VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

VYUŽITÍ STROJOVÉHO UČENÍ K ROZPOZNÁNÍ POHYBU FEEDEROVÉHO PRUTU

EXPLOITATION OF MACHINE LEARNING FOR IDENTIFICATION OF FEEDER ROD MOVEMENT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE **Bc. PATRIK VELE**

AUTHOR

VEDOUCÍ PRÁCE **Ing. VÁCLAV ŠIMEK**

SUPERVISOR

BRNO 2022

Ústav počítačových systémů (UPSY) Akademický rok 2021/2022

Zadání diplomové práce

Student:	**Vele Patrik, Bc.**
Program:	Informační technologie
Obor:	Počítačové a vestavěné systémy
Název:	**Využití strojového učení k rozpoznání pohybu feederového prutu** **Exploitation of Machine Learning for Identification of Feeder Rod Movement**
Kategorie:	Vestavěné systémy

Zadání:

1. Prostudujte základní aspekty rybolovné techniky feeder. Pozornost zaměřte především na typické pohyby při manipulaci s prutem.
2. Dle poznatků z bodu 1) zadání stanovte klíčové pohyby prutu pro detekci. Dále vyberte oblast prutu vhodnou pro umístění detekčního zařízení.
3. Seznamte se s aktuálními trendy v oblasti strojového učení. Pokuste se identifikovat techniky vhodné pro detekci pohybu prutu na základě údajů z inerciální jednotky (IMU).
4. Zvolte vhodné komponenty a vytvořte schéma zapojení pro detekční zařízení. Připravte motiv desky plošných spojů a proveďte jeho fyzickou realizaci.
5. Implementujte obslužný firmware, který mimo jiné umožní vytvoření datové sady jednotlivých typů pohybu prutu.
6. Vytvořte datovou sadu klíčových pohybů prutu. Aplikujte vhodnou techniku strojového učení dle bodu 3) zadání k jejich detekci.
7. Vhodnou formou demonstrujte funkčnost realizovaného řešení.
8. Zhodnoťte dosažené výsledky, diskutujte možnosti dalšího rozvoje a případná vylepšení.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 4 zadání.

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

Vedoucí práce:	**Šimek Václav, Ing.**
Vedoucí ústavu:	Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání:	1. listopadu 2021
Datum odevzdání:	18. května 2022
Datum schválení:	29. října 2021

Abstrakt

Cílem této diplomové práce je vytvořit zařízení, které využívá metody strojového učení k rozpoznávání pohybů feederového rybářského prutu na základě dat z inerciální měřící jednotky. Úvodní část je věnována rybolovné technice feeder, výběru důležitých pohybů a možnostem upevnění detekčního zařízení na prut. Následuje vytvoření teoretického základu v oblasti strojového učení, seznámení s inerciální měřicí jednotkou a problematikou klasifikace. Obdržené znalosti jsou použity k výběru vhodných technik pro řešení úlohy rozpoznávání pohybů prutu. V praktické části je navrženo a vytvořeno detekční zařízení založené na platformě ESP32. To je nejprve používáno jako snímač pohybu, který v kombinaci se zpracováním naměřených hodnot slouží jako generátor trénovací datové sady. Práce pokračuje implementací konvoluční neuronové sítě, procesem učení na vytvořené datové sadě a integrací nejúspěšnějšího modelu do detekčního zařízení. Závěr je věnován testování v praxi, vyhodnocení a možnostem budoucího vývoje. Výsledkem je malé, bateriově napájené zařízení, které po připevnění na libovolný feederový prut poskytuje vysoce úspěšnou detekci všech klíčových pohybů během lovu. Navíc díky bezdrátové komunikaci přes ESP-NOW umožňuje odesílat výsledky na různá zařízení.

Klíčová slova

Feeder, detekce pohybu, strojové učení, Tiny Machine Learning, konvoluční neuronová síť, vestavěný systém, ESP32, datová sada, ESP-NOW, TensorFlow Lite, Google Colaboratory, PlatformIO.

Abstract

The aim of this diploma thesis is to create a device that uses machine learning methods to recognize the movements of a feeder fishing rod based on data from an inertial measurement unit. The introductory part is devoted to the feeder fishing technique, the selection of important movements and the possibilities of attaching the detection device to the rod. This is followed by the creation of a theoretical basis in the field of machine learning, familiarization with the inertial measurement unit and the issue of classification. The acquired knowledge is used to select appropriate techniques for solving the task of recognizing the movements of the rod. In the practical part, a detection device based on the ESP32 platform is designed and created. This is initially used as a motion sensor, which, in combination with the processing of the measured values, serves as a generator of a training data set. The work continues with the implementation of the convolutional neural network, the learning process on the created dataset and the integration of the most successful model into the detection device. The conclusion is devoted to testing in practice, evaluation and possibilities of future development. The result is a small, battery-powered device that, when attached to any feeder rod, provides highly successful detection of all key movements during the hunt. In addition, thanks to wireless communication via ESP-NOW, it is possible to send the results to various devices.

Keywords

Feeder, movement detection, machine learning, Tiny Machine Learning, convolution neural network, embedded system, ESP32, dataset, ESP-NOW, TensorFlow Lite, Google Colaboratory, PlatformIO.

Citace

VELE, Patrik. *Využití strojového učení k rozpoznání pohybu feederového prutu*. Brno, 2022. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek.

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Patrik Vele
18. května 2022

Poděkování

Chtěl bych poděkovat panu Ing. Václavu Šimkovi za odborné vedení, ochotu, trpělivost, pomoc a cenné rady při zpracování této diplomové práce. Chtěl bych také poděkovat panu Ing. Vojtěchu Mrázkovi, Ph.D. za jeho ochotu a cenné rady. Velké poděkování patří i mé rodině a přátelům za jejich podporu během období celého studia. V neposlední řadě bych chtěl moc poděkovat své přítelkyni za pomoc, podporu a motivaci v podobě nikdy nekončícího optimismu.

Obsah

Seznam obrázků

1 Úvod

Rybolovná technika feeder se během krátké doby své existence stala velice populární a uznávanou disciplínou, která je dnes hojně používána jak pro běžný lov, tak ke sportovní činnosti díky celoročnímu pořádání nejrůznějších závodů včetně mistrovství světa. Právě velké množství konaných soutěží a neustálý vývoj nových produktů a feederu obecně dospěl k tomu, že kvality vrcholových závodníků jsou téměř srovnatelné, a o jejich výsledcích rozhodují čím dál menší detaily. Jedením takovým je i frekvence nahazování, která je v současné době během závodu přesně dodržována za pomoci ručních stopek. Přestože se jedná o funkční a jednoduchý způsob, tak závodníka neustále zaměstnává, a především mu neposkytuje žádný záznam pro případnou budoucí analýzu. Informace jsou tak v průběhu času závislé na paměti a odhadu člověka, což může hrát vlivem nepřesnosti zásadní roli. Právě tato oblast sběru dat, automatických záznamů a generování statistik představuje prostor pro integraci nějaké technologie, která by mohla být dalším pokrokem ve vývoji nejen závodního feederu.

V závislosti na této skutečnosti je stanoveným cílem této diplomové práce vytvořit zařízení, které bude během lovu automaticky sbírat data, z nich pak následně extrahovat užitečné informace a výsledky předávat uživateli. Jako zdroj dat bude sloužit rybářský prut, který jakožto hlavní pracovní nástroj rybáře poskytuje svými charakteristickými pohyby mnoho informací týkajících se lovu. Tou pravděpodobně největší výzvou celé práce pak bude detekce a správné vyhodnocení klíčových pohybů. K tomu bude použito strojové učení, které představuje silný nástroj s potenciálem řešit tuto úlohu s vysokou úspěšností, což dokazuje jeho dnešní hojné používání při řešení úlohy rozpoznávání aktivit člověka na základě pohybu. Navíc je v posledních letech díky velkému technologickému pokroku možné integrovat strojové učení přímo do malých přenosných zařízení. Odpadá tak nutnost odesílání dat na výkonný výpočetní stroj, což je přístup, který by celý proces značně komplikoval.

Samotná práce se skládá z několika hlavních částí. Ta první je věnována prostudování základních aspektů rybolovné techniky feeder. Z velké části je však pozornost věnována především specifickému prutu a manipulaci s ním během lovu, což je nezbytnou přípravou pro stanovení klíčových pohybů a dále také výběru oblasti prutu, kam bude detekční zařízení umístěno. Následuje vytvoření teoretického základu v oblasti strojového učení a seznámení se současnými možnostmi integrace těchto algoritmů do energeticky nenáročných zařízení. Na to navazuje již samotný výběr konkrétních technik, které by mohly být vhodné pro detekci jednotlivých pohybů prutu. Tato fáze vychází ze specifikace úlohy, seznámení s problematikou klasifikace, inerciální měřící jednotkou a úlohou rozpoznávání lidské činnosti. Práce pokračuje praktickou částí, konkrétně tedy návrhem zařízení, výběrem vhodných komponent, navržením, vytvořením a osazením desky plošných spojů a umístěním všech součástí do navržené a vytisknuté krabičky na 3D tiskárně. Vytvořené detekční zařízení je nejprve používáno jako snímač, ke kterému je vytvořeno ještě označovací zařízení. Vzájemná spolupráce této dvojice pak vytváří záznam jednotlivých pohybů během lovu, ze kterého je následným zpracováním vygenerována datová sada. Další část představuje implementaci konvoluční neuronové sítě, proces jejího učení na vytvořené datové sadě a převedení nejúspěšnějšího modelu do podoby umožňující nasazení na detekční zařízení. Poté je vytvořen výsledný software pro detekční zařízení. Ten se skládá z pěti hlavních bloků – čtení a příprava dat pro klasifikaci založená na principu posuvného okna, nasazení modelu a klasifikace pohybů, distribuce výsledků přes ESP-NOW, detekce stavu baterie zařízení a informování uživatele pomocí jednoduchého rozhraní, které se skládá z osmi LED diod. Následuje fáze testování funkčnosti a úspěšnosti produktu v praxi. V samotném závěru jsou pak diskutovány možnosti budoucího vývoje.

2 Rybolovná technika feeder

Feeder je moderní rybolovná technika, která pochází z Anglie [40]. Jedná se o lov na položenou, což znamená, že závaží je položeno na dně a pohyb nástrahy není rybářem během lovu aktivně ovlivňován [46]. Jinými slovy, prut je při lovu umístěn ve stojanu, vidličkách nebo podobném zařízení. Od klasické položené se ale feeder liší v několika bodech.

Tím hlavním je detekce záběru, který je signalizován samotným prutem pomocí pohybu měkké špičky. Nejsou tedy potřeba žádné další číhátka, hlásiče, policajti a další příslušenství. Feeder je tak mnohem rychlejší a především citlivější, protože dokáže rozpoznat i velmi jemné záběry. Díky této vysoké citlivosti je špička schopna přenést mnohem větší informaci, kterou zkušený rybář dokáže různými způsoby využít a zvýšit tak počet zdolaných ryb.

Z prvního rozdílu vyplývá i ten druhý, kterým je umístění prutu do stojanu tak, aby špička prutu svírala s vlascem vedoucím do vody úhel ideálně 90 stupňů. Tím se maximalizuje pohyb špičky, a tedy i rozpoznatelnost případného záběru.

Obrázek 2.1: Feederový prut [11]

Posledním klíčovým rozdílem je princip lovu, který jak už samotný název této rybolovné techniky napovídá, je založen na krmení. Základní myšlenkou feederu je vytvořit co nejmenší krmné místo, které je během lovu udržováno pravidelným dokrmováním. Takové místo ryby přiláká, a především udrží po celou dobu lovu, protože neustále očekávají další a další potravu. Výsledkem je pak mnohem větší úspěšnost v počtu ulovených ryb oproti klasické položené. Aby bylo tohoto efektu docíleno, je potřeba často nahazovat dostatečné množství návnady do stejného místa. Pro její maximálně efektivní dopravu do vody se používají speciální feederová krmítka, která se liší typem, velikostí, tvarem, hmotností i materiálem. Přesnost nahazování je určena vzdáleností a směrem. Vzdálenost je nastavena takzvaným zaklipováním vlasce na cívce navijáku. Směr pak záleží na schopnostech rybáře, který míří své hody na pevný bod protějšího břehu. Posledním parametrem krmení je frekvence nahazování, která závisí na mnoha aspektech. Pro představu lze však konstatovat, že při běžném lovu dochází k přehazování maximálně každých 10 minut. Často se ale stává, že frekvence je nastavena rybími záběry, které přicházejí chvíli po každém náhozu.

Popularitě feederu nahrává velká paleta použití. Sortiment, který je dnes dostupný umožňuje rybáři zaměřit se ryby od nejmenších velikostí až po trofejní kapry, případně je možný i lov dravců.

Z hlediska prostředí je feeder použitelný od stojatých vod, až po velké a silně proudící řeky. Existují i pruty pro daleké nahazování, se kterými není problém dosažení vzdálenosti přes 100 metrů.

Během krátké doby si u nás feeder oblíbilo mnoho rybářů, což mimo jiné dokazuje vytvoření samostatné sekce LRU Feeder na Českém rybářském svazu a pořádání oficiálních soutěží pro registrované sportovce, kterých je k datu 13.10.2021 registrováno celkem 399 [51]. Mezi oficiální pořádané soutěže pak podle soutěžního řádu patří náborové závody, pohárové závody, mezinárodní a mezistátní závody, mistrovství České republiky, 1. liga, 2. liga a divize [56]. Stejně tak je pořádáno i mnoho neoficiálních soutěží. Mezi ty poslední dobou nejvíce populární patří takzvané zimní ligy. Ty jsou často organizovány a navštěvovány registrovanými sportovci, kteří tak vyplňují volné zimní období mezi sezónami a udržují se v závodním rytmu. V praxi má tedy registrovaný sportovec téměř každý víkend v roce možnost účastnit se nějaké soutěže.

Obrázek 2.2: Feederový závod [63]

Svou popularitu má feeder i ve světě. Historicky první mistrovství světa bylo pořádáno v roce 2011 v Itálii a zúčastnilo se ho celkem 16 zemí včetně České republiky [17]. Na posledním konaném mistrovství v roce 2019 bylo zemí dokonce 20. Většinou se jednalo o evropské státy, mezi kterými se ale objevila například i Austrálie a Jihoafrická republika [10].

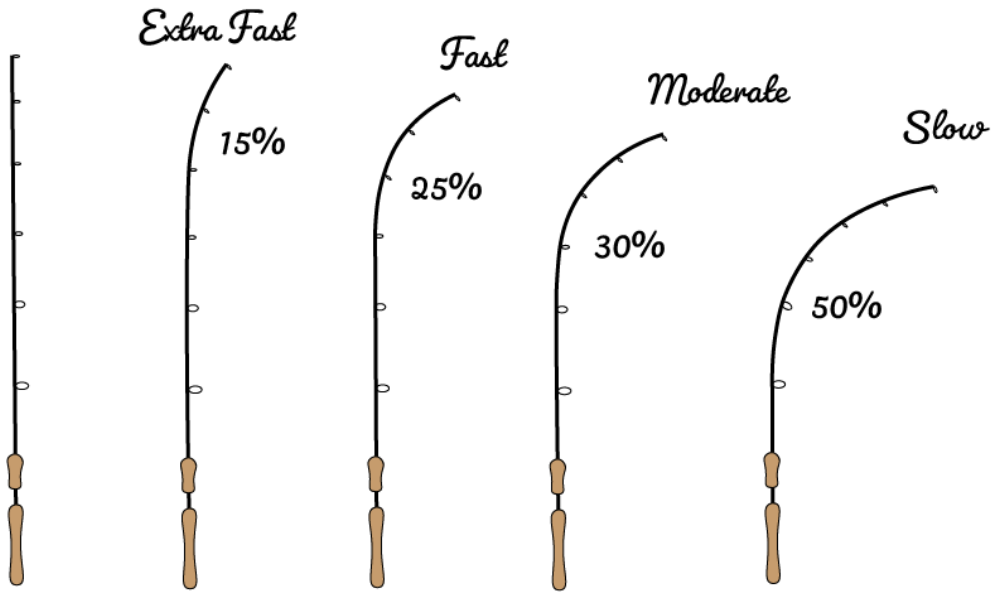
2.1 Feederový prut

Feederový prut je specifický tím, že musí umožňovat nahazování těžkých krmítek, zároveň dobře detekovat i jemné rybí záběry, musí zajistit rybáři jistotu při záseku, dobrou akci při zdolávání a v neposlední řadě musí také být lehký a pohodlný kvůli časté manipulaci. Protože zmíněné požadavky vedou na protichůdné vlastnosti, tak neexistuje ideální feederový prut, ale je vyráběna široká paleta prutů, kde každý z nich v určitém poměru kombinuje požadovaná specifika. Zanedbáním extrémních případů je možné feederové pruty rozdělit do pěti orientačních kategorií podle odhozové zátěže.

- picker – odhozová zátěž do 40 g, délka prutu 210 – 300 cm, lov na vzdálenost do 25 m
- light – odhozová zátěž 50 - 70 g, délka prutu 300 - 360 cm, lov na vzdálenost do 40 m
- medium – odhozová zátěž 70 - 110 g, délka prutu 330 – 390 cm, lov na vzdálenost do 60 m
- heavy – odhozová zátěž 110 - 150 g, délka prutu 360 – 390 cm, lov na vzdálenost do 80 m
- extra heavy – odhozová zátěž 150 - 200 g, délka prutu 390 – 420 cm, lov na vzdálenost i přes 100 m [23]

Z hlediska stavby se jedná především o dělené pruty, které jsou charakteristické vyměnitelnými barevnými špičkami různé tvrdosti. Citlivá špička slouží k signalizaci záběru, a proto je její správná volba klíčová. Pro lepší orientaci rybáře je tvrdost špičky označena hodnotou v uncích, která udává kolik uncí je potřeba na zatížení špičky, aby došlo k jejímu ohybu do pravého úhlu. Běžně používané hodnoty začínají u pickerových prutů od 0,5 oz a končí u extra heavy prutů na 5 oz [24].

Základ prutu však představuje blank, který přímo definuje prut a jeho klíčové vlastnosti, kde jednou z těch zásadních je akce. Ta může být buď rychlá nebo pomalá. Případně se může jednat o nějaký kompromis. Rychlejší pruty disponují přesnějšími a delšími hody, ale hůře se s nimi zdolávají ryby, protože nejsou schopné tak dobře vypružit rybí výpady. Naopak pomalejší pruty s parabolickou akcí nejsou vhodné na daleké a přesné hody, ale velice dobře pohlcují rybí výpady při zdolávání, což umožňuje použít i jemné montáže [23].



Obrázek 2.3: Akce prutu [22]

2.2 Manipulace s prutem

Pokud přeskočíme sestavení prutu a případné nastavením lovné vzdálenosti, což jsou z pohledu manipulace nezajímavé části, tak každý lov začíná nahozením. To je v případě feederu provedeno buď za účelem pouze krmit, nebo s cílem chytit rybu.

Pokud se jedná o krmení, tak je prut po nahození ustálen v jedné pozici po dobu, než se dostane krmítko s krmením na požadované místo, kterým je zpravidla dno. Poté následuje vysypání krmné směsi sérií několika malých záseků, které jsou provedeny během jednoho tahu prutu směrem nahoru. Ve chvíli, kdy je prut kolmo k zemi, následuje vytažení prázdného krmítka z vody.

Pokud bylo nahozeno s cílem ulovit rybu, tak je prut po náhozu umístěn do nějakého stojanu a je čekáno na záběr od ryby. Poloha jeho umístění závisí na několika aspektech, kde hlavní dva jsou proud vody a vítr. V silném proudu je výhodné nastavit prut tak, aby špička byla co nejvýše. Tím se minimalizuje část vlasce pod vodní hladinou, na kterou působí proud, což rybáři umožní používat lehčí krmítko a měkčí špičku. Naopak při silném větru je výhodné špičku co nejvíce přiblížit vodní hladině, aby se eliminovali její pohyby způsobené větrem působícím na vlasec. Zároveň by měl být prut vždy nastaven tak, aby s vlascem mířícím do vody svíral v ideálním případě úhel 90 stupňů.

Při záběru ryby pak následuje zásek, což je prudký pohyb prutu směrem nahoru. První fáze tohoto pohybu je trh, který postupně přechází v tah končící ve chvíli, kdy je prut kolmo k zemi. Pak

následuje přimotání vlasce a zdolávání ryby. Zásek ale nemusí být vždy přímo nahoru, protože pro maximální úspěšnost by měl vést směrem od vlasce vedoucího do vody. Umístění prutu ve stojanu tak může směr záseku částečně ovlivnit.

Obrázek 2.4: Zásek prutem směrem do strany [45]

V případě, že záběr od ryby do určité doby nepřijde, dochází k vytažení montáže. Při této akci je prut nejprve přizvednut ze stojanu a bezprostředně po tom dochází k namotávání vlasce s tím, že se prut postupně dostává do pozice, kde svírá s vodní hladinou úhel asi 45 stupňů do doby, než je montáž přitažena ke břehu.

Poslední signifikantní manipulací s prutem je jeho odložení a výměna za jiný. Tato praktika je většinou používaná pouze na závodech, kde má každý závodník připraveno několik různých prutů na břehu vedle sebe. V průběhu soutěže pak podle situace volí ten nejvhodnější z nich.

Na základě všech popsaných informací týkajících se manipulace s feederovým prutem je možné stanovit klíčové pohyby pro detekci, ke kterým během lovu dochází:

- Nahození
- Krmení
- Zaseknutí
- Stažení
- Odložení

2.3 Umístění detekčního zařízení na prut

Při prvním pohledu na prut se intuitivně nabízí možnost připevnění detekčního zařízení do oblasti rukojeti, která již disponuje mechanismem pro umístění navijáku. Této myšlence také nahrává měkký materiál, vhodný pro instalaci různých držáků. Dalšími výhodami jsou nulová ohebnost této části a minimální vliv na vlastnosti prutu v případě připevnění detekčního zařízení. Problémem je ale využívání rukojeti. Její část před navijákem je určena pro držení prutu. Vrchní část za navijákem je

při zdolávání opřena o předloktí rybáře a spodní část tvoří styčnou plochu při umístění prutu do držáku. Úplný konec rukojeti je při náhozu uchopen druhou rukou rybáře. Oblast rukojeti prutu je tedy i přes vhodné vlastnosti pro umístění detekčního zařízení nepoužitelná, a proto bude muset být zařízení umístěno na blanku prutu.

Avšak pro výběr správného místa je potřebné najít kompromis mezi minimálním ovlivněním vlastností prutu a maximálním rozsahem pohybu snímače. Jinými slovy, zařízení by mělo být dostatečně blízko navijáku, aby nenarušovalo akci prutu, rozložení hmotnosti, pákové poměry a podobně, a zároveň by mělo být dostatečně daleko, aby zajišťovalo velký rozsah naměřených hodnot, a tím i lepší podmínky na rozpoznání konkrétního pohybu. Mezi další požadavky patří minimální ohebnost prutu v místě upevnění zařízení a dostatečná vzdálenost mezi blankem a vlascem vedoucím v očkách, aby nedocházelo k dotyku vlasce s detekčním zařízením. Zahrnutím všech zmíněných kritérií se jako nejvíce vyhovující místo jeví oblast mezi začátkem blanku a prvním očkem prutu.

