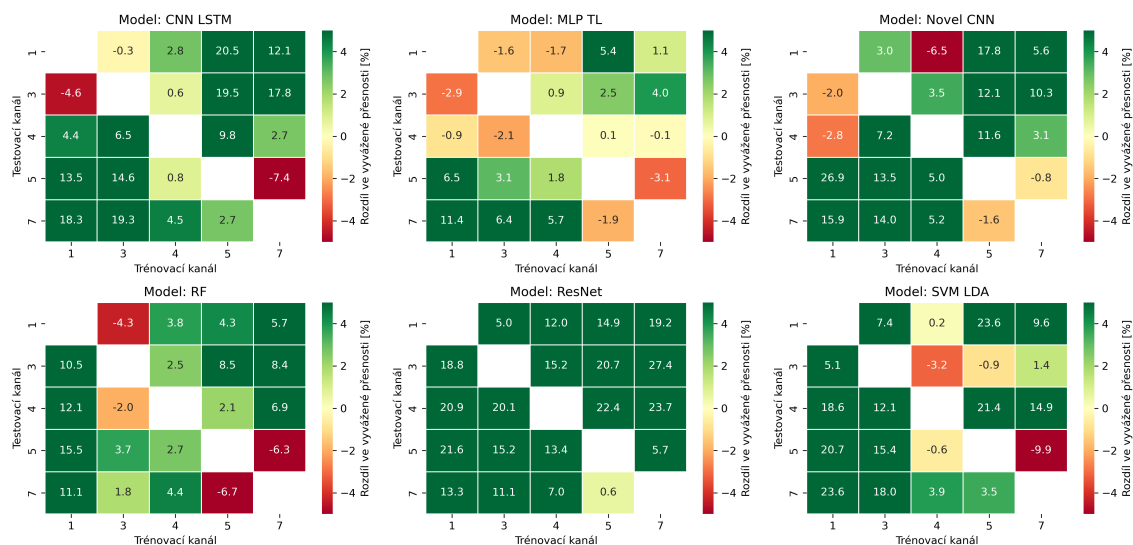
A.2 Scénář 4.4

V této části přílohy jsou prezentovány výstupy pro 3. scénář popsany v kapitole 4.4. Součástí přílohy je tabulka prezentující rozdíly v hodnotách sledovaných metrik mezi 1. a 3. scénářem a heatmapy prezentující rozdíly v hodnotách sledovaných metrik mezi 3. a 2. scénářem.

Model CNN LSTM		Ověřovaný kanál					AVG	Model MLP TL		Ověřovaný kanál					AVG	Model Novel CNN		Ověřovaný kanál					AVG
Metrika		1	3	4	5	7		Metrika		1	3	4	5	7		Metrika		1	3	4	5	7	
Přesnost [%]		20.6	21.4	23.3	4.3	10.0	15.9	Přesnost [%]		9.5	14.6	14.7	13.2	17.5	13.9	Přesnost [%]		28.2	20.7	24.1	23.1	26.3	24.5
Vyvážená přesnost [%]		12.6	12.7	12.0	11.5	8.5	11.5	Vyvážená přesnost [%]		11.0	13.9	13.9	14.1	13.3	13.3	Vyvážená přesnost [%]		25.7	20.6	21.6	18.8	18.7	21.1
F1 skóre [%]		16.9	18.9	19.4	2.6	8.1	13.2	F1 skóre [%]		6.2	9.8	9.9	9.2	13.4	9.7	F1 skóre [%]		19.9	14.0	16.6	16.7	19.6	17.4
ROC AUC [%]		6.7	13.3	8.6	10.7	11.0	10.1	ROC AUC [%]		7.6	11.2	9.2	12.4	13.2	10.7	ROC AUC [%]		19.4	17.5	15.3	17.6	15.1	17.0
Model RF		Ověřovaný kanál					AVG	Model ResNet		Ověřovaný kanál					AVG	Model SVM		Ověřovaný kanál					AVG
Metrika		1	3	4	5	7		Metrika		1	3	4	5	7		Metrika		1	3	4	5	7	
Přesnost [%]		15.8	9.2	7.5	16.2	8.1	11.4	Přesnost [%]		1.1	-1.4	-3.1	-34.3	-20.0	-11.5	Přesnost [%]		31.2	35.9	22.3	24.5	20.5	26.9
Vyvážená přesnost [%]		20.8	10.4	11.3	16.4	20.1	15.8	Vyvážená přesnost [%]		-2.4	-12.5	-19.8	-4.7	4.2	-7.0	Vyvážená přesnost [%]		30.4	32.3	17.6	20.6	15.1	23.2
F1 skóre [%]		11.1	6.9	5.3	12.6	4.8	8.1	F1 skóre [%]		1.1	-1.0	-2.7	-35.3	-19.0	-11.4	F1 skóre [%]		21.9	26.6	15.9	17.4	14.4	19.2
ROC AUC [%]		20.8	10.4	11.3	16.4	20.1	15.8	ROC AUC [%]		1.8	-5.8	-7.5	-3.0	5.4	-1.8	ROC AUC [%]		30.4	32.3	17.6	20.6	15.1	23.2