3 Strojové učení

Strojové učení tvoří podmnožinu umělé inteligence a je možné ho definovat jako soubor výpočetních metod, které využívají zkušenost k vylepšení výkonnosti nebo ke zvýšení přesnosti predikcí [32]. Jedná se tak o velice silný nástroj na některé problémy, pro které je velmi obtížné najít přesný algoritmus s dostatečnou úspěšností. Příkladem mohou být úlohy, které člověk řeší intuitivně správně. Mezi ně spadá rozpoznávání objektu na obrázku, zpracování jazyka, klasifikace dokumentů, učení se hraní her typu šachy a mnoho dalších.

3.1 Historie

První spojitosti s umělou inteligencí obecně sahají hluboko do historie. Například v 17. století publikoval Gottfried Wilhelm Leibnitz svou disertační práci, ve které pracuje s teorií, že myšlenky jsou výsledkem výpočtu, a věřil, že pokud by se mu podařilo popsat symbolickou reprezentaci těchto pravidel a vytvořil by metodu, která by tato pravidla kombinovala, byl by schopen generovat nové myšlenky [54] [52].

Za počátek umělé inteligence tak jak ji známe dnes jsou však považována až 50. léta 20. století, která obsahovala několik klíčových událostí. V roce 1950 byl Alanem Turingem prezentován Turingův test, jehož cílem bylo prověřit, zda se nějaký systém chová inteligentně [35]. V roce 1956 byl poprvé použit termín „umělá inteligence“. Jednalo se o letní výzkumný projekt na Dartmouthské univerzitě, na kterém pracovali John McCarthy, Marvin Minsky, Nathaniel Rochester a Claude Shannon [36].

Konkrétně pro strojového učení je zásadní rok 1957, kdy byl Frankem Rosenblattem vynalezen perceptron [18]. Poprvé tak mohlo dojít k učení stroje na příkladech za účelem vykonání určitých úloh. Tento objev okamžitě následovaly překvapivě dobré teoretické výsledky. V následujících několika letech výzkumu se ale nepodařilo vytvořit úspěšnou aplikaci, která by byla použitelná v praxi. Důvodem byly vysoké výpočetní požadavky, které překračovaly technologické možnosti 60. let [41]. Další zlom nastal v roce 1974, kdy Paul Werbos vyvinul algoritmus zpětné propagace, který umožnil trénovat vícevrstvé neuronové sítě. Takové sítě jsou na rozdíl od jednovrstvých schopné řešit i mnohem složitější problémy [25]. Obecně velký posun strojového učení začal až v 90. letech díky technologickému pokroku. Od té doby nabírá na popularitě a dochází ke stále rostoucímu vývoji [5].

V současné době strojové učení čítá velké množství metod a díky své síle je využíváno v nejrůznějších odvětví. Jedná se například o bankovní a finanční sektor, dopravu, zdravotnictví, služby zákazníkům, zemědělství, kybernetickou bezpečnost, herní průmysl, kosmonautiku a další.

3.2 Metody učení

Počítačový program disponující učícím algoritmem má schopnost se učit z množiny příkladů nazývané trénovací data, přičemž každý z příkladů je reprezentován vektorem reálných čísel, kde každé číslo představuje určitou vlastnost. Tyto hodnoty jsou v praxi získávány ze skutečných objektů, kterým může být například člověk a jednotlivé vlastnosti pak výška, váha, velikost nohy, barva očí, teplota a tlak. Samotné učení pak probíhá v závislosti na zvolené metodě. Jedná se buď o učení s učitelem, učení bez učitele nebo takzvaný reinforcement learning neboli zpětnovazební učení [38].

3.2.1 Učení s učitelem

Trénovací data jsou tvořená dvojicemi. Vždy se jedná o vstup v kombinaci s hodnotou požadovaného výstupu. Cílem algoritmu je naučit se vztah mezi vstupními a výstupními daty trénovací sady.

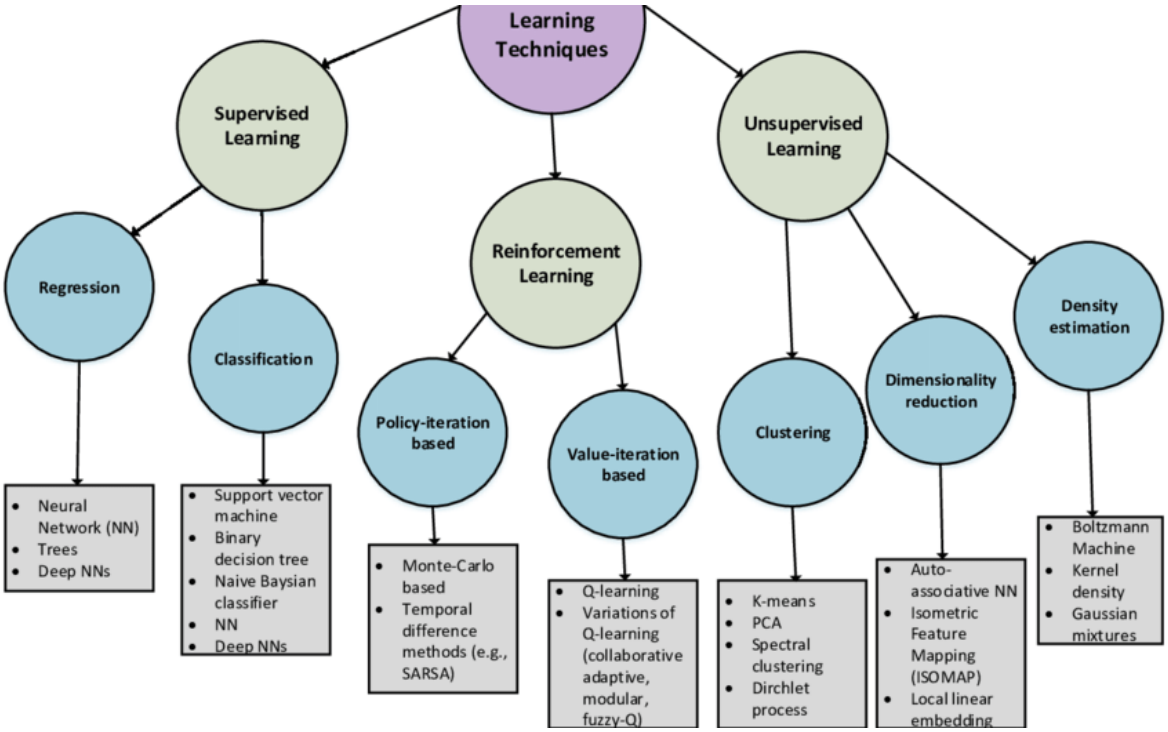
3.2.2 Učení bez učitele

Na rozdíl od kategorie učení s učitelem nedisponují vstupními daty s hodnotu požadovaného výstupu. Neexistuje tedy žádný konkrétní správný výsledek. Algoritmus se v tomto případě učí vnitřní strukturu dané tréninkové množiny, přičemž dochází k vytvářením relací a spojitostí mezi daty. Díky zmíněným vlastnostem je tento přístup možné použit i na velké trénovací sady.

3.2.3 Zpětnovazební učení

V tomto případě dochází k učení prostřednictvím udělováním odměn. Jsou známy počáteční i cílové body a úkolem systému je nalézt nejkratší správnou cestu. To je prováděno postupnými kroky v prostředí, za které jsou udělovány pozitivní nebo negativní odměny [5].

Každá z těchto tří metod učení je tvořena sadou různých algoritmů, které jsou schopny řešit typické úlohy jako je klasifikace, klasterizace, regrese a další. Specializací konkrétního problému je pak možné zúžit výběr a stanovit ze současné velké nabídky dostupných metod strojového učení ty pravděpodobně nejvhodnější na daný problém.



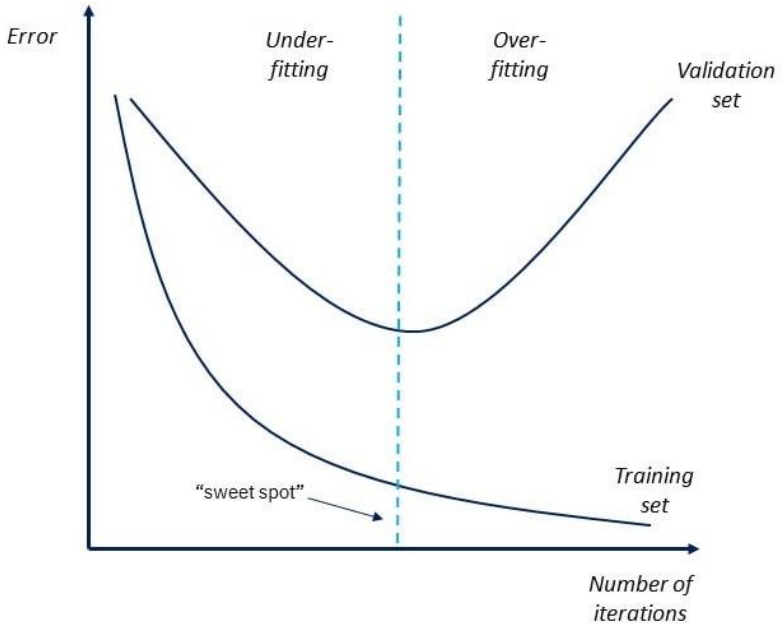
Obrázek 3.1: Rozdělení technik strojového učení [67]

3.3 Model

Soubor obsahující informace o tom, jak si program pamatuje to, co se naučil, a jakým způsobem to pak má být použito, se nazývá model. Ten je možné vyjádřit jako funkci $y = f(p, x)$, kde p jsou takzvané parametry modelu, díky kterým si program pamatuje, co se naučil. Funkce f pak představuje, jakým způsobem program pro daný vstup x vypočítá výstup y. V konkrétním případě je nutné f i počáteční stav p přesně specifikovat, protože v závislosti na tom budou vlivem procesu učení na trénovacích datech nalezeny ty nejlepší možné hodnoty p. Pro kontrolu je zavedena takzvaná chybová funkce, která má za úkol zachycovat úspěšnost modelu. Cílem je tedy minimalizace naměřené chyby nalezením vhodných parametrů p [38].

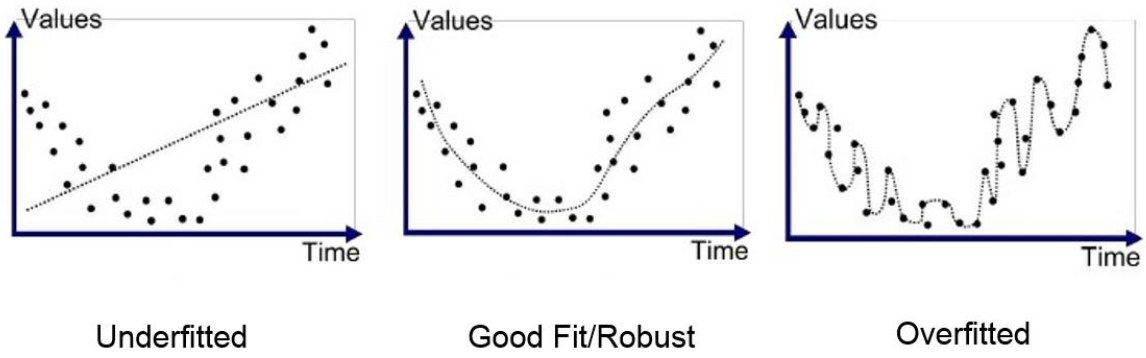
3.4 Přeučení a nedoučení

Každý model ke svému učení využívá trénovací data, aby dosahoval co nejlepších výsledků, přičemž jeden ze základních požadavků na model je jeho schopnost generalizace neboli nalezení obecně platných vazeb mezi vstupy a výstupy. Tato vlastnost je zásadní pro udržení úspěšnosti i po nasazení natrénovaného modelu na reálná data, která se v trénovací sadě nevyskytují. Aby nedocházelo k neočekávaným výsledkům po nasazení do praxe, je provedeno rozdělení množiny dostupných dat na trénovací a testovací. Obě tyto sady by měly být vhodně vytvořeny. Předpokládá se, že jednotlivé příklady jsou navzájem nezávislé a v obou sadách vzorkované se stejným pravděpodobnostním rozložením. Společně s tím je zavedena trénovací a testovací chyba. Během učení pak dochází současně ke kontrole úspěšnosti i na testovacích datech. Dobrý učící algoritmus by měl mít malou trénovací chybu a zároveň malý rozdíl mezi trénovací a testovací chybou. Pokud má model na konci učícího procesu velkou trénovací chybu, tak je nedoučený. Pokud má malou trénovací chybu, ale velký rozdíl mezi trénovací a testovací chybou, tak je přeučený. Ideální stav pro dosažení nejlepších výsledků se nachází v bodě, kdy testovací chyba začala růst [38].



Obrázek 3.2: Chybovost během procesu učení [12]

Další důležitou vlastností modelu je jeho kapacita, která vyjadřuje, jak složitá data je schopen se model naučit. V případě malé kapacity dochází k nedoučení, protože takový model nemá prostředky pro obsáhnutí složitosti dané úlohy, a tedy není schopen se dostatečně naučit ani trénovací sadu. Opačným extrémem je nadměrně vysoká kapacita, která může vést naopak k přeučení. Tento problém je však často řešitelný rozšířením tréninkové množiny [38].



Obrázek 3.3: Výsledky nedoučeného, dobře naučeného a přeučeného modelu v praxi [29]

3.5 Tiny Machine Learning

Tato podkapitola byla převzata z [30].

3.5.1 Vznik

První roky 21. století započaly exponenciální růst vývoje algoritmů strojového učení, což úzce souviselo s tehdejším technologickým pokrokem, který mimo jiné přinesl i výkonnější procesory a takzvaná velká dat. První vytvořené modely byly opravdu malé, protože výpočetní výkon zprostředkovával místní počítač disponující jedno nebo více jádrovým procesorem. Následoval přesun výpočtu na grafické karty, které jsou výhodné při práci s většími datovými sadami. Tento koncept byl podpořen zavedením cloudových služeb, které zprostředkovaly velký výpočetní výkon. Jednou z takových služeb je například Google Colaboratory. Další vývoj představoval výrobu tenzorových procesorových jednotek (TPU) a specifických integrovaných obvodů pro konkrétní aplikaci (ASIC). Tato zařízení dosahovala nejen mnohem většího výpočetního výkonu v porovnání s grafickými kartami, ale navíc byla rozšířena o schopnost distribuce učení mezi více systémy za cílem použitelnosti větších modelů. Tento trend zvětšování úlohy a výpočetního výkonu vyústil v roce 2020 vydáním GPT-3, což je největší neuronová síť, která kdy byla vytvořena. Obsahuje 175 miliard neuronů, dva krát více než lidský mozek, a je odhadováno, že její trénink spotřeboval přibližně 3 GWh elektrické energie. Tolik energie vyprodukuje průměrná jaderná elektrárna za 3 hodiny provozu [13].

Zvyšující se energetické požadavky a kritika s nimi spojená přiměly komunitu umělé inteligence zaměřit se na energeticky úspornější zařízení. Tento směr zahrnující vývoj efektivnějších algoritmů, výpočtů a reprezentace dat byl v minulosti i přes velký potenciál přehlížen, protože tehdejší úsporná zařízení nebyla dostatečně rozšířená a nedosahovala ani potřebných technologických možností pro využívání strojového učení. To se ale změnilo a vznikla nová oblast zvaná Tiny Machine Learning.

3.5.2 Podstata a benefity

Tiny Machine Learning představuje průnik vestavěných zařízení internetu věcí a strojového učení. Z hlediska celkového konceptu vychází TinyML z kritiky hlavní myšlenky IoT, jejíž podstatou je odesílání dat z místních zařízení do cloudu pro následné zpracování. Vznešené problémy a pochybnosti tohoto principu, které napomohly vývoji, se týkaly především soukromí, latence, ukládání dat a energetické účinnosti.

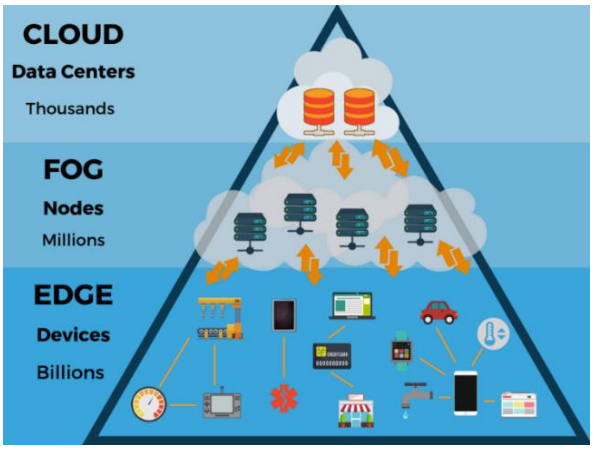
Soukromí
Při každém přenosu dat vzniká riziko jejich zachycení nějakou další stranou, což představuje narušení soukromí a další hrozby s tím spojené. Minimalizace této komunikace je jeden ze způsobů, jak snížit riziko případných zásahů do soukromí.

Latence
Hlavní myšlenkou IoT je již zmíněné odesílání dat do cloudu, kde jsou data zpracována a následně odeslána zpátky. Tento proces vyžaduje nějaký čas, který je dán především rychlostí internetu. Při pomalém připojení může vznikat tak vysoká latence, že se některé systémy stanou nepoužitelnými. Typickým příkladem jsou hlasoví asistenti, u kterých je rychlá odezva zásadní. V případě zpracování přímo na koncovém zařízení by tak došlo k odstranění nebo zásadnímu snížení potřebné komunikace a s tím spojené latence.

Ukládání dat
Existuje mnoho zařízení, která pracující s velkým množstvím dat obsahujících jen malé procento těch s nějakou vypovídající hodnotu. Konkrétním případem jsou záznamy z bezpečnostních kamer, kde většina záznamu představuje statický obraz bez nějakého dění. Implementací větší míry inteligence by v tomto případě vedlo k záznamu pouze klíčových chvil, což by zásadním způsobem snížilo požadavky na kapacitu úložiště a množství dat pro případný přenos.

Energetická účinnost
Přenos dat jakýmkoliv způsobem je energeticky náročná úloha. V porovnání s výpočty jde často o řádově větší množství požadované energie. To se projevuje u většiny bateriově napájených IoT zařízení, jejichž životnost na jedno nabití je primárně udávána frekvencí přenosu dat. Nejvíce efektivní je tedy vývoj zařízení umožňující provádět požadované výpočty bez nutnosti odesílání dat na výkonnější uzly.

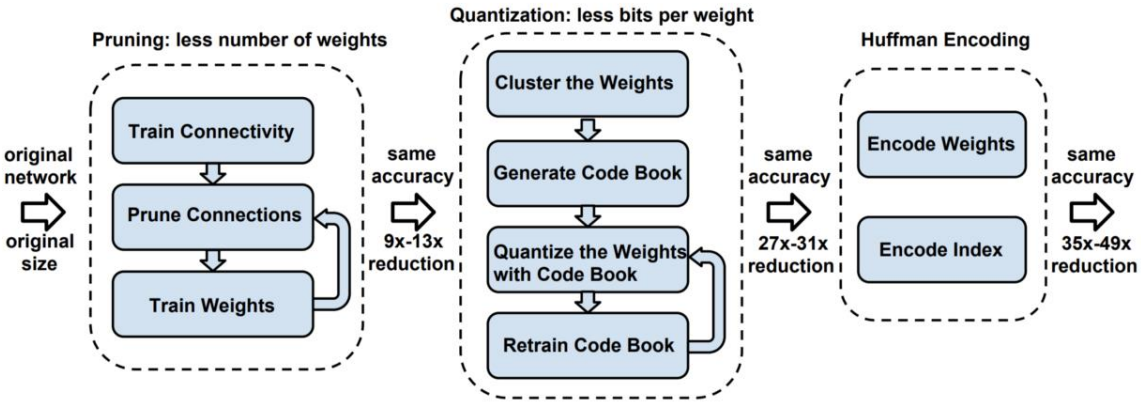
Podstatou je tedy přesun výpočtu do oblasti energeticky nenáročné výpočetní techniky a takzvaného edge computingu neboli souboru výpočetních zařízení, která se v pomyslné IoT infrastruktuře nachází nejblíže zdroji dat. Taková zařízení často disponují velmi malým výpočetním výkonem v kombinaci s pamětí v řádech maximálně jednotek megabytů, a proto je pozornost věnována především vývoji efektivnějších algoritmů a datových struktur, včetně technik s tím spojených. To je aplikováno i na modely, což má za důsledek zásadní redukci jejich velikosti bez vlivu na výslednou přesnost.



Obrázek 3.4: Hierarchie výpočetních zařízení z pohledu internetu věcí

3.5.3 Princip funkce

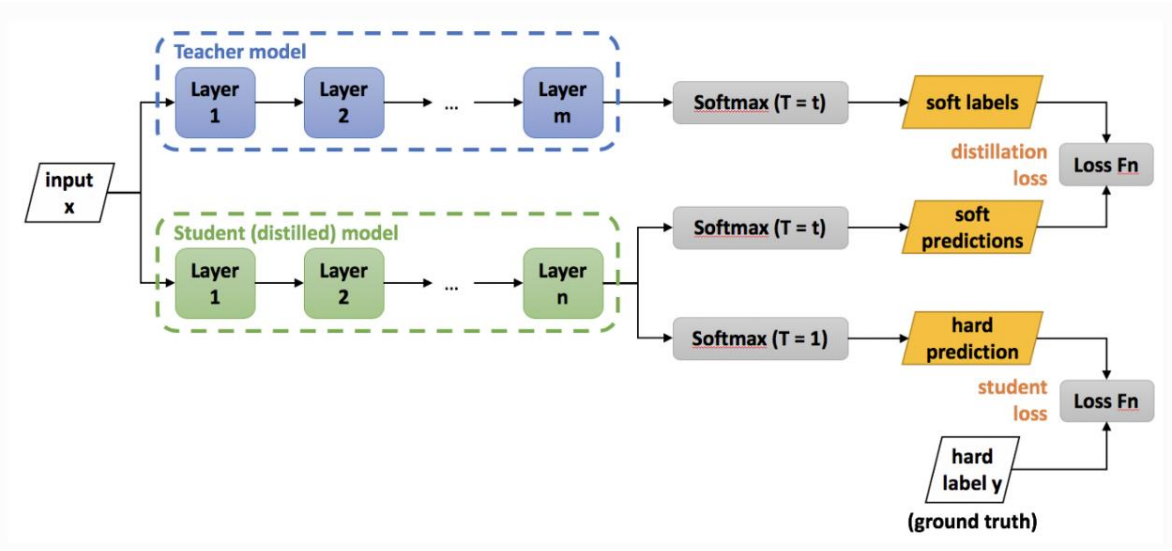
TinyML se z velké části neliší od klasického strojového učení. Využívají se stejné algoritmy pro vytvoření modelů a ty se pak standardně trénují na počítači nebo v cloudu. Hlavní rozdíl nastává až po dokončení tréninku, kdy začíná takzvaná hluboká komprese. To je proces, během kterého jsou na natrénovaný model aplikované techniky za účelem zásadní redukce velikosti bez vlivu na jeho přesnost. Konkrétně se jedná o posloupnost tří úprav složených z prořezávání, kvantování a kódování.



Obrázek 3.5: Hluboká komprese

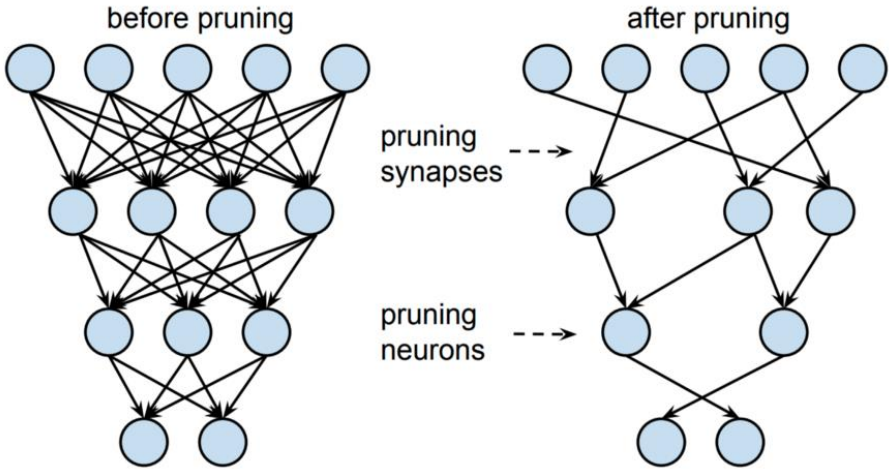
Prořezávání

Tato část zahrnuje kromě samotného prořezávání také metodu takzvané destilace. Hlavní myšlenka destilace je založena na tom, že rozsáhlé sítě s velkou kapacitou disponují nějakou mírou redundance, protože možnosti kapacity nejsou plně využity. Taková síť pak může být nahrazena sítí s menší kapacitou dosahující stejných výsledků z pohledu přesnosti. Proces destilace konkrétně složí k přenesení všech znalostí původního modelu označovaného jako učitel na nový studentský model s menším počtem parametrů. Dochází tak k minimalizaci velikosti původní sítě.



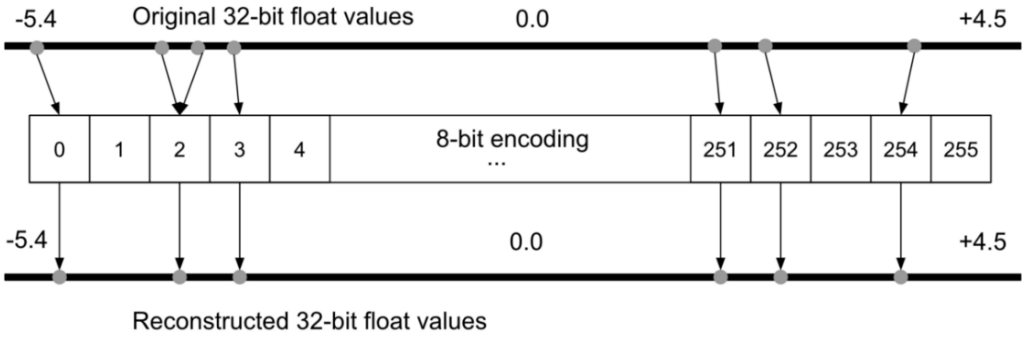
Obrázek 3.6: Proces destilace

Druhou metodou pro redukci velikosti modelu je prořezávání. To má za úkol nejprve odstranit části, které mají malý vliv na celkovou výstupní hodnotu, a poté dotrénovat ořezaný model tak, aby jeho výstup odpovídal výstupu původního modelu.



Obrázek 3.7: Proces prořezávání

Kvantování
Tato fáze slouží ke konverzi původního modelu do formátu vhodného pro cílovou architekturu. Důvodem je nekompatibilita, protože k vytváření modelů probíhá na počítačích s 64bitovou architekturou, zatímco cílová zařízení pracují většinou na 32bitové nebo 8bitové architektuře. Společně s tím dochází k redukci paměťových nároků, které jsou v případě kvantování 64bitových na 8bitové hodnoty sníženy na osminu.



Obrázek 3.8: Proces kvantování

Protože během převodu může dojít ke ztrátě informace vlivem chyby kvantizace, byl vytvořen speciální druh tréninku, který během procesu učení dovoluje síti používat pouze hodnoty dostupné na cílovém zařízení. Díky tomu nedochází k poklesu přesnosti výsledného modelu.

Kódování
Tato volitelná technika je aplikována v případě, kdy je minimalizace velikosti modelu zásadním požadavkem. Nejčastěji je k tomu využito Huffmanovo kódování, které zajistí ukládání dat s maximální efektivitou z pohledu paměťové náročnosti.

Po dokončení hluboké komprese je výsledný model převeden do takové interpretovatelné podoby, aby jej bylo schopné cílové zařízení používat. Typicky se jedná o reprezentaci pomocí pole hodnot v programovacím jazyce C.

4 Výběr vhodných technik pro detekci pohybu prutu

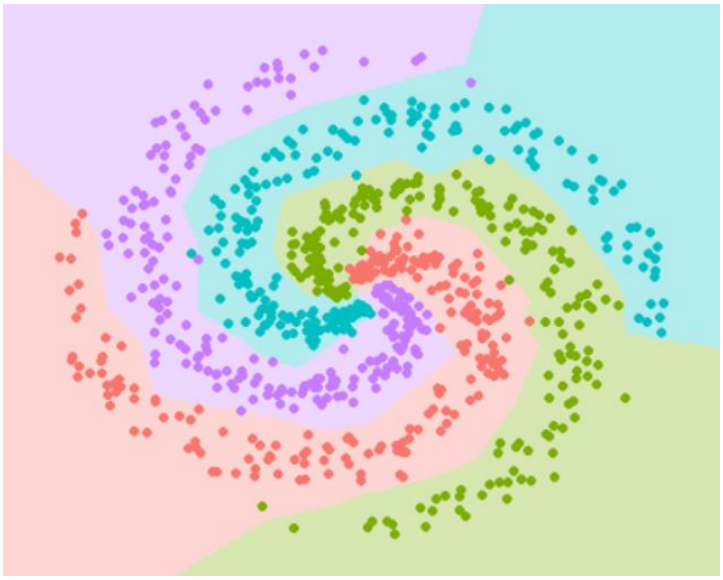
V této kapitole je nejprve specifikována úloha detekce pohybu prutu. V závislosti na tom je představena klasifikace jako typický problém v oblasti strojového učení. Následuje popis inerciální měřící jednotky včetně dat, která zaznamenává. Zmíněný je problém rozpoznávání lidské činnosti společně s jeho různými způsoby řešení. Na závěr jsou z dostupných informací stanoveny a popsány vhodné algoritmy strojového učení pro detekci pohybu prutu.

4.1 Specifikace úlohy

Na feederový prut bude pevně připevněno zařízení disponující inerciální měřící jednotkou, která bude snímat jeho pohyby v reálném čase. Získaná data pak budou následně zpracována vhodným algoritmem strojového učení, který určí, jestli došlo k nějakému z klíčových pohybů spadajících do předem stanoveného seznamu. Hlavním úkolem vybraného algoritmu bude tedy řešit typickou úlohu klasifikace.

4.2 Klasifikace

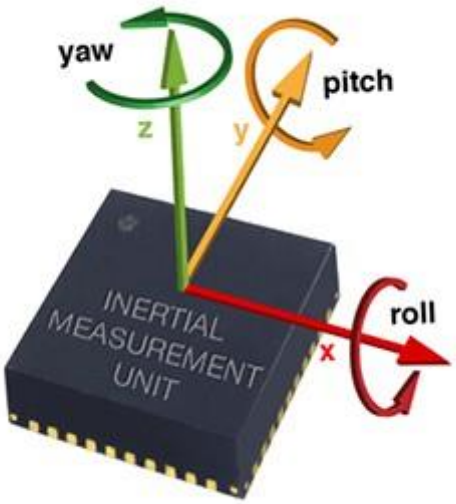
Jedná se o druh problému, kde je cílem zařadit vstupní data do jedné nebo více tříd, které jsou přesně specifikovány v trénovací sadě. Nejjednodušším případem je binární klasifikace požadující pouze dvě třídy, jednu pozitivní a druhou negativní. Typickým příkladem je označování e-mailů za spamy [2]. Mezi příklady s větším počtem tříd patří klasifikace obrázků a dokumentů, kategorizace produktů, predikce chování zákazníků, rozpoznávání lidské činnosti a další. V současné době existuje mnoho klasifikačních algoritmů, kde mezi ty známější patří lineární klasifikátory, logistická regrese, naivní Bayesův klasifikátor, perceptron, metoda podpůrných vektorů známá pod zkratkou SVM, kvadratické klasifikátory, rozhodovací stromy, náhodný les, neuronové sítě, Bayesovské sítě a další [39]. Obecně však není možné stanovit nejlepší metodu, protože úspěšnost každé z nich závisí na konkrétním problému a datové sadě.



Obrázek 4.1: Vzorky rozdělené do jednotlivých barevně oddělených tříd [30]

4.3 Inerciální měřící jednotka

Inerciální měřící jednotka neboli IMU je elektronické zařízení obsahující kombinaci několika gyroskopů, akcelerometrů a případně dalších doplňkových senzorů jako je například magnetometr nebo senzor teploty. Jedná se tak o poměrně komplexní snímač sloužící především k měření pohybu ve třírozměrném prostoru. Většina standardních IMU proto disponuje trojicí akcelerometrů a gyroskopů, kde každá dvojice pokrývá jednu ze tří vzájemně kolmých os. Typickým výstupem takové jednotky je úhel náklonu a zrychlení ve všech směrech [8].



Obrázek 4.2: Inerciální měřící jednotka [62]

4.4 HAR – Rozpoznávání lidské činnosti

Strojové učení mimo jiné přineslo skokové zlepšení úspěšnosti řešení úloh v nejrůznějších oblastech a zároveň otevřelo dveře možností k řešení komplikovanějších a dříve neřešitelných problémů. Jedním z takových je i rozpoznávání lidské činnosti, které již bylo různými způsoby vyřešeno. Dosažený výzkum tak poskytuje dobrou představu o současných možnostech, které budou v práci použity jako výchozí bod, protože rozpoznávání pohybů prutu má s touto úlohou částečnou podobnost.

Konkrétně se jedná o klasifikaci sekvencí pohybových dat zaznamenaných nejčastěji mobilním telefonem, přičemž výsledkem je jeden z předem definovaných pohybů typu běh, chůze, sezení, ležení, chůze po schodech a podobně. Prvním problémem této úlohy je požadavek na zpracování velkého množství dat. Většinou se jedná o desítky až stovky hodnot v každém ze tří směrů během jedné vteřiny, které je potřeba v reálném čase vyhodnocovat. Druhý problém představuje samotné přiřazení konkrétního pohybu ke vstupním datům, protože neexistuje žádný obecný analytický způsob vyhodnocení tak velkého množství informací s dostatečnou rychlostí. Vzhledem ke zmíněným problémům je vhodným nástrojem na tuto úlohu hluboké učení, které díky své schopnosti naučit se funkce vyššího řádu dosahuje nejlepších výsledků. Mezi ty nejúspěšnější a často používané architektury konkrétně patří konvoluční a rekurentní neuronové sítě [21] [26].

4.5 Vhodné algoritmy

V kapitole 4.2 byl zmíněno několik známých algoritmů strojového učení vhodných pro řešení úlohy klasifikace. Ze všech těchto potenciálních adeptů na použití pro rozpoznávání pohybů prutu se ale jako nejlepší volba jeví hluboké učení.

Jedná se o podmnožinu strojového učení založené na umělých neuronových sítích, které díky svým vlastnostem zvládají řešit i náročnější úlohy. Navíc oproti jiným algoritmům strojového učení nevyžadují žádné předzpracování pro zvýraznění charakteristických znaků dat před samotným učením, protože tento proces extrakce zajišťuje neuronová síť sama. Předzpracování totiž představuje obrovskou nevýhodu, protože se jedná o ruční provedení, které je odkázané pouze na lidské znalosti dané oblasti, a proto umožňuje zaměření se pouze na specifický úkol a extrakci jen signifikantních rysů [27]. Pro obecnější použití nebo data s obtížně detekovatelnými rozdíly, které bude pohyb prutu pravděpodobně generovat, je nejsilnějším nástrojem právě hluboké učení. Všechny tyto zmíněné výhody jsou podmíněny požadavkem na vyšší výpočetní výkon a velké množstvím trénovacích dat, což by ale nemělo představovat zásadní limitující faktory.

Volbu hlubokého učení podporují informace v kapitole 4.4, kde je představena úloha rozpoznávání lidské činnosti, která má určitou podobnost s úlohou rozpoznávání pohybů prutu. Mimo jiné jsou také zmíněny použité algoritmy, přičemž mezi ty nejúspěšnější patří konvoluční a rekurentní neuronové sítě.

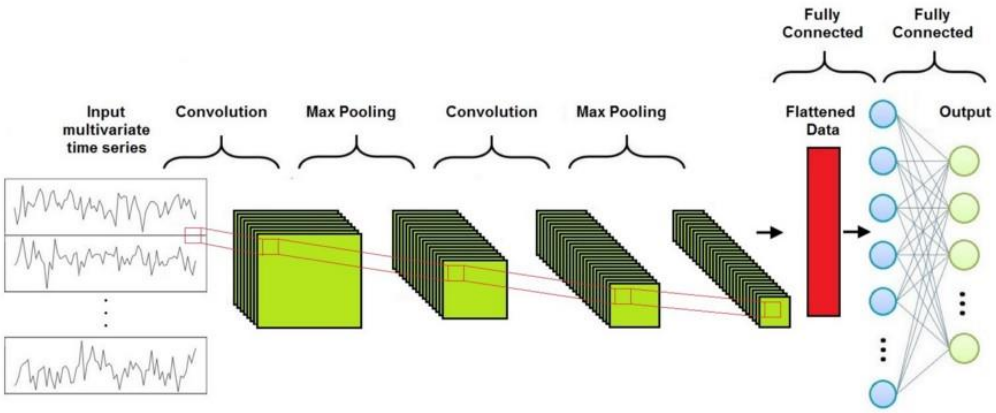
Dalším zdrojem informací je tento výzkum [27] zabývající se přímo použitím hlubokého učení k rozpoznávání činností člověka za použití dat ze senzoru pohybu. V práci bylo testováno několik různých modelů a na základě informací ze 4000 provedených experimentů na různých veřejně dostupných datových sadách bylo v kombinaci s dalšími dosaženými poznatky stanoveno, že k rozpoznávání krátkých aktivit s daným pořadím v určitém čase je vhodné použít LSTM, zatímco na dlouhodobě se opakující aktivity je nejlepší volbou konvoluční neuronová síť.

Jako nejlepší algoritmy pro rozpoznávání pohybů prutu se tedy jeví konvoluční neuronová síť a rekurentní neuronová síť typu LSTM.

4.5.1 Konvoluční neuronové sítě

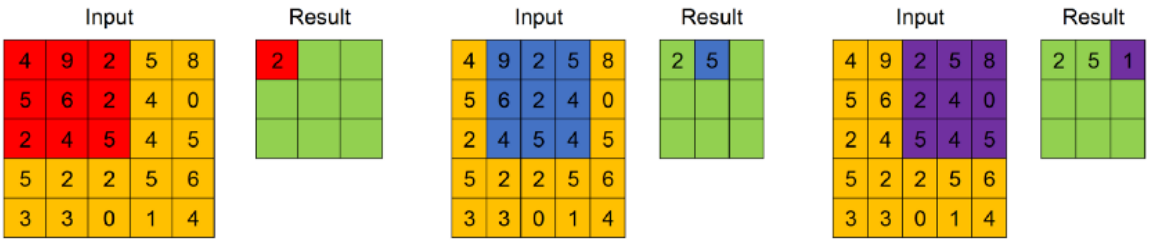
Tato podkapitola byla převzata z [44].

Jedná se o algoritmus hlubokého učení, který je hojně využíván k analýze obrazových dat. Nejčastějším vstupem jsou tedy snímky, ale může se jednat i o vícerozměrné časové řady, protože předností konvolučních neuronových sítí je jejich schopnost detekce prostorových a časových vzorů. Vždy se ale jedná o vstup s dvourozměrnou strukturou dat, čemuž je také vhodně přizpůsobena vnitřní architektura neboli uspořádání jednotlivých neuronů. Jako celek je možné konvoluční neuronovou síť popsat obecně jako posloupnost tvořenou skládáním konvoluční vrstvy, pooling vrstvy a plně propojené vrstvy.



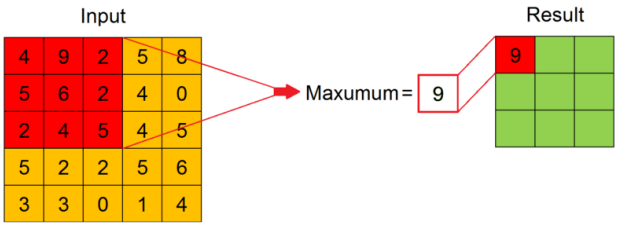
Obrázek 4.3: Konvoluční neuronová síť

Konvoluční vrstva je základem celé této sítě, jejímž hlavním úkolem je extrahovat klíčové rysy nacházející se ve vstupních datech. K tomu využívá operaci konvoluce, kterou aplikuje na každou část vstupních dat a určitý filtr, přičemž jednotlivé vrstvy zpravidla obsahují několik takových filtrů s různými hodnotami. Během tréninku jsou pak tyto hodnoty automaticky upravovány tak, aby docházelo k úspěšnější extrakci důležité informace a tím i k maximalizaci přesnosti celého modelu.



Obrázek 4.4: Aplikace operace konvoluce

Další v pořadí je pooling vrstva, která s minimální ztrátou informací redukuje rozměry výstupní mapy konvoluční vrstvy. To je prováděno postupným posunem okna pevné velikosti po celé mapě s tím, že v každém kroku je vybrána buď průměrná, anebo maximální hodnota v daném okně. Obvykle bývá lepší volbou ta maximální, protože potlačuje šum. Výsledkem je pak menší počet parametrů a výpočtů, což také snižuje riziko přetrénování.



Obrázek 4.5: Aplikace operace poolling

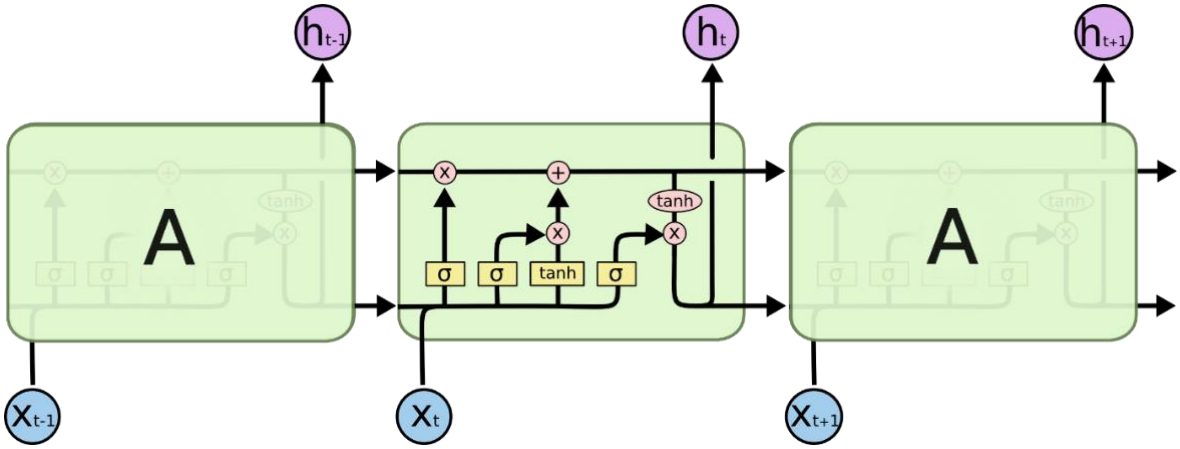
Výstup několika po sobě jdoucích dvojic konvoluce a poollingu je transformován do sloupcového vektoru představující vstup pro plně propojenou vrstvu. Ta je zpravidla implementována pomocí vícevrstvého perceptronu, který během tréninku využívá zpětnou propagaci chyby k dosažení maximální přesnosti modelu. Hlavním úkolem této části sítě je klasifikace dat do jedné z předem definovaných tříd na základě nalezených dominantních rysů. Výstupem je tedy vektor s počtem složek rovnající se počtu neuronů výstupní vrstvy plně propojené sítě, kde každá složka vektoru představuje pravděpodobnost příslušnosti do dané třídy.

4.5.2 Long Short Term Memory sítě

Tato podkapitola byla převzata z [37].

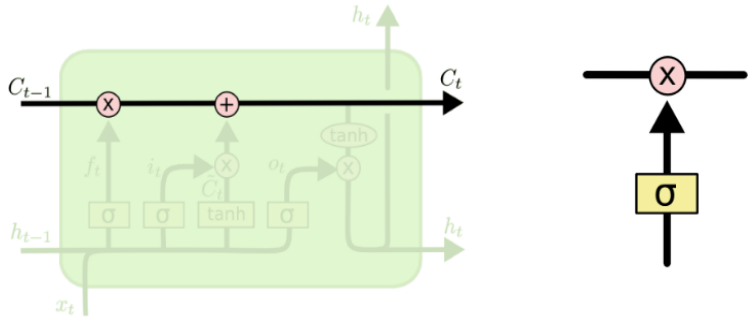
Jedná se o speciální druh rekurentních neuronových sítí, které vynikají schopností učit se dlouhodobé závislosti. Jejich specializaci proto představuje zpracování posloupnosti hodnot, přičemž hlavní myšlenkou je využití závislosti současných dat na těch předchozích. Od tohoto principu se odvíjí i celá architektura sítě. Ta je složená z na sebe navazujících opakujících se modulů, které aplikují stejnou funkci na každý prvek posloupnosti s tím, že jejich výstup je závislý na těch předchozích. Vnitřní strukturou se však LTSM od běžných rekurentních sítí odlišují, protože byly vyvinuty tak, aby se vyhnuly problémům s dlouhodobými závislostmi. Jeden modul tedy obsahuje

speciálně spolupracující prvky ze čtyř vrstev namísto jedné, díky čemuž mají tyto sítě již zabudovanou schopnost učit se, takže nevyžadují žádný složitý proces trénování.



Obrázek 4.6: Long Short Term Memory neuronová síť

Po stránce funkčnosti je základním prvkem LTSM takzvaný stav buňky představující pomyslný dopravní pás procházející přes všechny zřetězené moduly s tím, že na jeho cestě dochází k malým lineárním interakcím. Jedná se o přidání nebo odstranění informace ze stavu buňky pomocí takzvaných bran. Ty jsou díky kombinaci sigmoidní vrstvy neuronové sítě a operace násobení schopné vygenerovat hodnotu mezi 0 a 1, která stanoví, kolik toho z dané komponenty bude propuštěno dál.