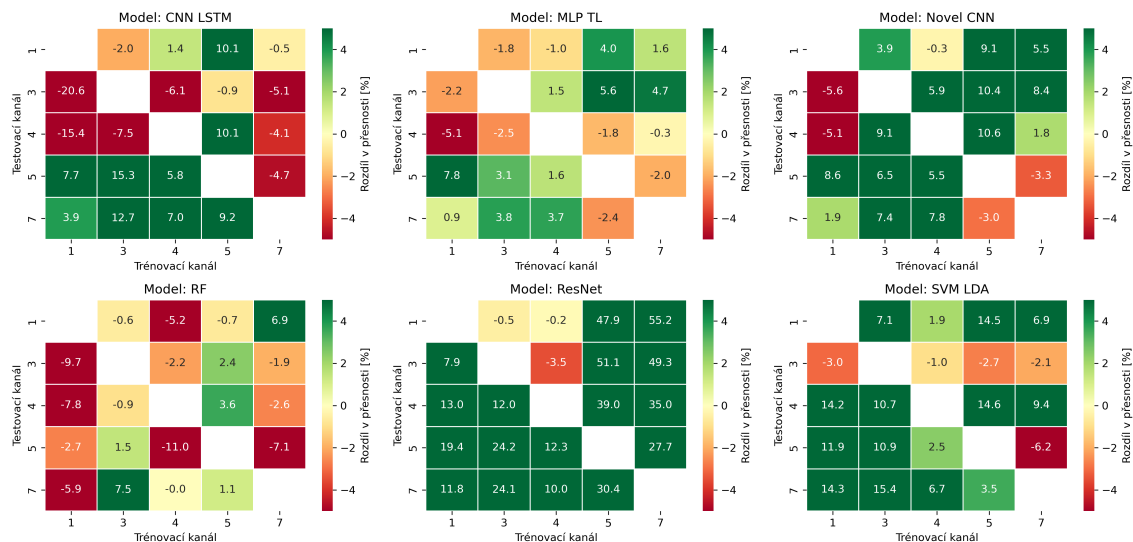
Tab. A.2: Pokles hodnot jednotlivých metrik ve scénáři 4.4 v porovnání s referenční hodnotou získanou tréninkem a ověřováním na stejném kanálu.