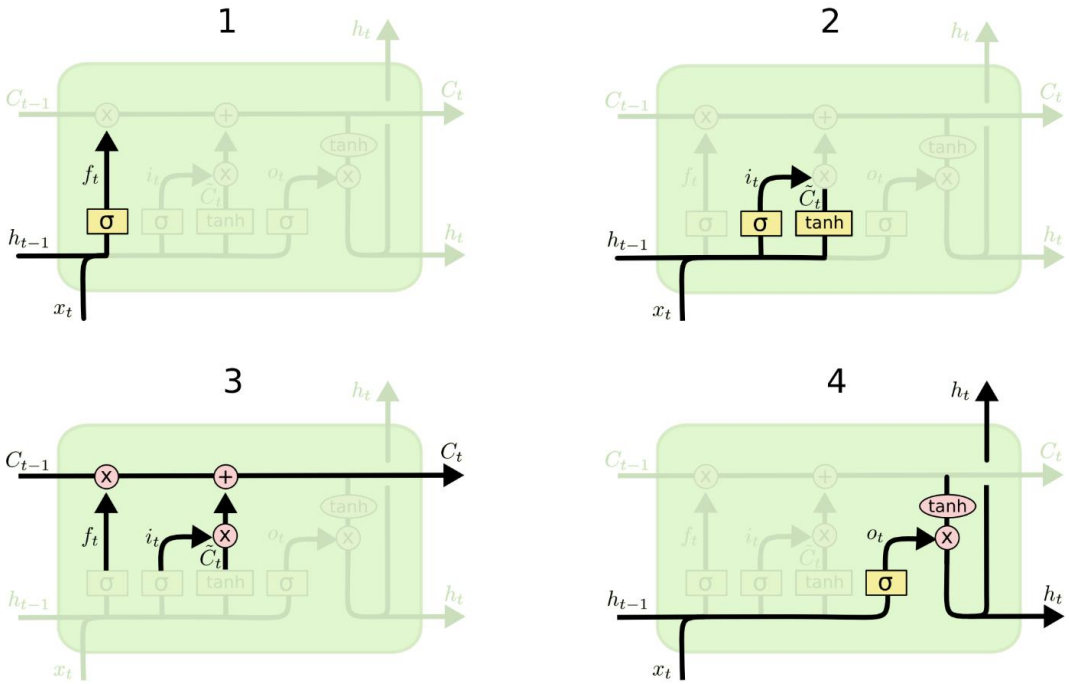
Obrázek 4.7: Vlevo stav buňky a vpravo brána

Prvním krokem při práci této sítě je rozhodnutí, jaké informace budou ze stavu buňky odstraněny. To zajišťuje zapomínající vrstva, která na základě výstupu předchozího modulu h_{t-1} a současného vstupu x_t vygeneruje číslo mezi 0 a 1 určující, jak důležité je zapamatovat si hodnotu stavu buňky z předchozího modulu.

Druhým krokem je určit, jaké nové informace budou ve stavu buňky uloženy. To je zprostředkováno kombinací dvou vrstev. První z nich generuje vektor hodnot $\hat{C}_t$ představující kandidáty pro přidání do stavu buňky a druhá, nazývaná jako vrstva vstupní, rozhoduje, které konkrétní hodnoty budou aktualizovány.

Následuje aplikace obou předchozích kroků. Nejprve dojde k aktualizaci starého stavu buňky C_{t-1} na podobu nového stavu C_t vynásobením f_t neboli zapomenutím určitých informací. Pak jsou přidány hodnoty vygenerovaných kandidátů, které jsou zmenšeny na základě rozhodnutí, jak moc je potřeba určité hodnoty aktualizovat.

Poslední krok představuje specifikaci, co bude výstupem. Nejprve je tedy rozhodnuto sigmoidní vrstvou, jaké části stavu buňky budou použity pro výstup. Poté je aplikována funkce tanh na hodnoty stavu buňky, aby byl jejich rozsah od -1 do 1. Nakonec jsou oba tyto vektory spolu vynásobeny, což zajistí vhodnou podobu výstupu složeného jen z požadovaných hodnot.



Obrázek 4.8: 4 kroky popisující práci LTSM

5 Výběr komponent

Volba konkrétního hardwaru se vždy odvíjí od požadavků na výsledné měřící zařízení, a proto je žádoucí nejprve vytvořit seznam těch základních vlastností, od kterých se bude následně odvíjet výběr vhodných součástek. Zařízení by mělo splňovat následující:

- Možnost jednoduchého připevnění k rybářskému prutu
- Snímání pohybu prutu pomocí IMU
- Detekce pohybu v reálném čase pomocí strojového učení
- Bezdrátové odesílání výsledků
- Napájení z baterie
- Možnost dobíjení

Na základě těchto vlastností bylo stanoveno, že se zařízení bude skládat ze 4 hlavních částí – mikrokontroléru, inerciální měřící jednotky, napájecího obvodu a krabičky včetně mechanismu pro upevnění. Zbytek kapitoly je věnován první trojici. Konkrétně jejímu podrobnějšímu popisu včetně výběru vhodných komponent a představení klíčových vlastností každé z nich. V neposlední řadě je zde také nastíněno, jak spolu jednotlivé části souvisí, a jak budou navzájem spolupracovat.

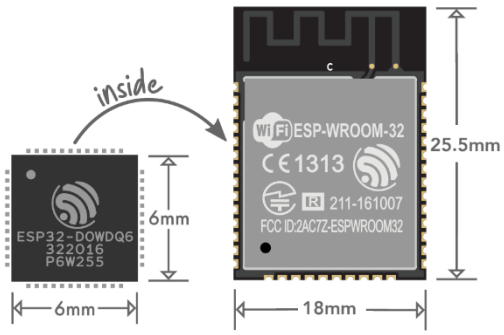
5.1 Mikrokontrolér

Mikrokontrolér je tou nejdůležitější komponentou celého projektu, a proto je její správný výběr zásadním úkolem, který je potřeba provést přednostně a s dostatečným uvážení všech požadavků, možností a případných budoucích modifikací, problémů a úprav.

Samotný proces výběru začal aplikací té nejvíce specifické podmínky, kterou je podpora strojového učení. Konkrétně se jednalo o nalezení knihovny Tensorflow Lite, která jako jedna z mála umožňuje nasadit modely strojového učení na různé mikrokontroléry. Z dostupných informací na oficiálních stránkách je možné se dočíst, že je vyžadována 32bitová platforma, přičemž testování bylo provedeno na mnoha procesorech založených na architektuře Arm Cortex-M, ESP32 a dalších [60]. Protože se bude jednat o náročnější výpočty v reálném čase, tak bylo potřeba zaměřit se na výkonnější z nich. Výběr byl tedy zúžen pouze na procesory Arm Cortex M4, Arm Cortex M7 a ESP32, které v testu výkonnosti Coremark dosahují nejlepších výsledků [55] [57]. Konečnou volbou bylo ESP32, konkrétně modul ESP32-WROOM-32E-N16, a to především díky WiFi a Bluetooth umístěných přímo na čipu, dvoujádrovému procesoru, počtu pinů, rozhraní a funkcí, velké komunitě, nízké ceně, a z pohledu této práce i absenci zásadních nedostatků [15].

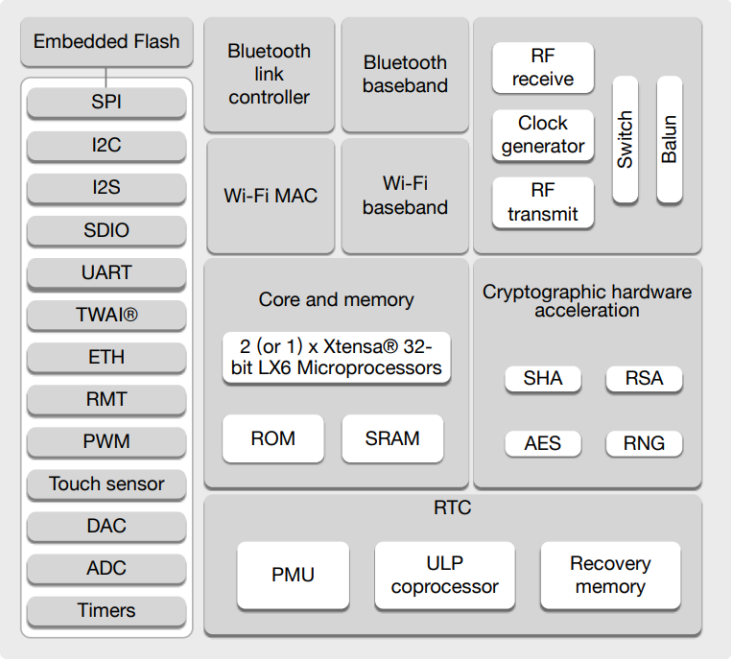
ESP32

ESP32 je dostupné ve třech formách: čip, modul a vývojová deska. Modul je deska plošných spojů obsahující čip ESP32, externí flash paměť, anténu a některé moduly i externí paměť PSRAM. Celkově je tato součástka navržena tak, aby bylo možné ji pohodlně integrovat na jiné desky plošných spojů.



Obrázek 5.1: ESP32 čip a modul [14]

Z hlediska specifikace disponuje zvolený model ESP32-WROOM-32E-N16 dvoujádrovým 32bitovým mikroprocesorem Xtensa® LX6 na frekvenci až 240 MHz, 520 KB interní paměti SRAM a 16 MB externí flash pamětí. Signifikantním rysem je přítomnost obvodů pro komunikaci na frekvenci 2,4 GHz, a to jak ve standardu Wi-Fi 802.11 b/g/n s propustností až 150 Mb/s, tak také ve standardu Bluetooth včetně energeticky úsporné verze BLE. Dalším specifikem je 26 programovatelných vstupně-výstupních pinů, které mohou volitelně zastávat i specifické funkce [15]. Mezi ně patří 12-bitový ADC převodník s až 18 kanály, dva 8-bitové DAC převodníky, 10 dotykových senzorů, Hallovo čidlo, několik rozhraní typu SPI, I^2C, I^2S, UART, ale i robustnější jako je SDIO, Ethernet, CAN 2.0 a další. Mimo jiné disponuje ESP32 několika módy pro řízení spotřeby. Ta se při běžném provozu pohybuje od 20 do 68 mA, při bezdrátové komunikaci to může být ve špičkách až 379 mA, ale použitím vhodného módu je možné minimalizovat spotřebu až na 5 µA. Všechny další vlastnosti a podobnosti je možné dohledat přímo v dokumentaci ESP32 [57].



Obrázek 5.2: Blokové schéma ESP32 [40]

5.2 Inerciální měřící jednotka

Další zásadní komponentou je samotná IMU pro sběr informací, která bude připojena k ESP32 pomocí vhodného rozhraní tak, aby mohla tento mikrokontrolér zásobovat potřebnými daty pro následné zpracování a vyhodnocení provedeného pohybu.

Jak už bylo zmíněno výše v kapitole 4.3, tak základem IMU je trojice akcelerometrů a trojice gyroskopů. Existují ale i verze rozšířené o další senzory. Ty však pro tento projekt nejsou vhodné, protože je vzhledem ke zpracování na málo výkonném mikrokontroléru nutné minimalizovat data ke zpracování. Společně s tím nebude ani potřeba specifických senzorů s velkou přesností, pokročilými možnostmi, extrémní odolností a podobně. Výběr byl tedy zúžen na oblast základních modelů s jediným zásadním požadavkem, a to zvolit IMU s dostatečně velkou vzorkovací frekvencí, aby nedošlo k nějakým budoucím limitacím. Vybrán byl konkrétně čip BMI160 od firmy Bosch Sensortec.

Důvodem volby byl především specializovaný výzkum, při kterém bylo otestováno a porovnáno několik často používaných IMU s podobnými vlastnostmi [21]. Z nich vyšel jako vítěz právě zvolený BMI160, který například při testování šumu dosahoval řádově lepších výsledků v porovnání se staršími, ale velmi často používanými MPU9150 a MPU9250.

BMI160
Jedná se o malou, úspornou inerciální měřící jednotku s nízkým šumem, která disponuje 16-bitovým tříosým akcelerometrem a 16-bitovým tříosým gyroskopem. Akcelerometr nabízí měření zrychlení v rozsahu od 2 do 16 g se vzorkovací frekvencí až 1600 Hz a gyroskop umožňuje měřit úhlovou rychlost v rozsahu 125 až 2000 °/s se vzorkovací frekvencí až 3200 Hz. Tyto parametry by tedy měly poskytovat velkou rezervu v souvislosti s výše stanovenou podmínkou o vzorkovací frekvenci, protože předpokládaný dostatečný počet vzorků za vteřinu by se mohl pohybovat okolo stovky. K dalším podstatným specifikům jednotky patří spotřeba 925 µA a podpora komunikačních rozhraní SPI a I^2C [65]. Pro zajímavost lze na závěr ještě doplnit, že BMI160 je mimo jiné integrována i v telefonech iPhone X [21].

5.3 Napájecí obvod

Tato část se věnuje baterii, jakožto zdroji energie pro celé zařízení, dále pak jejímu nabíjení, detekci stavu nabití, a nakonec konverzi napětí na požadovanou úroveň pro jednotlivé komponenty zařízení.

5.3.1 Baterie

Zařízení bude napájeno pouze z baterie umožňující opětovné dobíjení, a proto je nutné zvolit vhodný dobíjecí typ. Tím by i přes dnešní obrovský výběr mohl být Lithium-iontový polymer akumulátor (LiPo), který je v současné době pravděpodobně nejpoužívanějším zdrojem energie u malých přenosných zařízení. Druhou možností je Lithium-iontová baterie (Li-ion), která pracuje na podobném principu jako LiPo, s tím rozdílem, že elektrolyt uvnitř je ve stavu kapalném namísto pevném. Z toho důvodu jsou tyto baterie těžší, mají jasně daný tvar, ztrácí kapacitu v průběhu času, déle se nabíjí, ale jsou levnější díky jednodušší výrobě a mají vysokou hustotu energie [44]. Výběr konkrétního typu se pak odvíjel především od spotřeby a požadované doby výdrže zařízení na jedno nabití.

Obrázek 5.3: Vlevo 18650 Li-ion baterie [47] a vpravo LiPo baterie [7]

Hrubý odhad spotřeby proudu celého zařízení by se mohl pohybovat v průměru okolo 100 mA. Toto číslo se skládá z horní hranice pracovní spotřeby ESP32 představující 68 mA s občasnou skokovou hodnotou až na 379 mA během vysílání [15], která by ale vzhledem k dlouhým časovým rozestupům s krátkou dobou odesílání neměla spotřebu výrazným způsobem zvyšovat. Další nezanedbatelnou část odebíraného proudu v jednotkách miliampér bude představovat každá z LED diod. Těch bude zařízení obsahovat hned osm, ale dlouhodobě by měly být aktivní maximálně 2 z nich, což by společně s občasnou aktivitou těch ostatních šesti nemělo v průměru přesáhnout 20 mA. Poslední, avšak zanedbatelná je IMU s odběrem menším než 1 mA [65], stejně tak i případné další pomocné obvody. Celkovou orientační průměrnou hodnotu 100 mA není ani možné zásadním způsobem snížit použitím nějakého úspornějšího módu, kterými ESP32 disponuje, protože se bude jednat o zařízení pracující v reálném čase.

Požadovaná životnost na jedno nabití je dána délkou oficiálního feederového závodu, který se skládá ze dvou pětihodinových kol a tréninku o délce maximálně 10 hodin [56]. Celkový požadavek je tedy 20 hodin, čímž je také pokryt i případný celodenní lov.

Součinem proudu a času, tedy 100 mA a 20 hodin, je získána minimální požadovaná kapacita baterie 2000 mAh. Nejedná se tak o menší kapacitu do 1000 mAh, kde by použití malé, lehké, vhodně tvarované LiPo baterie s přesnou kapacitou bylo pravděpodobně nejlepší volbou. V tomto případě byla zvolena Li-ion baterie ACCU-INR18650-25R typu 18650 s napětím 3,6 V, protože má standardní rozměr a je velmi rozšířená, takže může být v případě potřeby jednoduše vyměněna. Mezi další výhody patří nízká cena a kapacita 2500 mAh, která představuje dostatečnou rezervu i pro případnou vyšší hodnotu reálně odebíraného proudu anebo ztrátu samotné kapacity v průběhu času [1].

5.3.2 Nabíjení

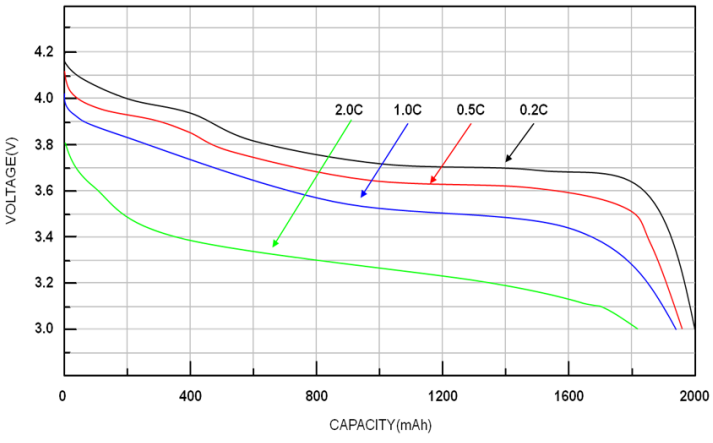
Použitá Li-ion baterie vzhledem ke svému složení vyžaduje určitá pravidla při nabíjení. Důležité je použít nabíječku s rozsahem 4,2 – 4,4 V, po nabití, a tím i dosažení maximálního napětí 4,2 V je nutné baterii okamžitě odpojit od zdroje a poslední podmínka se vztahuje na nabíjecí proud, jehož hodnota by neměla přesáhnout polovinu celkové kapacity baterie. To znamená, že baterie s kapacitou 2000 mAh by měla být nabíjena proudem o maximální hodnotě 1 A. Pokud nebude některá ze zmíněných třech podmínek dodržena, dojde k poškození baterie, a proto existují speciální nabíječky s obvody, které disponují ochranou proti přebíjení, přepětí a nadproudu [49].

Jednou z takových komponent je i čip MCP73830, jehož vnitřní struktura představuje kompletní nabíjecí obvod pro jednočlánkovou Li-ion nebo LiPo baterii, a proto bude i použit jako nabíjecí část celého výsledného zařízení. Jedním z jeho klíčových parametrů je schopnost nabíjení proudem o hodnotě až 1 ampér. Tím je vzhledem ke kapacitě baterie 2500 mAh splněna podmínka, aby byl v tomto případě proud menší než 1,25 A. Zároveň se ale jedná o dostatečně velkou hodnotu, aby došlo k rychlému nabití baterie. Druhou důležitou vlastností čipu je jeho vstupní napětí v rozsahu 3,75 až 6 V, což umožní realizovat nabíjení pěti volty přes standardní konektor USB-C [53].

5.3.3 Detekce stavu baterie

Důležitou vlastností každého bateriově napájeného zařízení je schopnost informovat uživatele o stavu zbývající kapacity baterie, respektive o předpokládané zbývající době provozu zařízení.

Jedním z nejjednodušších způsobů řešení je připojení vstupu ADC převodníku na baterii a v závislosti na napětí určit stav nabití. Právě tento přístup bude použit i zde, protože zvolený mikrokontrolér ESP32 disponuje 12-bitovým ADC převodníkem. Jistou nevýhodou je však menší přesnost výsledku vlivem převodu, protože je zpravidla používána obecná charakteristika, která se od té skutečné může lišit. Zároveň se také tato charakteristika liší i v závislosti na vybíjecím proudu, avšak pro účely tohoto projektu nehrají malé rozdíly stavu nabití zásadní roli, a proto se jedná o zanedbatelnou nevýhodu.



Obrázek 5.4: Vybíjecí charakteristika Li-ion baterie [33]

Aby vše správně fungovalo, tak je nutné přizpůsobit měřené napětí možnostem převodníku, který pracuje na rozsahu 0 – 3,3 V s tím, že jeho převodní charakteristika [9] není lineární po celé své délce, takže je vhodné se držet v rozsahu od 0,2 do 3,1 V. Napětí baterie by se ale mělo pohybovat od 2,5 do 4,2 V [1], takže je nutné použít napěťový dělič tvořený dvěma rezistory. Ty jsou pro jednoduchost zvoleny v poměru 1:1, čímž dojde ke snížení měřeného napětí na polovinu, tedy na rozsah 1,25 – 2,1 V, který již spadá do měřitelného spektra. Co se týče konkrétní velikosti rezistorů, tak se jako nejlepší volba jeví často používaná hodnota 10 kΩ, protože zajistí proud protékající děličem v řádu stovek μA, což je kompromisem představující zanedbatelnou spotřebu a dostatečný proud pro spolehlivé měření napětí.

5.3.4 Konverze napěťové úrovně

Celý obvod by měl pracovat na napětí 3,3 V, a proto je nutné provést konverzi bateriové úrovně 3,6 V. K tomu slouží napěťové regulátory, které z určitého napěťového rozsahu vytvoří konkrétní stabilní hodnotu. Při jejich práci ale dochází k určitým ztrátám energie, což zahrnuje i pokles napětí, pro který je v tomto případě prostor maximálně 0,3 V. Pro tyto případy však existují regulátory s malým úbytkem napětí označované zkratkou LDO. Volba té správné komponenty je ale spojena se stanovením požadovaných parametrů.

Vstupní napětí bude v rozsahu 2,5 – 4,2 V, což je dáno úrovní nabití baterie. Výstupní napětí bude 3,3 V, což je standardem pro ESP32, jehož pracovní rozsah je 2,3 – 3,6 V [57]. Zároveň je to i hodnota spadající do požadovaného rozsahu IMU BIS160 o napětí 1,71 – 3,6 V [65]. Je tedy zřejmé, že úbytek na regulátoru by měl být co nejmenší. V ideálním případě nulový a při poklesu napětí baterie pod 3,3 V by muselo dojít k okamžitému dobití, čímž by ani nedošlo k úplnému vybití a využití kapacity.

Ve skutečnosti však umožňuje většina LDO pracovat i s o něco nižším napětím, než je dáno součtem jejich výstupu a úbytku, a proto je možné použít LDO s úbytkem až 200 mV za podmínky, že jeho minimální podporované vstupní napětí je menší nebo rovné 2,5 V. Pak by v nejhorším případě nebylo výstupem požadovaných 3,3 V, ale nejspíš 2,3 V, které spadají do rozsahu úrovní jak ESP32, tak IMU.

Posledním klíčovým parametrem je výstupní proud. Ten by měl být alespoň 500 mA, což je výrobcem doporučovaná hodnota pouze pro samotné ESP32 [57], takže vzhledem k ostatním komponentám jako jsou především LED diody by měl být ještě vyšší.

Jako nejlevnější volba splňující všechny stanovené podmínky se jevil LDO s označením XC6210B332MR. Ten pracuje se vstupním napětím 1,5 – 6 V, má pevné výstupní napětí požadovaných 3,3 V, výstupní proud až 800 mA a úbytek napětí 50 mV na každých 100 mA výstupního proudu. V teoretické nejvyšší špičkové zátěži se spotřebou 800 mA bude tedy úbytek asi 400 mV, zatímco ve většině času bude při předpokládaném odběru 100 mA pokles napětí pouhých 50 mV. Důležitá je také závislost výstupního napětí na vstupním napětí, která mimo jiné udává, jak se změní výstup v závislosti na poklesu vstupu pod pevně danou výstupní hodnotu napětí. V tomto případě by se mělo jednat o lineární intuitivní pokles. Pokud se tedy úroveň napětí baterie dostane až na spodní hranici 2,5 V, výstupní napětí bude stejná hodnota snížená o úbytek napětí na regulátoru v závislosti na proudu, takže při standardním odběru 100 mA se bude jednat o 2,45 V [64]. Z tohoto pohledu je nejvíce kritická oblast pokryta s dostatečnou rezervou, protože požadovaná spodní hranice je 2,3 V. Podmínkou je však použití dostatečně velkých blokovacích kondenzátorů pro eliminaci případného špičkového odběru, který by mohl představovat problémy v případě nízkého stavu baterie.

6 Hardwarová realizace

Tato kapitola se věnuje procesu vytvoření konečné podoby zařízení. Nejprve práce probíhá v programu KiCad, ve kterém je vytvořeno schéma zapojení jednotlivých součástek a závislosti na něm pak i navržena deska plošných spojů. Následuje navržení vhodné krabičky s upevňujícím mechanismem v programu Tinkercad. Nakonec je vyrobená deska plošných spojů osazena zvolenými součástkami a umístěna do krabičky vytisknuté na 3D tiskárně.

6.1 Schéma zapojení

Postupně jsou podle doporučených zapojení a dostupných informací v příslušných dokumentacích zapojeny jednotlivé součástky vybrané v kapitole 4. Dochází také k přidávání některých dalších komponent jako jsou tranzistory, kondenzátory, rezistory, LED diody a další. Nakonec jsou všechny tyto dílčí části spojeny v jeden celek a vzniklá tak výsledné zapojení celého zařízení, které je demonstrováno zjednodušenou formou v podobě blokového schéma.

6.1.1 ESP32

Tou nejkomplexnější komponentou z hlediska zapojení je ESP32-WROOM-32E-N16. V dokumentaci od výrobce je možné dohledat, že tento modul disponuje zapínacím pinem EN a několika strapping piny, jejichž hodnoty při startu volí mód, ve kterém bude ESP32 pracovat [15]. K zapnutí čipu dojde připojením pinu EN k napájení. Aby ale nedocházelo k problémům spojeným s nedostatkem napájení při startu nebo nedostatečnému času pro dosažení požadovaných hodnot na určitých pinech, je doporučeno přidat externí zpožďovací RC článek s hodnotami 10 kΩ a 1 µF. Co se týče strapping pinů, tak pro standardní chod není potřeba přidávat žádné externí součástky, protože je vše vyřešeno pomocí interních obvodů s pull up a pull down rezistory. Rozdíl nastává při požadavku na nahrání nového softwaru, kdy je potřeba takzvaný „download mód“, který je spuštěn, pokud je po startu pin GPIO0 nastaven na logickou 0 oproti výchozí 1. Připojení se tedy provádí externě pomocí 10 kΩ pull down rezistoru. Tuto funkcionalitu včetně restartu zařízení a nahrání programu dnes ale řeší většina USB-UART převodníků, které jsou používány právě při programování těchto modulů. V tomto případě tomu pravděpodobně nebude jinak, a proto je namísto integrace dalších obvodů pouze vyvedena šestice pinů potřebných k nahrání nového softwaru. Konkrétně se jde o piny EN, GPIO0, RX, TX, 3V3 a GND. Protože ale není jasné, jaké zařízení bude k těmto výstupům připojeno, jsou ještě k pinům RX a TX do série přidány ochranné rezistory o hodnotě 470 Ω, které omezí případný zkratový proud.

Další nezbytnou součástí jsou blokovací kondenzátory, které slouží k vyhlazení krátkých energeticky náročných špiček ESP32. Dokumentace doporučuje připojit co nejblíže modulu dvojici o hodnotách 10 µF a 100 nF, ale vzhledem k velkým výkyvům a kritické spodní hodnotě napájecího napětí popsané na konci kapitoly 5.3.4 bude pro jistotu použit 47 µF tantalový kondenzátor v kombinaci s 100 nF keramickým [63].

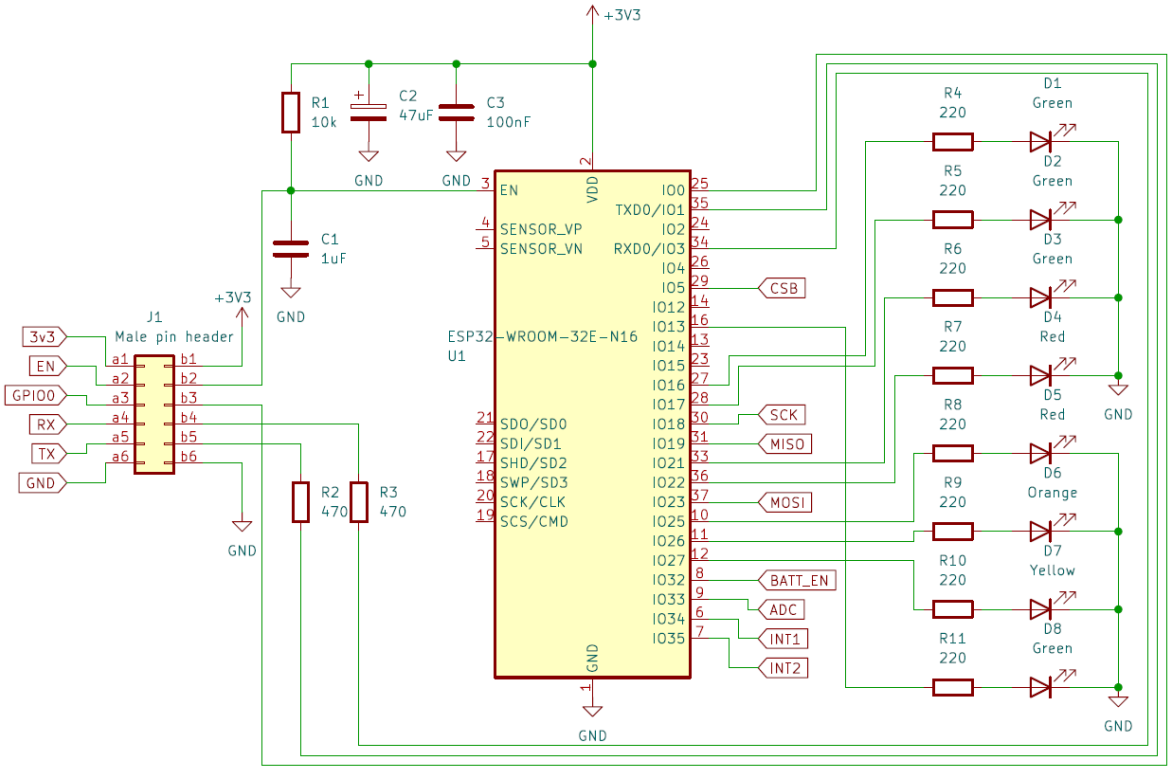
Poslední přidaný obvod bude sloužit k informování uživatele, a proto bude primárně složen z LED diod. Celkově se bude jednat o 4 zelené, 2 červené, 1 oranžovou a 1 žlutou. Zelené budou signalizovat zapnuté zařízení, připojení k přijímači a odesílání dat, červená bude značit chybový stav a čtveřice složená ze zelené, žluté, oranžové a červené bude zobrazovat stavu baterie. Každá z diod bude díky dostatečnému množství vývodů ESP32 připojena přes rezistor k samostatnému pinu s podporou PWM, aby v případě potřeby nebyl problém softwarově provést úpravy funkčnosti každé z nich. Obecně by měla být celková spotřeba co nejnižší, a proto zvolená červená KP-2012SRC-PRV, zelená KP-2012CGCK, oranžová 599-0130-007F a žlutá KP-2012SYCK budou každá doplněna o rezistor s hodnotou 220 Ω, který by měl podle Ohmova zákona zajistit proud diodou v jednotkách miliampér. Výkonnost by tak z pohledu svítivosti měla být v rozsahu 10 až 20 procent,

což představuje naddimenzovanou horní hranici, která bude pravděpodobně ještě softwarově snížena, aby byl celkový provoz co nejúspornější.

Vypočítané přesné hodnoty proudu v závislosti na očekávané horní a dolní hranici napájecího napětí popsaného v kapitole 5.3.4 jsou zobrazeny v následující tabulce:

Napětí na zdroji			2,45 V (minimum)		3,3 V (maximum)	
Barva LED diody	Napětí na diodě	Hodnota rezistoru	Napětí na rezistoru	Proud	Napětí na rezistoru	Proud
Červená	1,85 V	220 Ω	0,6 V	2,7 mA	1,45 V	6,6 mA
Zelená	2,1 V	220 Ω	0,35 V	1,6 mA	1,2 V	5,5 mA
Oranžová	2,1 V	220 Ω	0,35 V	1,6 mA	1,2 V	5,5 mA
Žlutá	2 V	220 Ω	0,5 V	2,3 mA	1,3 V	5,9 mA

Tabulka 6.1: Hodnoty protékajícího proudu LED diodami v závislosti na napájecím napětí



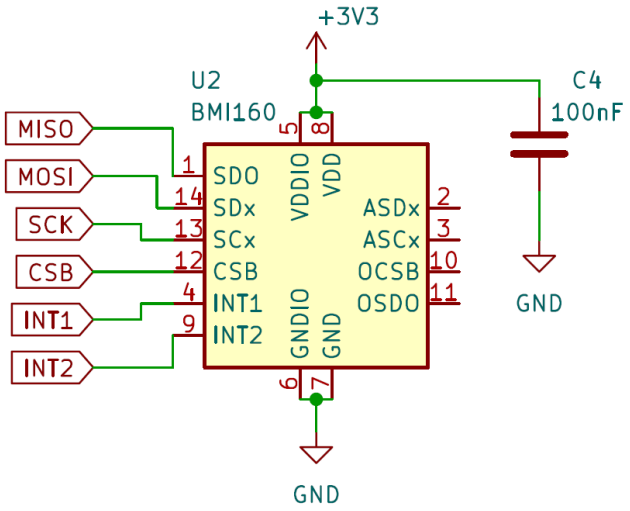
Obrázek 6.1: Zapojení ESP32

6.1.2 BMI160

Jak již bylo zmíněno, tak BMI160 podporuje komunikační rozhraní SPI i I^2C, a proto bylo nutné před tvorbou zapojení zvolit jednu z variant. Výběr konkrétně SPI byl poměrně jasnou volbou, protože se jedná o jedinou klíčovou komponentu pro komunikaci s ESP32, tak nebyl důvod nevybrat to rychlejší z komunikačních rozhraní.

Na základě tohoto rozhodnutí bylo vytvořeno zapojení, které přesně kopíruje uvedené doporučení v dokumentaci od výrobce [65]. Konkrétní použitá varianta je tedy s použitím pouze primárního komunikačního rozhraní v podobě SPI se 4 linkami a oběma piny pro přerušení, které přestože nebudou pravděpodobně využívány, tak do budoucna otevírají možnost pro případný vývoj, protože BMI160 disponuje řadou funkcionalit. Velmi zajímavá je například možnost generovat impulz na základě detekovaného pohybu přesahující danou mez. Takový impulz by mohl sloužit pro

buzení mikrokontroléru, jehož uspání by vedlo k mnohonásobnému snížení spotřeby, a tím i prodloužení výdrže zařízení na jedno nabití. Právě proto budou tyto dva piny pro přerušení připojeny k pinům ESP32 podporující možnost buzení.

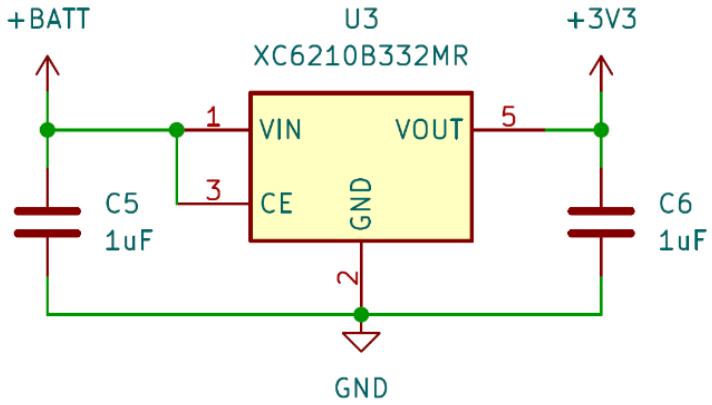


Obrázek 6.2: Zapojení IMU

6.1.3 Napájecí obvod

Tato část se skládá ze schématu zapojení napěťového regulátoru, kontroléru pro správu nabíjení a napěťového děliče napomáhajícímu detekci stavu baterie.

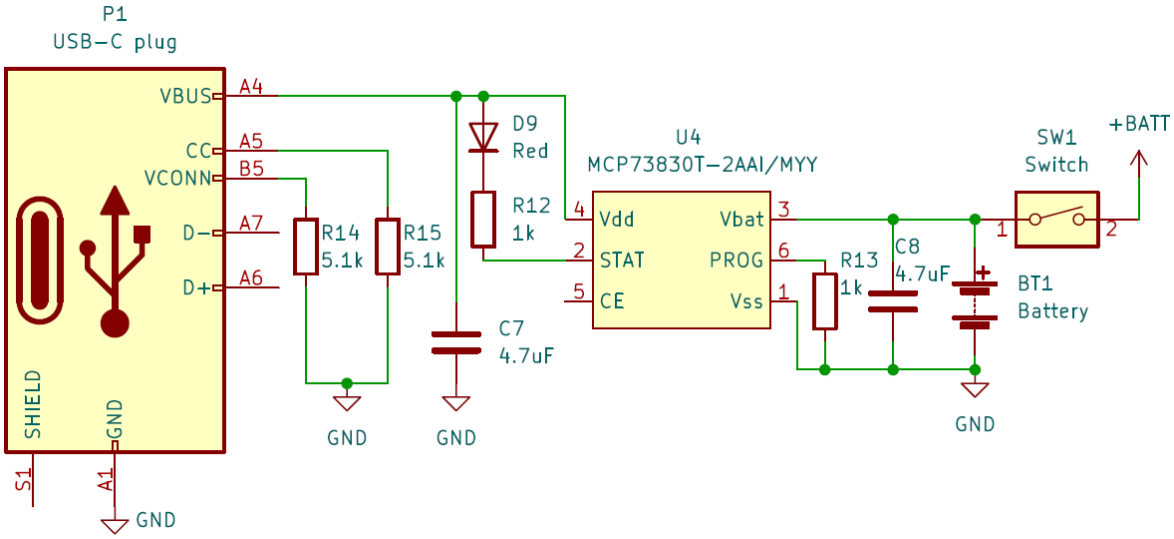
K napěťovému regulátoru by měly být podle dokumentace přidány dva keramické kondenzátory pro zajištění stability při případném kolísání zdroje nebo zátěže [64]. Jeden na vstupní svorky o hodnotě 1 µF a druhý na výstupní svorky s hodnotou minimálně 1 µF. Tato nejnižší doporučená kapacita bude dostačující, protože jedinou potenciálně kritickou komponentou z hlediska kolísání spotřeby by mohlo být ESP32, ale to již bylo doplněno o naddimenzované blokovací kondenzátory pro eliminaci tohoto jevu.



Obrázek 6.3: Zapojení LDO

Zapojení kontroléru pro správu nabíjení je taktéž totožné s typickým zapojením udávaným ve výrobcem vytvořené dokumentaci [53]. Rozdíl je pouze v hodnotě rezistoru zapojeném mezi pinem PROG a V_{SS}, který v závislosti na odporu od 1 kΩ do 10 kΩ určuje velikost nabíjecího proudu. Konkrétní velikost 1 kΩ pak byla zvolena na základě uvedeného vztahu $I_{NAB} = 1000 / R_{PROG}$ zvolena tak, aby bylo dosaženo maximálního možného nabíjecího proudu o hodnotě 1 A. Zároveň je však splněna podmínka o maximálním možném nabíjecím proudu popsaná v kapitole 5.3.2. Pro úplnost

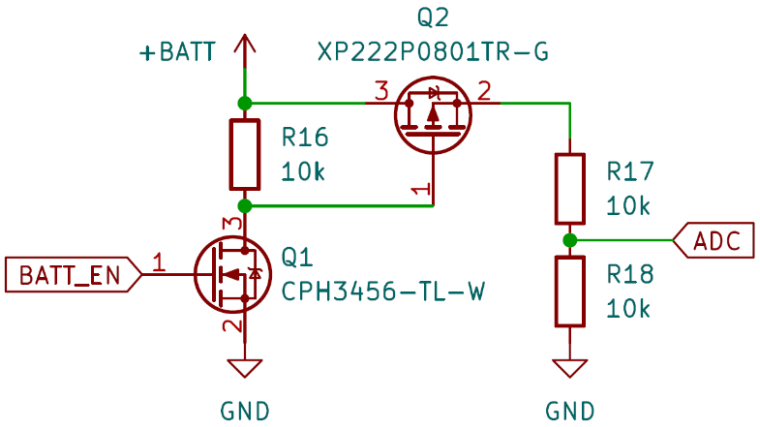
je nutné dodat, že jako signalizační LED dioda, která není v dokumentaci specifikována, je použita tato červená KP-2012SRC-PRV. Nad rámec udávaného zapojení je ještě přidána zásuvka se dvěma konfiguračními rezistory o hodnotě 5,1 kΩ pro zajištění nabíjení přes USB-C až 15 watty [14]. Poslední doplněnou součástkou je vypínací kolébkové tlačítko pro odpojení baterie od obvodu, vyjma toho nabíjecího.



Obrázek 6.4: Zapojení nabíjecího modulu

Posledním obvodem této části je napěťový dělič složený z dvojice 10 kΩ rezistorů. Ten je připojen k baterii a ADC převodníku ESP32, které je díky tomu schopné detekovat, jaký je aktuální stav baterie.

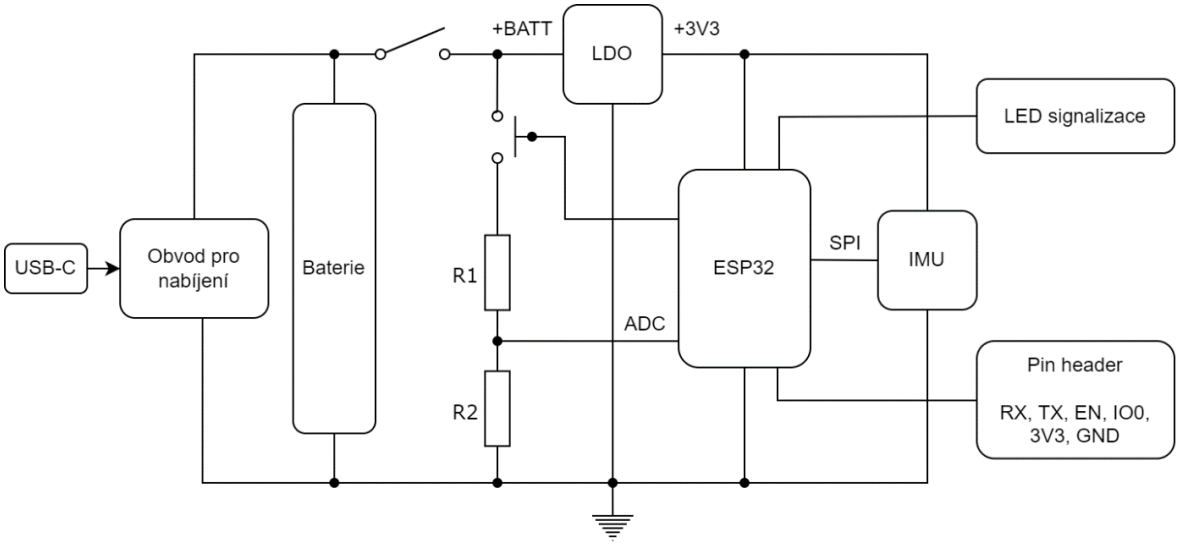
Pro případ budoucí modifikace cílící na minimalizaci spotřeby, která již byla naznačena v kapitole 6.1.2, je přidána ještě dvojice unipolárních tranzistorů. Jeden typu P, pro připojení a odpojení napěťového děliče, a druhý typu N, pomocí kterého bude ESP32 tranzistor typu P spínat. V nepřipojeném stavu tak bude spotřeba tohoto obvodu téměř nulová.



Obrázek 6.5: Zapojení děliče napětí

6.1.4 Blokové schéma zapojení

Výsledné schéma zapojení celého zařízení je spojením všech popsaných částí. Zde je pro představu zjednodušené grafické znázornění v podobě blokového schématu.



Obrázek 6.6: Blokové schéma celého zařízení

6.2 Deska plošných spojů

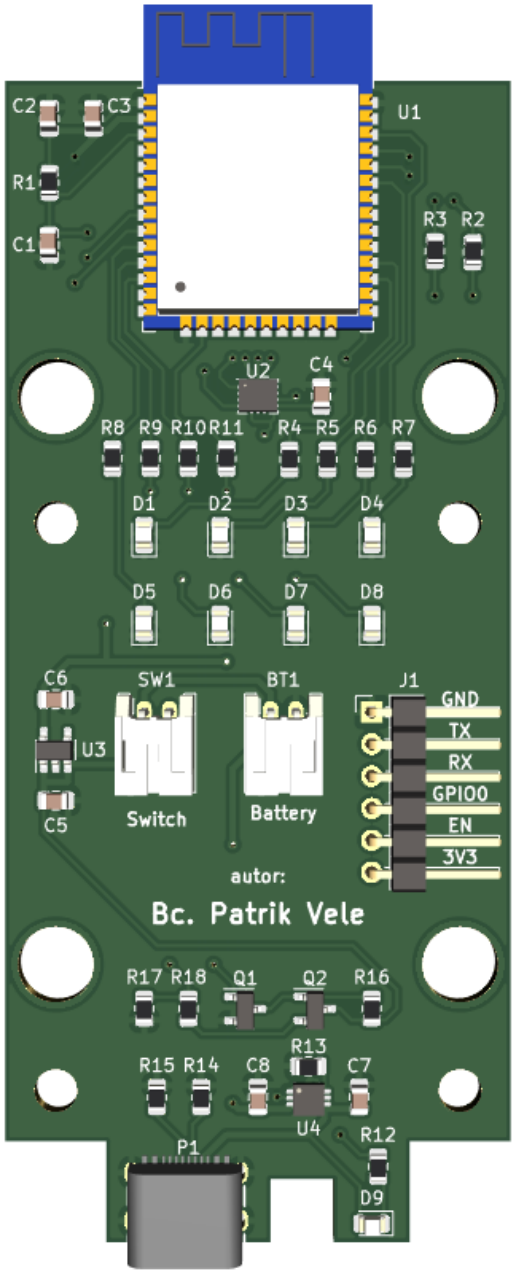
Dalším krokem na cestě k fyzické podobě zařízení je návrh desky plošných spojů vycházející ze schématu zapojení jednotlivých součástek. Jedná se o flexibilní proces, který je přímo odpovědný za některé hlavní vlastnosti celého zařízení, a proto je nezbytné před začátkem práce stanovit všechny požadavky.

Kromě obecného požadavku na minimalizaci desky je v tomto případě důležitý i tvar. Deska, a celé zařízení musí být co nejužší, aby kopírovalo prut a minimálně tak ovlivňovalo jeho vlastnosti. V rozích by měl být prostor pro spojení horního a spodního dílu krabičky. Po krajích by měly být 4 malé díry o průměru 3,4 mm pro uchycení desky standardními šrouby velikosti M3 a dále také 4 velké díry pro upevňovací mechanismus o průměru 6 mm. Nahoře by mělo být ESP32, aby jeho anténní část nebyla obklopena případnými rušivými obvody, naopak dole by měl být nabíjecí USB konektor, LED dioda signalizující nabíjení a otvor pro vedení vodičů skrz desku. Na desce pak musí být vhodně uspořádány informační LED diody a přístupné konektory s popisem.