Rozdíl mezi 3. a 2. scénářem pro vyváženou přesnost



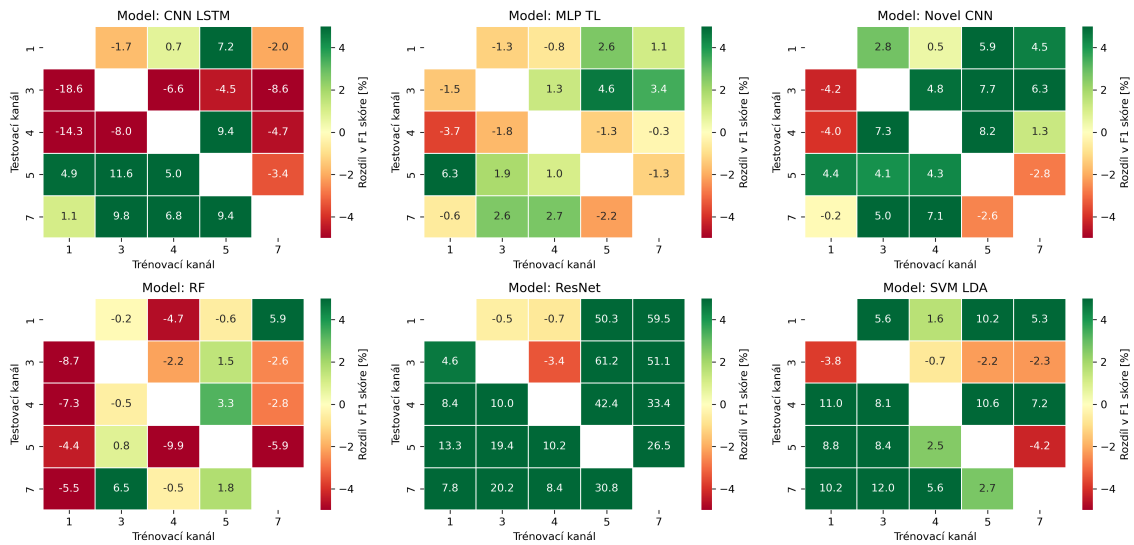
Obr. A.9: Heatmapa s rozdíly ve sledované metrice „vyvážená přesnost“ mezi 3. a 2. scénářem.

Rozdíl mezi 3. a 2. scénářem pro přesnost



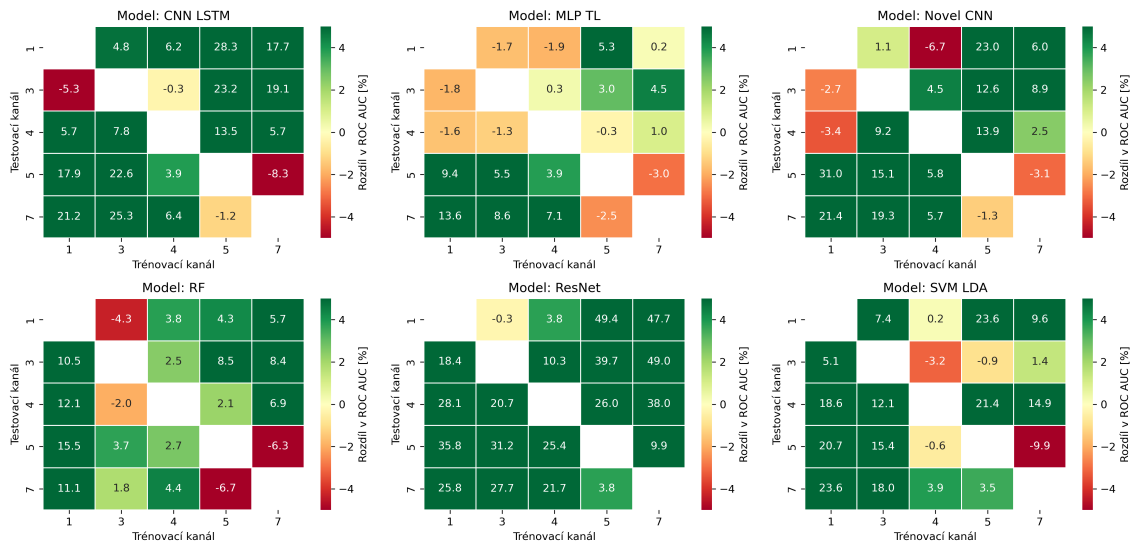
Obr. A.10: Heatmapa s rozdíly ve sledované metrice „přesnost“ mezi 3. a 2. scénářem.

Rozdíl mezi 3. a 2. scénářem pro F1 skóre



Obr. A.11: Heatmapa s rozdíly ve sledované metrice „F1 skóre“ mezi 3. a 2. scénářem.