Minimalizaci rozměrů zamezuje použitá baterie typu 18650, která v kombinaci s upevňujícími otvory a zapínacím tlačítkem stanovuje minimální vnitřní rozměry zařízení a tím i celé desky na 100 x 40 mm. Ostatní požadavky nepředstavují žádnou komplikaci a mohou být splněny různými způsoby, protože pro použité malé součástky, které jsou z velké části tvořeny rezistory, kondenzátory a diodami ve standardních SMD pouzdrech o velikosti 0805, jsou tyto rozměry desky o něco větší, než by bylo při důkladné minimalizaci nutné.

Konkrétní zvolené rozložení proto splňuje všechny výše stanovené požadavky, ale zároveň využívá celkový prostor pro přehledné rozmístění jednotlivých společných bloků, maximalizuje vzdálenosti součástek od konektorů, otvorů a informačních LED diod, dovoluje umístit potřebné popisy a umožnuje použití naddimenzovaných cest, kdy pro spojení s očekávaným proudem nad 100 mA je použita šířka 0,5 mm a pro ostatní spoje je použita šířka 0,25 mm. Konkrétně spoj 0,5 mm by

měl pro představu být podle výpočtu schopen vést proud o hodnotě až 2,4 A, přičemž nejvyšší proud v obvodu bude 1 A, pro který je doporučena cesta o šířce alespoň 0,15 mm [48].



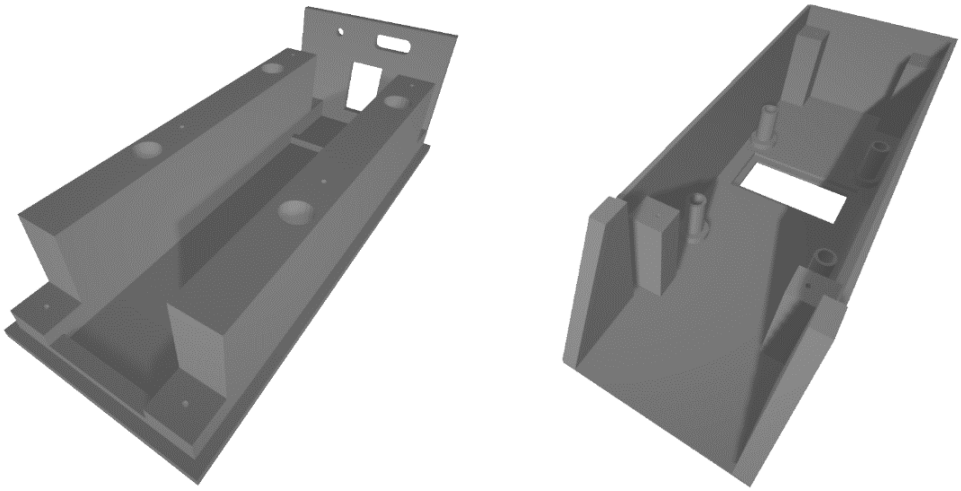
Obrázek 6.7: Deska plošných spojů

6.3 Krabička a upevňující mechanismus

Poslední úkol z hlediska realizace fyzické podoby zařízení je umístit všechny komponenty do vhodné krabičky umožňující připevnění na rybářský prut. Důležité je nezapomenout, že výsledný produkt bude v praxi vystavován prudkým pohybům a vzhledem k venkovnímu použití i různým podmínkám nepříznivého počasí, a proto jsou na jeho konstrukci kladeny specifické nároky, které vyžadují návrh krabičky na míru. K tomu je využíván program Tinkrcad.

Prvním, a naprosto zásadním požadavkem je minimalizace krabičky a přiblížení jejího těžiště těžišti celého prutu. To zajistí, že dojde k minimálnímu ovlivnění vlastností prutu, a zároveň budou redukovány nežádoucí síly vznikající při pohybu, což nahrává kvalitnímu upevnění bez nutnosti použití nadměrné síly, která by mohla vést k potenciálnímu poškození prutu. Druhým požadavkem je voděodolnost na takové úrovni, aby se při dešti nedostala voda do krabičky, konektoru USB-C a otvoru pro signalizační LED diodu. Jinými slovy, aby bylo možné zařízení používat při lovu v libovolném počasí. Další důležitou podmínkou je vytvoření upevňujícího mechanismu, který celé zařízení spolehlivě zajistí ve stejné poloze i při prudkých pohybech. Zároveň bude dostatečně malý, a to především v oblasti pod prutem, aby nijak neohrožoval vlasec vedoucí z navijáku do prvního očka, kde vzniká riziko kontaktu především při rychlém odvíjení během náhozu. Poslední podmínkou je vytvoření otvoru pro zapínací tlačítko, signalizační LED diodu, USB-C konektor a také otvoru s rámem pro instalaci průhledného plexiskla, přes které budou viditelné všechny signalizační LED diody.

Samotná realizace je z důvodu jednoduššího tisku rozdělená na krabičku a držák, přičemž obě části se skládají ze spodního a horního dílu. Celkový tvar, ale především délka a šířka krabičky vychází z rozměrů desky plošných spojů. Naopak výška je dána baterií. Co se konkrétního rozmístění komponent týče, tak baterie bude jakožto nejtěžší díl umístěna rovnoběžně s prutem ve středu spodní části krabičky, aby minimálně ovlivňovala vlastnosti prutu. Nad ní, a zároveň těsně pod horním dílem, bude deska plošných spojů, což zajistí dobrou viditelnost signalizačních LED diod přes plexisklo přidělané v otvoru na horní straně krabičky. Přední strana je věnována zapínacímu tlačítku, USB-C konektoru a signalizační LED diodě, přičemž všechny tyto kritické části jsou chráněny před deštěm díky krytu v podobě prodloužení horního dílu. Výsledná krabička je bez přesahu horního dílu dlouhá 100 mm, široká 45 mm a vysoká 33 mm.



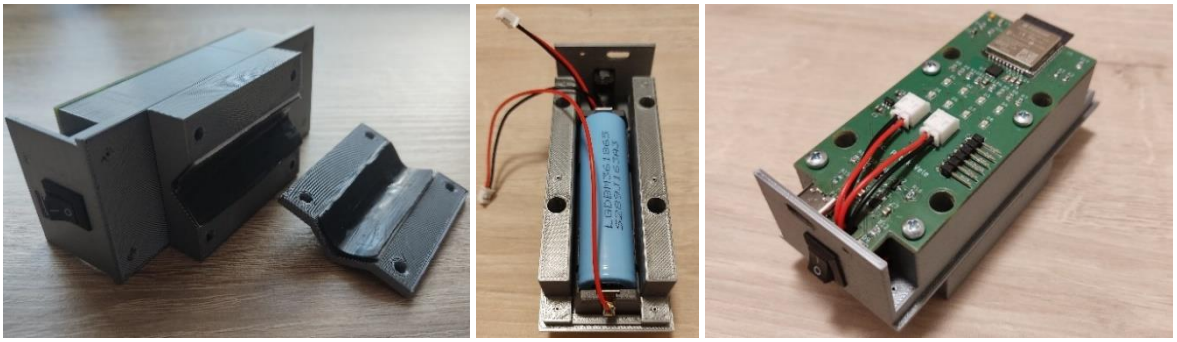
Obrázek 6.8: Navržená krabička

Upevňující mechanismus je tvořen dvěma díly s vnitřním výřezem tvaru písmene „V“. Tento tvar styčných ploch zajistí, že na prutu s libovolným průměrem budou vždy 4 styčné plochy, které zajistí maximální stabilitu a pevnost. Tomu velmi napomáhá i délka držáku 55 mm. S upevněním souvisí i čtveřice otvorů vedoucí skrze celé zařízení, která slouží k instalaci 4 šroubů typu M3, pomocí kterých je celé zařízení přichyceno k prutu.

6.4 Výsledná podoba zařízení

Současná nepříznivá situace ohledně nedostatku a měnící se dostupnosti čipů zapříčinila, že výsledné zařízení muselo být částečně poupraveno a místo zvolené inerciální měřící jednotky BMI160 je pro účely této práce použita ICM20689, a to především díky své aktuální dostupnosti.

Zkonstruování zařízení začíná výrobou krabičky, což představuje nejen vytisknutí všech čtyřech navržených částí na 3D tiskárně, ale následné slepení horního dílu držáku se spodním dílem krabičky, přilepení ochranného plexiskla a opatření styčných ploch držáku, které přijdou do kontaktu s prutem, nějakým protiskluzovým měkkým materiálem. V tomto případě je jako protiskluzový materiál použita část smršťovací bužírky, která je přilepená oboustrannou lepící páskou. Do takto připravené krabičky je následně přilepené pouzdro s baterií, nainstalováno zapínací tlačítko a přišroubována osazená základní deska.



Obrázek 6.9: Konstrukce detekčního zařízení

Výsledná podoba zařízení je dokončena přidáním horního krytu krabičky a připevněním celého produktu k rybářskému prutu pomocí čtveřice matek a dostatečně dlouhých šroubů typu M3.

Obrázek 6.10: Připevněné detekční zařízení na prut

7 Vytvoření datové sady

V kapitole je popsán celý postup od čtení dat ze senzoru, až po vytvoření použitelné datové sady pro školení modelů strojového učení. První část, která je implementována v Arduino frameworku, se věnuje sběru dat a jejich přepravě do počítače. Druhá část, která se odehrává v prostředí Google Colaboratory, pak navazuje na výstupní data z části první. Ta jsou postupně zpracována a převedena do podoby standardní datové sady.

7.1 Sběr dat

Sběr dat je úloha, která by se dala charakterizovat jako posloupnost dílčích
Sběr dat je posloupnost úloh, kde každá má několik řešení, omezení, výhod a nevýhod. Cílem je pak nalezení nejvýhodnějšího řešení, které bude mít všechna omezení minimální vliv. V tomto případě se jedná o trojici úloh, které jsou řešeny ve zbytku této kapitoly:

1) Jak maximalizovat množství zaznamenané informace s omezenými zdroji zařízení?
2) Jak data označovat?
3) Jak dostat data ze zařízení do počítače?

7.1.1 Maximalizace informace s omezenými zdroji

Vzhledem k použití a návrhu zařízení není možné, aby z něj byla data získávána v reálném čase drátově. Zbývá tedy možnost bezdrátová. Ta ale nesplňuje podmínku jistého záznamu souvislého bloku dat, protože může dojít k výpadku a část informace nebude přenesena. Taková data jsou pak pro účely detekce pohybu nepoužitelná. Řešením by byla spolehlivá komunikace s potvrzením o doručení, avšak ta zase nesplňuje dostatečnou rychlost přenosu. Je tedy jasné, že jedinou možností je data ukládat přímo v zařízení a odesílat je až po dokončení záznamu.
Poměrně elegantním řešení záznamu je využití vestavěné flash paměti, kterou ESP32 disponuje. Problém je opět v rychlosti, protože otevření souboru a sekvenční zápis může v závislosti na počtu zapsaných dat trvat až stovky milisekund. Z toho plyne, že zápis po jednom bytu v případě této aplikace nedává smysl. ESP32 ale disponuje poměrně velkou RAM pamětí, ze které je možné uživatelsky využít více jak 120 kB. Přidáním této možnosti je konečně nalezeno použitelné řešení.

Z hlediska funkčnosti se jedná o princip takzvaného swapování. V tomto případě se jedná o proces, kdy jsou naměřené hodnoty postupně ukládány do předem připravené oblasti v paměti RAM označované jako buffer. Vyčerpáním kapacity tohoto bufferu následuje druhá fáze, kterou je uvolnění bufferu. To je zprostředkováno překopírováním veškerého zaznamenaného obsahu do textového souboru na integrovanou paměť flash. Tímto způsobem je proces analogicky opakován, dokud nedojde i k naplnění této paměti. Tím se přechází do třetí fáze, kde jsou sjednoceny všechny takto vytvořené soubory v jeden, ve kterém jsou zaznamenané souvislé bloky o velikosti RAM bufferu odděleny znakem nového řádku.

V tuto chvíli teprve vyvstává na povrch optimalizační část, protože RAM buffer o velikosti 100 kB představuje v této aplikaci značné omezení. Pokud by totiž frekvence čtení dat byla 500 Hz ve všech šesti osách, každá z hodnot by byla uložena jako datový typ double o velikosti 8 B a každý pohyb by trval 5 sekund, tak by bylo potřeba 120 kB. Dostupná paměť by tedy nestačila ani na záznam jednoho pohybu, a proto je nezbytné zaměřit se na všechny zmíněné parametry a pokusit se je zoptimalizovat.

Prvním krokem je snížení vzorkovací frekvence. Ta byla po provedení několika testovacích pohybů stanovena na hodnotu 50 Hz, kdy ještě stále dochází k dostatečnému zaznamenávání všech změn. Počet os zůstává stejný a datový typ se změní na float o velikosti 4 B. Touto trojicí úprav se možnost záznamu dostává na hranici 83 sekund, tedy až 27 pohybů za předpokladu, že každý pohyb má délku 3 vteřiny. Takovýto výsledek se již dá považovat za dostatečný a použitelný.

Další optimalizaci je nutné aplikovat na proces ukládání dat na flash paměť, aby byl prostor s dostupnou velikostí asi 1200 kB co nejlépe využit. Všechny informace budou uloženy do textovém souboru, kde každý znak zabírá 1 B, a proto je nutné upravit data tak, aby s co nejmenším počtem znaků disponovala maximální možnou informační hodnotou. V první řadě je potřeba nastavit rozsah akcelerometru na ±8G a gyroskopu na ±2000 °/s. Vzhledem k tomu, že se jedná o pohyby prutu, tak by nebylo od věci zvolit maximální možné hodnoty, ale i menší rozsah akcelerometru se při testování jevil jako dostatečně naddimenzovaný. Navíc tato volba nahrává další úpravě, kterou je převedení na jednotky cm/s² a tím i rozsah na ±7850 cm/s². Tato podoba čísel je z pohledu nosnosti informace jednotlivých znaků mnohem výhodnější, protože prvních 5 znaků každého čísla obsahuje při pohybu s největší pravděpodobností znaménko a 4 platné číslice, zatímco v přechozím případě byl vždy jeden znak blokován desetinnou čárkou a velice často také první znak blokovaný nulou u desetinného čísla. Další výhodou stanovení přesné velikosti je možnost ukládání znaků do souboru za sebe, bez nutnosti použití oddělovačů jako jsou mezery či jiné speciální znaky, čímž je ušetřeno znovu velké množství místa. Právě tato optimalizace v podobě stanovení přesného počtu znaků pro všechna čísla na hodnotu 5 je aplikována. Nakonec je ještě důležité zmínit, že každý zápis do souboru z jednoho čtení senzoru se skládá ze 31 znaků, kde prvních 15 zabírají data akcelerometru, následujících 15 data gyroskopu a závěrečný jeden znak je tag označující typ pohybu.

Poslední pomůckou k lepšímu využití omezených zdrojů je zavedení trojice pracovních módů, kde první představuje standardní ukládání všech pohybů, druhý ukládá pouze data označená jako pohyb a třetí naopak ukládá pouze data bez pohybu. Při běžném lovu je většina času bez pohybu, takže by výsledná data byla extrémně nevybalancovaná, což je v případě omezené paměti velice nevýhodné, a proto je tento přístup nezbytný pro maximalizaci efektivity.

7.1.2 Označování dat

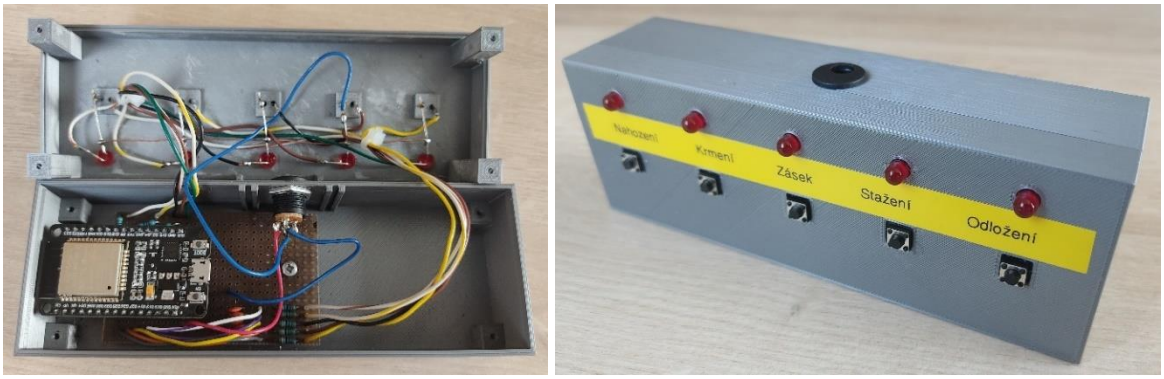
Jedná se o vytvoření datové sady pro učení, takže jak již bylo zmíněno na konci předchozí kapitoly, každá zaznamenaná data je nutné popsat příslušnou hodnotou označovanou jako tag, který vyjadřuje, o jaký druh pohybu se jedná. V kapitole 2.2 bylo vybráno celkem 5 klíčových pohybů, a proto byl rozsah tagu intuitivně nastaven na hodnoty 0 až 5, kde:

- 0 - žádný pohyb
- 1 - nahození
- 2 - krmení
- 3 - zásek
- 4 - stažení
- 5 - odložení

Je jasné, že toto označování je nutné vytvářet ručně a v reálném čase. Navíc jak již bylo zmíněno na začátku předchozí kapitoly, tak připojování kabelů nebo dalších modulů k detekčnímu zařízení nepřipadá v úvahu. Nejlepší volbou je v tomto případě vytvořit zařízení druhé, které bude s tím detekčním komunikovat bezdrátově a poskytovat mu tak v reálném čase informaci o aktuálním prováděném pohybu.

Základem tohoto označovacího zařízení je opět ESP32. Tentokrát však v podobě vývojové desky, která je doplněna pěticí LED diod a pěticí tlačítek. Celá tato konfigurace je umístěna v navržené a vytisknuté krabičce na 3D tiskárně. Napájení je zajištěno přes standardní DC konektor,

ke kterému je zhotoven kabel s USB koncovkou, aby bylo možné zařízení napájet z power banky, což je prakticky asi jediný způsob, jak se zařízením pohodlně pracovat v přírodě u vody. Posledním důležitým prvkem je štítek s popisky, který přiřazuje každé LED diodě a každému tlačítku konkrétní pohyb.



Obrázek 7.1: Označovací zařízení

Co se funkčnosti týče, tak pro bezdrátovou komunikaci s detekčním zařízením je využíváno ESP-NOW, což je jednoduchý komunikační protokol vytvořený firmou Espressif, který představuje velice rychlou a snadnou komunikaci mezi zařízeními s pakety o velikosti až 250 B [42]. Pro tuto aplikaci se tak jedná o ideální řešení.

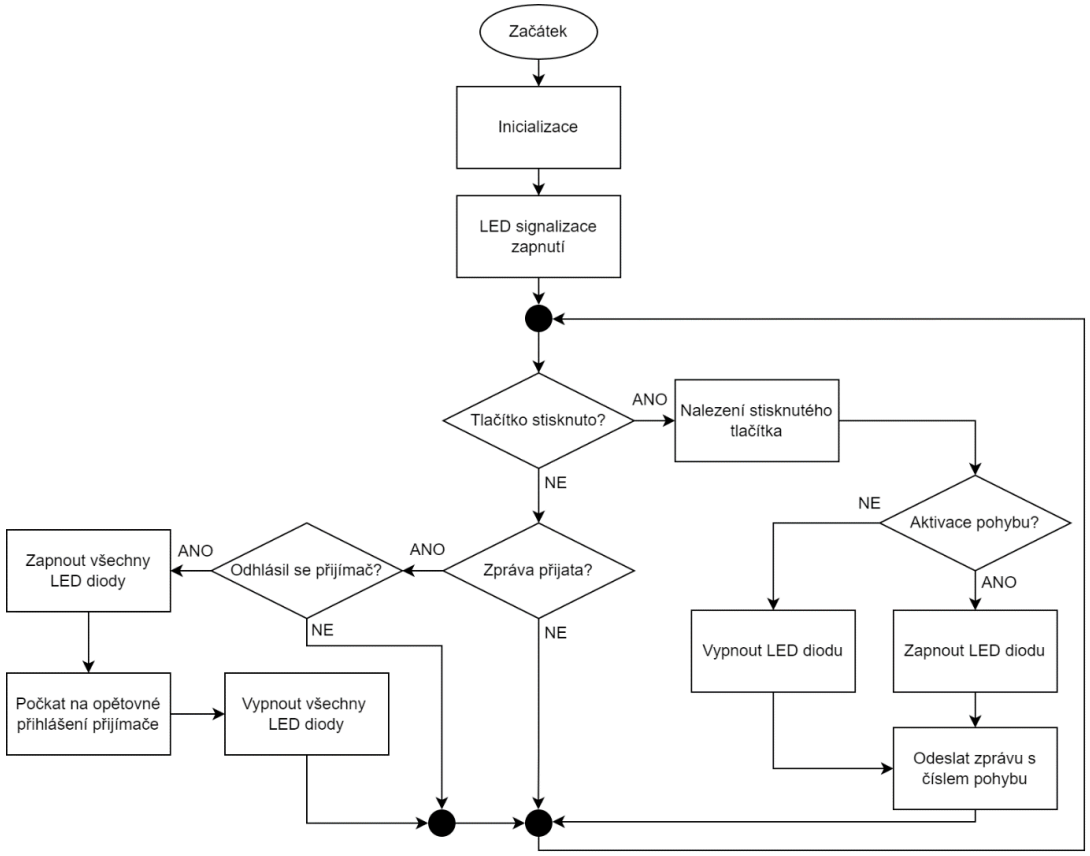
Samotná implementace komunikace přes ESP-NOW je podmíněna pouze definicí struktury zpráv na obou komunikačních stranách a specifikací přijímače na straně vysílače. Identifikátor cílového zařízení je v tomto případě MAC adresa, kterou je možné do terminálu vypsat pomocí následujícího programu.

```
#include <Arduino.h>
#include <WiFi.h>

void setup()
{
    Serial.begin(115200);
    while (!Serial)
        delay(10);
    Serial.println();
    Serial.println("Device MAC address: " + WiFi.macAddress());
}

void loop() {}
```

Obrázek 7.2: Ukázka kódu pro výpis MAC adresy ESP32

Proces označování pak funguje tak, že v momentě stisknutí tlačítka odešle označovací zařízení zprávu detekčnímu zařízení s informací, o který pohyb se jedná. Od chvíle, kdy je tato zpráva detekčním zařízením přijata, jsou všechna zaznamenaná data označována příslušným tagem. Tento proces končí ve chvíli, kdy je přijata identická zpráva, která značí ukončení prováděného pohybu. Jako zpětná vazba správného stisku na označovacím zařízení je pro uživatele vždy aktivní pohyb označen rozsvícenou příslušnou LED diodou. Poslední funkcí označovacího zařízení je rozsvícení

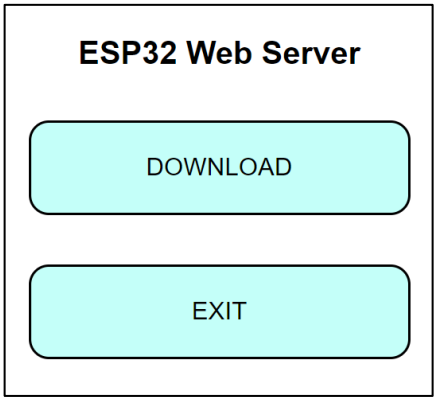
všech LED diod, což signalizuje, že byla z detekčního zařízení přijata zpráva o ukončení záznamu z důvodu plné paměti.



Obrázek 7.3: Blokové schéma programu označovacího zařízení

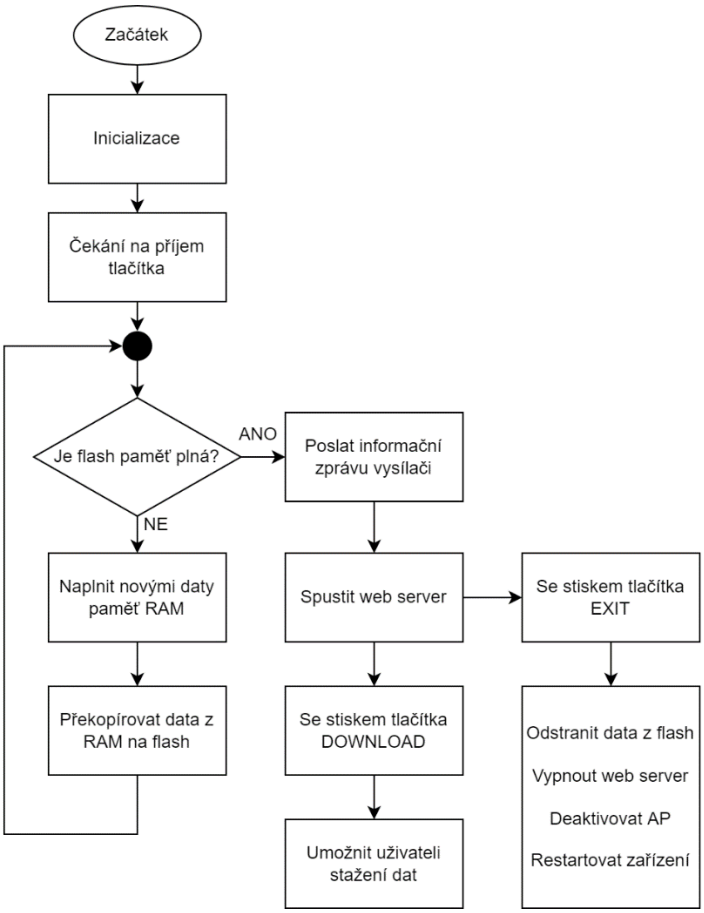
7.1.3 Přenos dat do počítače

Zaznamenaná, správně označená a uložená data v textovém souboru na flash paměti detekčního zařízení je potřeba nějakým způsobem dostat do počítače, kde může proběhnout další zpracování s následným vytvořením datové sady. K této situaci dochází především u vody, kde jsou možnosti omezené, a proto je zvolen přístup v podobě spuštění webového serveru přímo na detekčním zařízení. Tím dochází k eliminaci mnoha problémů, protože jedinou potřebnou pomůckou je mobilní telefon nebo jakékoliv jiné zařízení vybavené WiFi rozhraním, se kterým se stačí připojit k přístupovému bodu s názvem „ESP32“ a heslem „12345678“ a následně přejít v prohlížeči na doménu s adresou „7.7.7.1“. Otevře se jednoduchá webová stránka, která byla vytvořena v jazyce HTML a doplněna grafikou za pomoci kaskádových stylů. Na ní se nachází dvojice tlačítek. První je tlačítko DOWNLOAD, které aktivuje stažení výstupního textového souboru s daty. Druhé EXIT, které odstraní výstupní soubor z detekčního zařízení, ukončí webový server, vypne přístupový bod a restartuje zařízení. Po restartu je pak zařízení připraveno na nový proces sběru dat.



Obrázek 7.4: Webová stránka provozovaná webovým serverem detekčního zařízení

7.1.4 Blokové schéma programu



Obrázek 7.5: Blokové schéma programu pro sběr dat detekčním zařízením

Úvodem celého programu je inicializace, která zajišťuje změření a vizuální zobrazení stavu baterie, přípravu a nastavení inerciální měřící jednotky, alokaci RAM bufferu o kapacitě 100 kB, inicializaci bezdrátové komunikace typu ESP-NOW a přečtení nebo vytvoření info.txt souboru, který obsahuje informaci o tom, jestli se na flash paměti nachází některé datové soubory z předchozího měření a případně kolik jich je. Pak přichází ohlášení se označovacímu zařízení odesláním zprávy přes ESP-NOW a následuje vyčkávání na příjem zprávy po prvním stisku tlačítka, čímž začíná celý proces záznamu. Během něj je s frekvencí 50 Hz provedeno jedno čtení dat z inerciální měřící jednotky a obdržené hodnoty uloženy do připraveného RAM bufferu. Naplněný buffer je pak následně překopírován do souboru na flash paměť, kde je pro tento účel vyhrazeno místo 1 MB. Postupně je tak vytvořeno celkem 9 souborů a ty na závěr sjednoceny do jednoho output.txt souboru. Samotné sjednocení je velice náročná úloha vyžadující na zpracování asi 15 minut. Před tím je však uživatel informován k ukončení záznamu pomocí zprávy přes ESP-NOW. Po vygenerování výstupního souboru pak dochází k vytvoření přístupového bodu a spuštění webového serveru. Uživatel má tak možnost si do libovolného zařízení disponující WiFi rozhraním stáhnout naměřená data přes jednoduché webové rozhraní kliknutím na tlačítko DOWNLOAD. Stejně tak má možnost tlačítkem EXIT data následně odstranit a připravit tak zařízení na další měření.

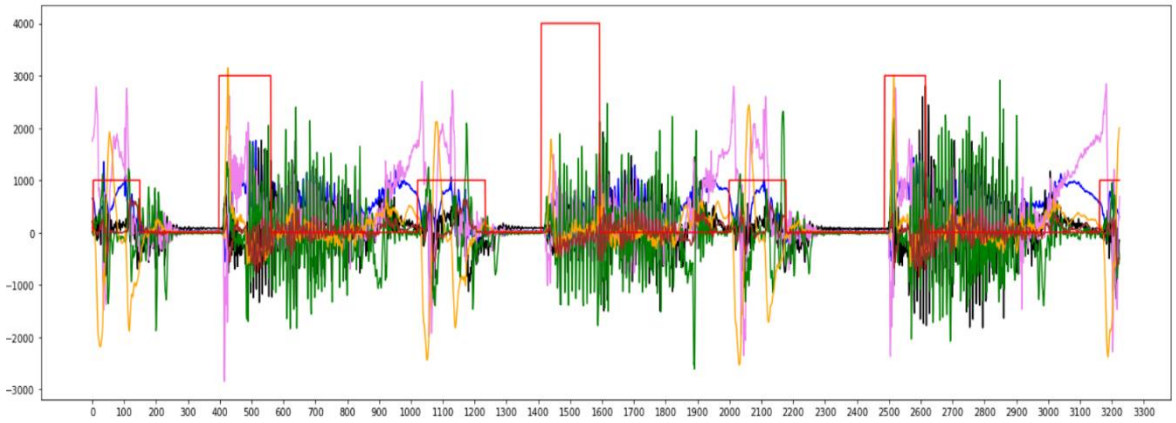
7.1.5 Zavrhnutí pohybu odložení

Vzhledem ke zmíněné současné situaci s nedostatkem komponent se podařilo vyrobit pouze jedno detekční zařízení, což vedlo k zamyšlení se nad informační hodnotou pohybu „odložení“ v případě jednoho prutu. Po následné konzultaci s ostatními závodníky bylo rozhodnuto tento pohyb do současného řešení nezahrnovat, protože současná konfigurace by poskytovala nulovou informační hodnotu.

7.2 Zpracování dat

Sběrem dat v reálném prostředí vznikly nezpracované záznamy pohybů uložené v textových souborech. V tomto případě se jednalo o dvojici měření, kde první bylo provedeno v módu 1 a druhé v módu 2. Výsledkem je dvojice datových souborů output1.txt a output2.txt, které musí být vhodným způsobem zpracovány tak, aby výsledkem byla použitelná datová sada. Celý tento proces se bude odehrávat v prostředí Google Colaboratory, a proto je před začátkem nezbytné uložit oba datové soubory na Google Disk, který bude sloužit jako úložiště.

Samotné zpracování dat začíná načtením obsahu souboru a rozdělením zaznamenaných částí, které jsou odděleny znakem nového řádku. Každá část je pak rozdělena na jednotlivé vzorky o 31 znacích, a každý vzorek je dále rozdělen na konkrétní hodnoty. Poslední úpravou je převod na celá čísla, což je pouze aplikace již původního úmyslu. Následuje kontrolní zobrazení.



Obrázek 7.6: Nezpracovaná data z detekčního zařízení

Z grafu se vizuálně detekují chybějící označení a případné další problémy. Poté jsou ručně nastaveny oblasti, kam je dodatečně přidán tag. Všechny pohyby jsou pak následně převedeny do podoby reprezentace, která je dána trojicí {začátek, konec, tag}, a podle výběru konkrétní části jednotlivě zobrazeny uživateli. Ten má pak na základě přiblíženého zobrazení každého pohybu možnost ručně stanovit přesný začátek pohybu pomocí hodnoty posunu danou počtem vzorků. Jeho druhou možností je celý pohyb označit jako neplatný pomocí čísla -1, což je zpravidla případ, kdy není pohyb zaznamenán celý a je tedy nutné ho odebrat.

Na základě přiblíženého zobrazení je také nezbytné stanovit i počet vzorků, který bude všechny pohyby reprezentovat. Ten byl pro tuto aplikaci zvolen na 100, což při vzorkování 50 Hz odpovídá 2 sekundám.

Zde pak končí podstatná manuální práce a začíná aplikace nastavených parametrů. To znamená, že platné pohyby jsou nejprve na základě posunů a délky vycentrovány, přebývající části společně s neplatnými pohyby odstraněny, zbývající části bez pohybu jsou navzorkovány velikostí pohybů a zbývající části o nedostatečné délce jsou opět odstraněny. Výsledkem je posloupnost jednotlivých pohybů a klidových částí o délce 100 vzorků.

Tato verze je uložena do souboru dataset.csv na Google Disk, kde první řádek obsahuje velikost rámce neboli počet vzorků každého pohybu. V tomto případě je to hodnota 100. Další řádek je hlavička popisující sloupce jednotlivých os a také sloupec tagu. Za ní pak pokračují data, kde je na každém řádku uložen jeden vzorek.

```
Frame size:  100

        AX   AY    AZ  GX  GY  GZ  TAG
0      580 -494  2238 -43 -21   0    1
1      452 -460  2051 -37 -28  -3    1
2      452  -60  2069 -28 -29  -4    1
3      466  116  1614   6 -29  -2    1
4      441 -212  1249  46 -23   0    1
...    ...  ...   ...  ..  ..  ..  ...
34195  415  187    89   0 -10 -11    4
34196  317  289   242  11 -11  -6    4
34197  382  -14    -7  39 -11  -6    4
34198  453 -232   197  64  -5  -8    4
34199  693 -301   132  96  -9 -13    4

[34200 rows x 7 columns]
```

Obrázek 7.7: Formát dat v souboru dataset.csv

Nejedná se ještě o výslednou datovou sadu, ale mezilehlý krok, který dává možnost jednoduchého přidávání dalších dat v případě budoucích měření. Právě toho je využito již v případě souboru output2.txt, kdy jsou data po zpracování pouze přidána na konec souboru dataset.csv za ta již existující.

S dostatečný počtem takto připravených dat je již možné vytvořit standardní datovou sadu použitelnou pro školení nějakého modelu strojového učení. K tomu je vytvořen další program, který podle očekávání začne s tím, že načte data ze souboru dataset.csv. Pro ujištění o dostatečné velikosti datové sady pak zobrazí uživateli histogram četnosti jednotlivých pohybů.

```
Zadny pohyb :  145
Nahozeni :  57
Krmeni :  46
Zasek :  42
Stazeni :  52
Odlozeni :  0
```

Obrázek 7.8: Histogram jednotlivých pohybů v datové sadě

Proces pokračuje rozdělením na data o velikosti N x 6 a tag o velikosti N. Dále jsou sloučeny jednotlivé pohyby a tím se velikost dat mění na M x 100 x 6. S tím je adekvátně zmenšen počet tagů tak, aby každému pohybu odpovídal jeden tag. Velikost vektoru tagu se tím redukuje z N na M. Poslední úpravou je vybalancování jednotlivých vzorků, což je rozložení, které algoritmy strojového učení u datové sady očekávají a bez splnění této podmínky by pravděpodobně nedocházelo k jejich správné funkci. Vybalancování je konkrétně provedeno tak, že dojde k výběru z každé třídy tolik vzorků, jako je počet nejméně vyskytujícího se pohybu v datové sadě. Tato data pak byla náhodně rozdělena na trénovací a validační v poměru 80:20. Výsledkem je 168 trénovacích příkladů v souboru train_X s příslušnými výsledky v souboru train_y a 42 validačních příkladů v souboru test_X s příslušnými výsledky v souboru test_y. Všechny tyto soubory jsou uloženy na Google Disk v jednom archivu s název train_test_dataset.npz a představují výslednou datovou sadu ve standardním formátu.

8 Implementace konvoluční neuronové sítě

V kapitole 4 byly vybrány dvě vhodné techniky pro řešení úlohy rozpoznávání pohybů feederového prutu. Tou první z nich byla právě konvoluční neuronová síť, která je hlavní náplní této kapitoly. Konkrétně zde bude popsán její vytvoření, natrénování co neúspěšnějšího modelu a nakonec i převedení do podoby umožňující integraci modelu do detekčního zařízení.

Zásadní je v tomto případě výběr vývojové platformy, která vše toto umožní. Jednou z mála, a zároveň asi tou nejpoužívanější v tomto směru, je platforma TensorFlow se svou knihovnou TensorFlow Lite. Jedná se o stěžejní bod zajišťující uskutečnitelnost celé práce, a proto byla tato volba provedena již na začátku, kde hrála poměrně důležitou roli při výběru mikrokontroléru v kapitole 5.1. Je však důležité neopomenout, že tato knihovna má jistá omezení. Tím prvním je podpora jen několika málo zařízení, mezi kterými je samozřejmě zastoupeno i ESP32 [16]. Druhou limitaci představuje omezený seznam podporovaných operací, které mohou modely strojového učení využívat [60]. V seznamu se například nenachází operace 1D konvoluce, a proto bude nutné použít při práci 2D konvoluci.

Nyní jsou objasněny všechny potřebné informace a je tedy možné přejít k implementaci. Ta začne vytvořením modelu, pokračovat bude jeho tréninkem a převodem nejprve na TensorFlow Lite model a následně na pole bajtů programovacího jazyka C. To vše bude probíhat v prostředí Google Colaboratory.

S využitím knihovny Keras a inspirací architekturou LeNet byla vytvořena prvotní verze konvoluční neuronové sítě.

```
model = Sequential()
model.add(layers.Conv2D(filters=5, kernel_size=(3,1), activation='relu', input_shape=(100,6,1)))
model.add(layers.MaxPooling2D(2, 2))
model.add(layers.Conv2D(filters=10, kernel_size=(9,1), activation='relu'))
model.add(layers.MaxPooling2D(2, 2))
model.add(layers.Flatten())
model.add(layers.Dense(units=200, activation='relu'))
model.add(layers.Dense(units=20, activation='relu'))
model.add(layers.Dense(units=5, activation = 'softmax'))
```

Obrázek 8.1: Počáteční konfigurace konvoluční neuronové sítě

Je vidět, že je použita zmíněná 2D konvoluce. Společně s tím muselo dojít i k rozšíření vstupu o jeden další rozměr, což však nemá vliv na data, protože se jedná o převod rozměru 100 x 6 na rozměr 100 x 6 x 1. Další podstatnou úpravou bylo nastavení velikosti poslední plně propojené vrstvy, která udává počet očekávaných vstupů, což je v tomto případě 5. Ostatní parametry byly nastaveny pouze intuitivně.

Následoval trénink a tím i otestování úspěšnosti modelu. Nejprve byla však načtena datová sada, jednotlivá pole s daty pak rozšířena o jeden další rozměr a konkrétní hodnoty nakonec ještě normalizovány, respektive převedeny do rozsahu od -1 do 1. Následovala kompilace vytvořené modelu a konečně i spuštění tréninku, které vyžadovalo nastavení počtu epoch a velikost batch. Velikost batch udává, kolik prvků datové sady projde modelem, než dojde k úpravě jeho parametrů [3]. Volbou 42 je dáno, že během jednoho školení na trénovací sadě čítající 168 příkladů dojde ke 4 aktualizacím parametrů modelu. Parametr počet epoch zase udává, kolikrát bude celá datová sada použita k procesu učení [3]. Jako výchozí je zvolena hodnota 100. Po několikanásobném provedení tréninku se ve většině případů pohybovala úspěšnost modelu v rozmezí 90 – 100 %. Důvodem takto vysoké počáteční úspěšnosti je fakt, že jednotlivé pohyby jsou navzájem dosti odlišné a v podání zkušeného rybáře je jejich provedení téměř identické.

Následovalo testování jednotlivých kombinací parametrů za účelem najít ty nejlepší a maximalizovat tak úspěšnost modelu. Prvním byla velikost jádra konvolučních vrstev. Vzhledem ke

skutečnosti, že trénovací data byla vzorkována s minimální možnou frekvencí, předpokládá se, že velikost prvního filtru by měla být co nejmenší, aby zachytil ty nejmenší informace. Naopak druhý filtr by měl být větší, aby zachytil komplexnější informace. Větší filtr je sice výpočetně náročnější, ale díky předřazené pooling vrstvě je možné si to dovolit, protože vstup do druhé konvoluční vrstvy je oproti té první poloviční. Vzhledem k funkčnosti těchto filtrů se používají liché rozměry, a proto byly vybrány následující kombinace:

- 3 x 1 + 9 x 1
- 5 x 1 + 9 x 1
- 5 x 1 + 15 x 1

Nejlepšího výsledku dosahovala předpokládaná nejmenší kombinace 3 x 1 + 9 x 1. Druhé testování porovnávalo počty filtrů konvoluce, konkrétně možnosti:

- 3 + 5
- 5 + 10
- 10 + 20

Nejlépe vyšla opět intuitivně nastavená hodnota 5 + 10. Třetí test porovnával velikost batch o hodnotách 8, 21, 42, 84 a 168, kde nejúspěšnější bylo číslo 21. Tento výsledek byl očekávaný, protože s malou datovou sadou by měla být častější aktualizace modelu výhodnější. Nabízí se otestovat i hodnotu menší, ale současný stav dosahuje úspěšnosti 100 %, takže by nebylo dosaženo požadovaného efektu.

Porovnáváním a hledáním ideálních parametrů vznikl model, který často dosahuje maximální úspěšnosti. Navíc celý tento proces vytvořil určitou představu o chování konvoluční neuronové sítě na datové sadě. Nabité znalosti společně se znalostmi o datech je možné zkusit promítnout do modelu a provést ještě nějaké manuální úpravy s cílem zmenšení velikosti a udržení maximální možné úspešnosti, protože v současném stavu má model 44800 parametrů.

Konkrétním prvním krokem je aplikace konvoluce na všechny 3 osy akcelerometru a gyroskopu, protože při pohybu často nastává specifická změna hodnot na všech těchto osách zároveň. Druhou úpravou je ponechání krajních hodnot, protože všechny vzorky mají ručně vycentrovaný začátek, který často představuje signifikantní část pohybu. V první pooling vrstvě vybrat maximum z výsledku konvoluce aplikované na osy akcelerometru a gyroskopu a ve druhé pooling vrstvě vzít maximum z pěti vzorků, což je také krok, který výrazně přispívá na redukci velikosti modelu. Na optimalizaci velikosti pak míří všechny následující úpravy, které začínají zmenšením počtu filtrů druhé konvoluce z 10 na 5. Díky tomu je výsledkem flatten vrstvy vektor o délce 50. Na stejnou velikost je redukována i první plně propojená vrstva. Druhá propojená vrstva se úplně odstraní. Výsledný model vypadá takto:

```
model = Sequential()
model.add(layers.Conv2D(filters=5, kernel_size=(3,3), strides=(1,3), padding = 'same',
                        activation='relu', input_shape=(100,6,1)))
model.add(layers.MaxPooling2D((2,3), padding = 'same'))
model.add(layers.Conv2D(filters=5, kernel_size=(9,1), strides=(1,1), padding = 'same',
                        activation='relu'))
model.add(layers.MaxPooling2D((5,1), padding = 'same'))
model.add(layers.Flatten())
model.add(layers.Dense(units=50, activation='relu'))
model.add(layers.Dense(units=5, activation = 'softmax'))
```

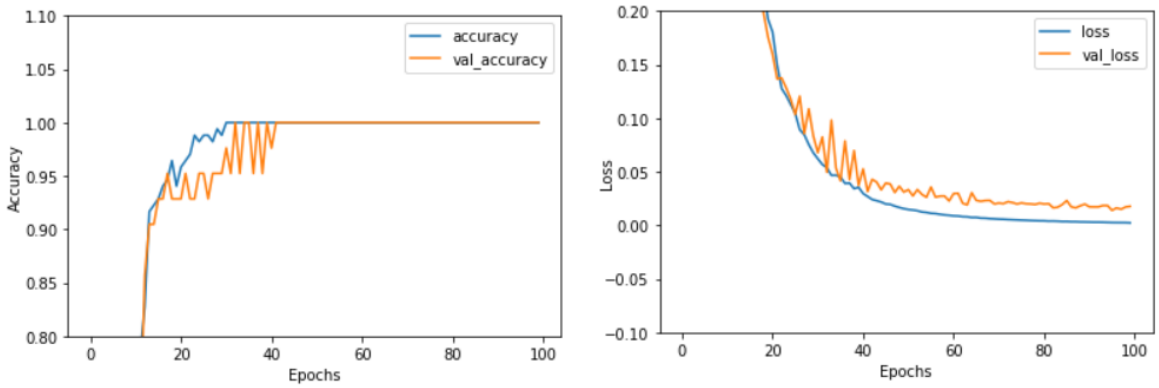
Obrázek 8.2: Výsledná konfigurace konvoluční neuronové sítě

Na řadu přichází testování, kde tento model dosahuje oproti přechozímu ještě větší úspěšnosti, respektive stabilněji a rychleji dochází k úspěšnosti 100 %. Důvodem je především použití konvoluce na všechny tři osy akcelerometru a gyroskopu. Jedná se tak o další úspěch, protože se podařilo zvýšit úspěšnost, a zároveň téměř 15x zmenšit počet parametrů z původních 44800 na 3085.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 100, 2, 5)         50

 max_pooling2d (MaxPooling2D  (None, 50, 1, 5)         0
 )

 conv2d_1 (Conv2D)           (None, 50, 1, 5)          230

 max_pooling2d_1 (MaxPooling  (None, 10, 1, 5)         0
 2D)

 flatten (Flatten)           (None, 50)                0

 dense (Dense)               (None, 50)                2550

 dense_1 (Dense)             (None, 5)                 255

=================================================================
Total params: 3,085
Trainable params: 3,085
Non-trainable params: 0
```

Obrázek 8.3: Souhrn modelu konvoluční neuronové sítě

Tímto je úspěšně vytvořený model konvoluční neuronové sítě a je možné přejít k druhé fázi, kterou je učení. Před tím je ale důležité nalézt vhodný počet epoch, protože v tomto případě není možné zastavit proces školení ve chvíli, kdy začne docházet k přeučení, protože tento jev při 100 % přesnosti nenastává. Nejjednodušším způsobem nalezení dostatečného počtu epoch tak představuje spuštění nadměrně dlouhého tréninku, a na základě zobrazení úspěšnosti v čase pak vybrat ideální hodnotu.



Obrázek 8.4: Úspěšnost modelu

Z grafu je vidět, že k dosažení maximální přesnosti dochází někde kolem 40. epochy. Každý trénink je ale různý, a proto je potřeba vybrat hodnotu s dostatečnou rezervou. V tomto případě je zvolen počet 70.

Zbývá jen připojit hardwarový akcelerátor v podobě grafické karty a spustit 100 různých testování. Protože úspěšnost většinou dosahuje maximální možné hodnoty, je nejlepší ten model, který má minimální výslednou hodnotu loss funkce. Takový model je na konci procesu uložen na Google Disk ve standardním formátu SavedModel.