Rozdíl mezi 3. a 2. scénářem pro ROC AUC



Obr. A.12: Heatmapa s rozdíly ve sledované metrice „ROC AUC“ mezi 3. a 2. scénářem.

B Obsah elektronické přílohy

Rozdělení elektronické přílohy této práce odpovídá rozdělení této práce.

V první praktické části práce byly podrobně ověřovány publikované metody pro klasifikaci, jejichž implementace jsou umístěny ve složce „Modely“. Součástí složky je i spustitelný soubor „Evaluate.py“, ve kterém jsou definovány scénáře pro každý model.

Ve druhé části byly popisovány další publikované metody pro klasifikaci a u některých z nich byly stanoveny parametry pro porovnání výpočetní a paměťové náročnosti. Implementace těchto modelů a spustitelné soubory pro stanovení parametrů jsou umístěny ve složce „Narocnost“.

Poslední částí této práce je stanovení vlivu frekvence na přenositelnost jednotlivých modelů. Zdrojové kódy pro realizaci scénářů, statistické výpočty a grafické výstupy jsou spolu se složkami, obsahujícími vybrané výstupy jednotlivých scénářů, umístěny ve složce „Vliv frekvence“. Výstupy obsažené ve složkách jednotlivých scénářů jsou popsány v rámci kapitol, které se věnují jejich popisu a vyhodnocení.

K implementaci modelů a dalších funkcí pro účely této práce byl využit programovací jazyk Python ve verzi 3.9.20, a zejména pak knihovny Tensorflow 2.10.1 [32], Scikit-learn 1.5.2 [33], Psutil 7.0.0 [37], Tracemalloc [38] a Seaborn 0.13.2 [85]. Komentáře ve zdrojovém kódu jsou pro případný další rozvoj této práce uváděny v anglickém jazyce, všechny výpisy programu jsou ale v jazyce českém. Následující text znázorňuje strukturu elektronické přílohy této práce.

```
/.....kořenový adresář přiloženého archivu
├── Funkce.....pomocné funkce pro ověřování publikovaných řešení
│   ├── DataOperations.py.....funkce pro práci s daty
│   ├── ResultsOperations.py.....funkce pro vyhodnocení výstupů modelů
│   └── TrainingOperatios.py.....funkce pro trénink a evaluaci modelů
├── Modely.....modely z kapitoly 3.2
│   ├── CNNLSTM_Class.py
│   ├── Evaluate.py.....spustitelný soubor s jednotlivými scénáři
│   ├── MLPTL_Class.py
│   ├── NovelCNN_Class.py
│   ├── ResNet_Class.py
│   ├── RF_Class.py
│   └── SVM_Class.py
├── Vliv frekvence.....výsledky a funkce pro stanovení vlivu frekvence
│   ├── Scenar1
│   ├── Scenar2
│   ├── Scenar3
│   ├── Calculations.py.....výpočty nad sledovanými metrikami
│   ├── Main.py.....spustitelný soubor s jednotlivými scénáři
│   └── StatisticsTest.py.....statistické testy
```

—	<code>Visualisation.py</code>	grafické výstupy
—	<code>Vysledky_eval.csv</code>	
—	<code>Vysledky_train.csv</code>	
—	<code>Narocnost</code>	soubory pro stanovení náročnosti modelů
—	<code>ComplexityFunctions.py</code>	funkce pro stanovení náročnosti
—	<code>Described_Models.py</code>	Implementace modelů z kapitoly 3.3
—	<code>E2E_MemoryUsage_Psutil.py</code>	spustitelný soubor pro stanovení paměťové náročnosti během běhu metodou <i>psutil</i>
—	<code>E2E_MemoryUsage_Tensorflow.py</code>	spustitelný soubor pro stanovení paměťové náročnosti během běhu metodou <i>tensorflow</i>
—	<code>E2E_MemoryUsage_Tracemalloc.py</code>	spustitelný soubor pro stanovení paměťové náročnosti během běhu metodou <i>tracemalloc</i>
—	<code>Main.py</code> ...	spustitelný soubor pro stanovení výpočetní a paměťové náročnosti
—	<code>MWTCNN_Class.py</code>	implementace modelu <i>MWT_CNN</i>