```
BEST MODEL
train_acc:  1.00
train_loss:  0.010640406049787998
test_acc:  1.00
test_loss:  0.010716889053583145
```

Obrázek 8.5: Úspěšnost nejúspěšnějšího modelu

Poslední fází je převedení modelu do podoby použitelné na ESP32. K tomuto účelu jsou na oficiálních stránkách TensorFlow dostupné potřebné nástroje včetně popisu jejich použití. Prvním nástrojem je převodník modelu na TensotFlow Lite model [34]. Ten je zakomponován do samostatného programu, který načte uložený model z Google Disku, provede konverzi a uloží výsledek uloží zpět, tentokrát do souboru tflite_model.tflite. Pro představu míry redukce má původní model velikost 188 kB a nový pouze 15 kB. Druhý nástroj, který tento redukovaný soubor následně převede do požadovaného formátu v podobě pole bytů programovacího jazyka C, je opět zakomponován do samostatného programu, který zajišťuje načtení Tensorflow Lite modelu z Google Disku, jeho konverzi a uložení výsledku zpět do souboru c_model.c [59]. Konkrétní velikost pole je 15504 B, což je v porovnání s uživatelsky dostupnou RAM pamětí ESP32 přes 120 kB, opravdu malý model.

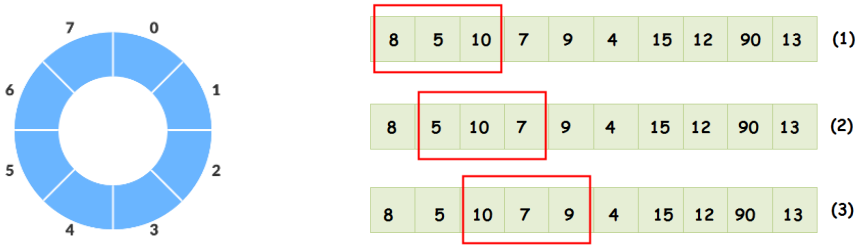
Na závěr je nutné připomenout, že druhá vhodná technika pro řešení úlohy rozpoznávání pohybů prutu byla v kapitole 4.5 vybrána rekurentní neuronová síť typu LSTM. Ta však z důvodu maximální možné úspěšnosti modelu konvoluční neuronové sítě pozbývá možnosti dosažení lepších výsledků, a proto implementována nebude.

9 Software pro detekci

Výsledný program detekčního zařízení je tvořen čtveřicí hlavních bloků, které vzájemně spolupracují. Právě všem těmto dílčím komponentám jsou věnovány následující podkapitoly, přičemž v té poslední je provedeno jejich sjednocení a vyobrazení v podobě blokového schématu celé aplikace. Jako vývojové prostředí je použito PlatformIO a samotná implementace je prováděna ve frameworku Arduino.

9.1 Čtení a příprava dat pro klasifikaci

Jak již bylo zmíněno v práci dříve, tak čtení dat ze senzoru je prováděno s frekvencí 50 Hz a každý pohyb trvá 2 vteřiny, což vytváří posloupnost 100 vzorků. Cílem je v reálném čase klasifikovat jednotlivé pohyby, takže je potřeba využít datovou strukturu, která ve vhodné reprezentaci uchovává data o velikosti jednoho pohybu a zároveň umožňuje pohodlné a rychlé přidávání nových vzorků a odstraňování těch starých. Řešením je kruhová fronta. Do té jsou periodicky přidávány nové vzorky, které v případě nedostatku místa začnou nahrazovat ty nejstarší. Celá funkčnost je tedy na principu posuvného okna, které může být kdykoliv předáno modelu ke klasifikaci.



Obrázek 9.1: Kruhová fronta a posuvné okno [4] [6]

9.2 Nasazení modelu a klasifikace

Tato část je založena na knihovně TensorFlowLite_ESP32 a postup integrace modelu do ESP32 vychází z návodu, který je dostupný na stránkách TensorFlow [31] [58].

Samotný proces integrace modelu do zařízení začíná alokací paměti pro vstupní, výstupní a další pomocná pole, která budou modelem interně používána během výpočtu. Velikost této paměti není jasně dána, a proto je nutné experimentálně najít správnou hodnotu. V tomto případě se jako dostačující jeví velikost 10 kB, ale vzhledem k dostatku paměti byla pro jistotu zvolena hodnota dvojnásobná, tedy 20 kB. Druhým krokem je již samotné načtení modelu ze souboru, který byl dříve vytvořen a obsahuje reprezentaci modelu v podobě pole bajtů. Dále jsou načteny všechny operace, které bude ke své práci potřebovat. Jde o funkcionality, které poskytují jednotlivé vrstvy modelu, takže tento krok vlastě zpřístupňuje použití všech vrstev, ze kterých se model skládá. Alokovaná paměť, model a načtené operace jsou pak předány funkci, která z nich vytvoří jeden velký použitelný celek označovaný jako interpretr. Skrze něj jsou pak prováděny klasifikace. Před tím je ale ještě vytvořen vstupní a výstupní tenzor modelu, což je provedeno alokací v alokované paměti pro model. Nyní je model úspěšně nasazen do zařízení, kompletně nastaven a připraven pro klasifikaci.

Samotná klasifikace pak probíhá provedením trojice úkonů. První je předání vstupních dat interpretovi pomocí ukazatele. Zde je podmínkou správná délka dat a reprezentace v jednorozměrném poli. Druhým krokem je zavolání funkce *Invoke()*, která zprostředkuje klasifikaci a uloží výsledky do výstupního tenzoru. Posledním krokem je pak pouze načtení dat z výstupního tenzoru.

Rychlost celého tohoto procesu klasifikace byla testována a čas strávený výpočtem se pohybuje kolo 16 milisekund, což vzhledem ke vzorkovací frekvenci 50 Hz dovoluje provádět

predikci v každém kroku. Rychlost ale v tomto případě nehraje zásadní roli, a proto je nakonec stanovena četnost predikce po pěti krocích, tedy každých 100 milisekund. Taková frekvence vyhodnocování je naprosto dostatečná, a navíc nechává velkou časovou rezervu na vykonávání jiných procesů.

9.3 Distribuce výsledků

Vzhledem k typu zařízení je bezdrátová komunikace jedinou možností, jak být v kontaktu s okolním světem. To platí především v případě výsledků, které je potřeba dopravit na nějaké cílové zařízení, ať už se jedná o počítač, telefon nebo vestavěné zařízení. Důležitá je ale volba konkrétního komunikačního protokolu. Zde volba padla na ESP-NOW, a to hned z několika důvodů, které vyplývají z podstaty tohoto produktu.

Jedná o koncové zařízení se senzorem, u kterého se předpokládá, že bude integrováno do komplexnějšího ekosystému s větším množstvím podobných snímačů, které budou svými výsledky vytvářet co nejlepší popis prováděného lovu. Důraz je tedy kladen především na jednoduchost a rychlost přenosu, což je oblast, ve které ESP-NOW vyniká. Další velkou výhodou je protokolové oddělení komunikace mezi těmito koncovými zařízeními a uživatelem, protože ve zmíněném ekosystému by tato koncová zařízení mohla odesílat výsledky ke zpracování na hlavní prvek pomocí ESP-NOW, a ten by pak následně komunikoval s uživatelem některým univerzálnějším protokolem jako je Bluetooth nebo WiFi. Právě toto schéma bylo využito i v případě sběru dat. Na to navazuje další pozitivum, kterým je použití jen jednoho protokolu mezi všemi koncovými zařízeními v celém projektu.

Pokud by měla být pozornost přesunuta k samotné implementaci komunikace přes ESP-NOW, tak prvním krokem je učinit rozhodnutí, o jaký typ komunikace se jedná, a pak už stačí pouze specifikovat co a kam je potřeba posílat. V tomto případě se jedná o jednosměrnou komunikaci, kdy detekční zařízení pracuje pouze jako vysílač. Zpráva, která bude odesílána, obsahuje informaci, zda se jedná o kontrolní zprávu typu ping, dále pak konkrétní pohyb a úspěšnost, se kterou byl daný pohyb detekován. Zbývá specifikovat přijímací stranu. To je v případě ESP-NOW prováděno pomocí MAC adresy, kterou je možné v případě ESP32 získat výpisem do terminálu pomocí programu 7.2. Samotná distribuce výsledků pak probíhá nastavením požadovaných hodnot do vytvořené zprávy a zavoláním metody *esp_now_send(…)*, která zprostředkuje odeslání.

9.4 Detekce stavu baterie

Stanovení zbývající kapacity začíná vždy tak, že je k baterii připojen měřící obvod v podobě odporového děliče, který je tvořený dvěma stejně velkými rezistory. Následuje měření napětí na spodním rezistoru pomocí analogově digitálního převodníku. Obdržená hodnota je vzhledem k dělícímu poměru děliče 1:1 vynásobena dvěma, čímž dochází k výpočtu napěťové úrovně baterie. Následuje konverze napětí na kapacitu pomocí převodní tabulky. Tato tabulka byla vytvořena ručně na základě vybíjecí charakteristiky Li-ion baterie na obrázku 5.4 a obsahuje dvojice {spodní úroveň napětí, kapacita}, které zprostředkovávají orientační kapacitu baterie s přesností na desítky procent.

Spodní úroveň napětí [mV]	Kapacita [%]
0	0
3500	10
3550	20
3600	30
3650	40
3700	50
3750	60
3800	70
3850	80
3900	90
4000	100

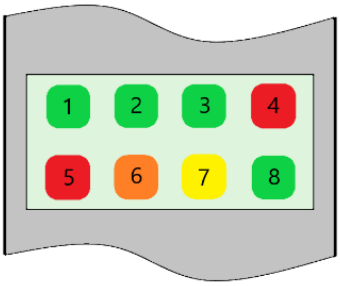
Tabulka 9.1: Převodní tabulka

Výsledkem detekce je sice pouze orientační hodnota, která se od té skutečné pravděpodobně částečně odlišuje, ale pro účely této aplikace jde o řešení dostatečné, protože v tomto případě je stav baterie signalizován uživateli pomocí čtveřice LED diod, kde se případná malá odchylka nijak neprojeví.

9.5 Informování uživatele

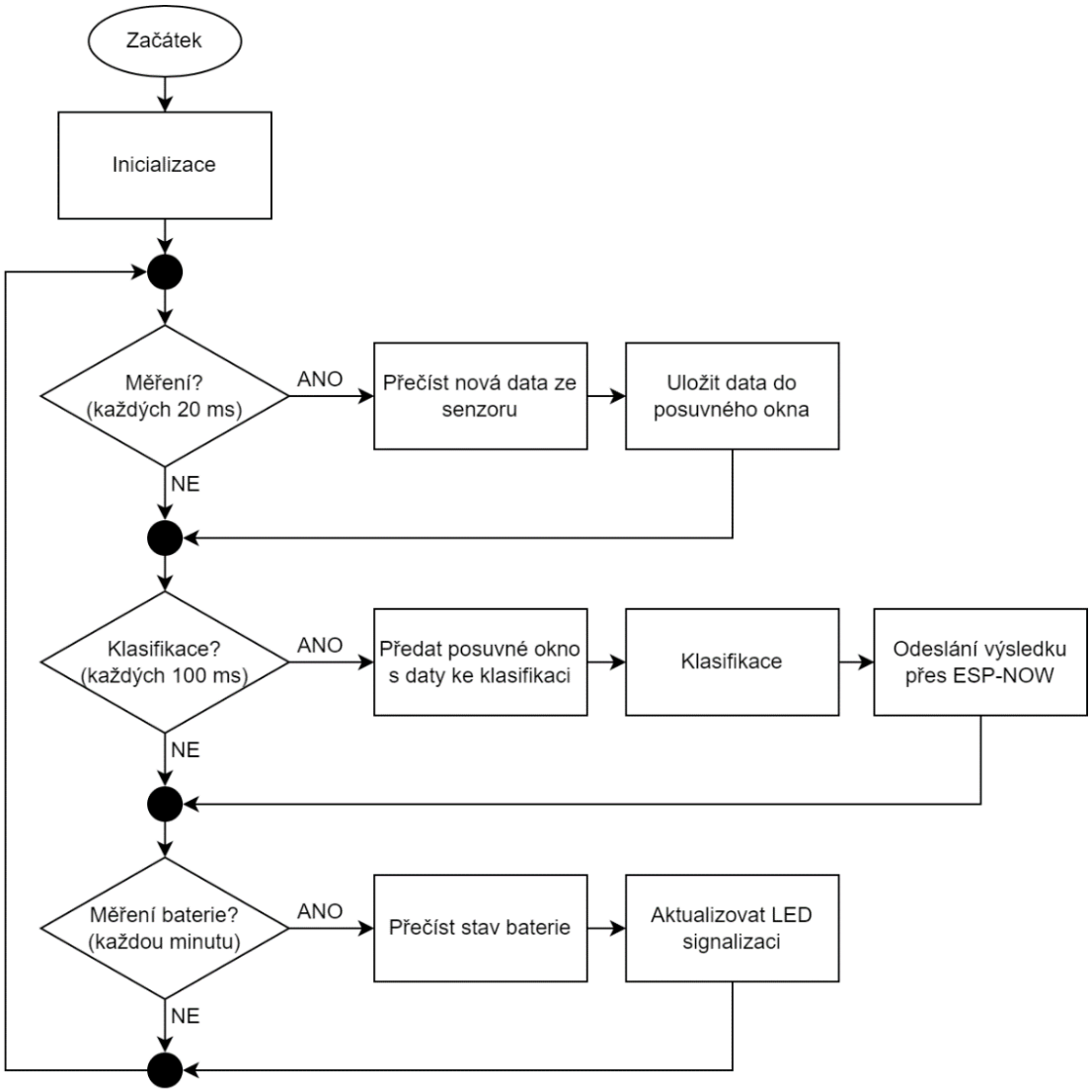
Zařízení disponuje osmi LED diodami umístěnými pod plexisklem na horní straně krabičky. Ty mají za úkol vizuálně informovat uživatele.

První dioda pracuje jako status běžícího zařízení. Druhá signalizuje, že je v blízkosti dostupné cílové zařízení, kterému mají být odesílána data. Zjištění této informace probíhá odesláním zprávy typu ping přes ESP-NOW, která musí být přijímači úspěšně doručena. Pokud se tak nestane, přijímací zařízení není v dosahu nebo je vypnuté. V opačném případě je rozsvícena právě druhá LED dioda, která signalizuje, že by měl být přijímač dostupný. Třetí dioda je aktivní s odesíláním dat přes ESP-NOW. Poslední dioda v horním řádku je spuštěna pouze v případě, že nastala nějaká fatální chyba a zařízení přestalo pracovat. Typickým příkladem může být neúspěch úvodního nastavení inerciální měřící jednotky. Spodní řádek obsahující také čtyři LED diody má celý jedinou funkci, a sice zobrazení stavu baterie.



Obrázek 9.2: Ilustrace informačních diod detekčního zařízení

9.6 Blokové schéma programu



Obrázek 9.3: Blokové schéma programu detekčního zařízení

Po spuštění je vykonána klasická úvodní inicializace všech potřebných komponent celého programu. V tomto případě se jedná o vytvoření a inicializaci všech globálních proměnných a objektů, nastavení možnosti výpisu na sériový monitor přes rozhraní UART, inicializaci a vypnutí signalizačních LED diod, změření aktuálního stavu baterie, inicializaci inerciální měřící jednotky, inicializaci TensorFlow Lite zajišťující mimo jiné i načtení a přípravu modelu k použití. Dále pak inicializace pokračuje zjištěním velikosti výstupního vektoru modelu, inicializací ESP-NOW, kontrolou aktivity přijímače pomocí ping zprávy přes ESP-NOW, a nakonec ještě naplněním posuvného okna daty z inerciální měřící jednotky.

Dále pokračuje již hlavní část programu. Jedná se o nekonečnou smyčku, ve které jsou v nastavených časových intervalech periodicky vykonávány jednotlivé úkony. Tím nejčastějším je čtení dat z inerciální měřící jednotky, které je vzhledem k definované vzorkovací frekvenci 50 Hz prováděno každých 20 milisekund. Po pěti vzorcích, tedy každých 100 milisekund, jsou pak data uložená v plovoucím okně předána modelu k vyhodnocení. Obdržený výsledek je společně s jeho přesností odeslán přes ESP-NOW přijímacímu zařízení. Třetí akcí je aktualizace stavu baterie. To je prováděno každou minutu.

10 Testování a vyhodnocení výsledků

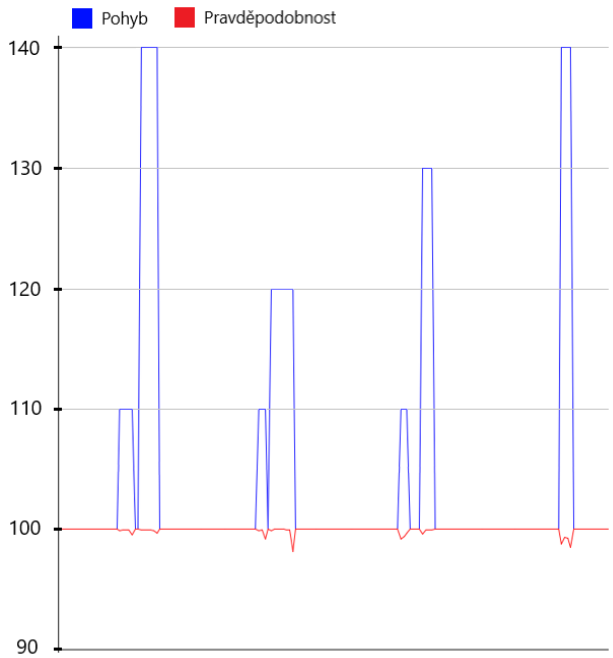
Před začátkem testování je nutné zvolit vhodný způsob vizualizace, se kterým bude možné provádět snadné, rychlé a jednoznačné vyhodnocení dosažených výsledků. K tomu se také vážou specifické požadavky. Hned tím prvním je samotný způsob komunikace v podobě protokolu ESP-NOW. Druhá podmínka představuje okamžité zobrazení obdržených výsledků v reálném čase. To platí jak pro číslo konkrétního pohybu, tak pro pravděpodobnost, se kterou byl pohyb detekován. V neposlední řadě musí být zobrazení dostatečně čitelné. Ideálním případem je pak možnost zastavení vizualizace, aby bylo možné provést podrobnější prozkoumání.

Jednoduché řešení, které splňuje všechna stanovená kritéria, je založeno na vývojové desce s ESP32 modulem. Ta v první řadě pracuje jako přijímač zpráv z detekčního zařízení přes ESP-NOW. Obdržená data pak následně vhodným způsobem upraví, aby byla zlepšena jejich čitelnost při grafickém zobrazení. Tento proces se skládá ze dvou částí. Tím prvním je filtrace, která propouští pouze pohyby detekované s vyšší pravděpodobností, než je nastavená hranice. Vše ostatní je považováno za žádný pohyb s detekovanou pravděpodobností 100 %. Po filtraci následuje změna rozsahů, která jednotlivé pohyby mapuje z intervalu <0;4> do intervalu <100;140> a pravděpodobnost převádí z dvoumístných desetinných čísel na procenta. Na závěr jsou takto připravená data odeslána přes UART rozhraní do počítače, ke kterému je vývojová deska s ESP32 připojena skrze USB kabel. V reálném čase pak dochází k vynášení výsledků do grafu pomocí nástroje sériový ploter, který je součástí programu Arduino IDE.

Samotné testování začalo hledáním ideální filtrační hranice. Počáteční hodnota byla tedy nastavena na pravděpodobnost 80 %. Grafické zobrazení s touto hranicí však bylo poměrně nečitelné, protože jak v průběhu času docházelo ke vkládání nových dat do posuvného okna a tím i jeho pomyslnému pohybu, tak nastával předpokládaný jev, a sice postupné zvyšování a následné zmenšování pravděpodobnosti, což byla reakce na to, jak velká část pohybu se v posuvném okně zrovna nacházela. Filtrační hranice byla tedy postupně navyšována, a tím docházelo k z pomyslnému zmenšování intervalu spolehlivosti generovaného normálního rozložení pravděpodobnosti. Za výslednou hranici byla nakonec zvolena pravděpodobnost 98 %, což je hodnota, při které dochází ke spolehlivému rozpoznání všech pohybů, a zároveň je minimalizováno riziko chybných detekcí. S tímto nastavením byl proveden hodinový testovací lov, během kterého byla na obrazovce notebooku vizuálně kontrolována detekce pohybů na grafu s vykreslováním v reálném čase. Výsledkem bylo správné rozpoznání všech provedených klíčových pohybů.

Obrázek 10.1: Konfigurace při testování v praxi

Funkčnost však není ideální, protože správné rozpoznání pohybu je často doprovázeno druhou detekcí, která je chybná. Důvod je ten, že ve skutečnosti trvá pohyb například 5 sekund, ale k jeho rozpoznání je zvolena jen signifikantní část o dvou sekundách, takže zbývající doba představuje dostatečný prostor na to, aby se v něm mohly nacházet pohyby připomínající signifikantní část pohybů jiných. Jak je tedy posuvné okno v čase plněno, tak se v něm v určitou chvíli vyskytne část pohybu připomínající pohyb jiný, a dojde k chybné detekci. Typickým příkladem je nahození. Pro něj jsou signifikantní první 2 sekundy, kdy dochází ke švihu a ohození zátěže. Bezprostředně po tom je však prut přizvednut kolmo vzhůru do doby, než se zátěž dostane do požadované vzdálenosti, zastaví se o klip a svou vahou stáhne špičku prutu zpět mírně k zemi. Právě tato druhá část pohybu obsahující zvednutí a mírné položení je velmi podobná stažení montáže, kdy je prut přizvednut a následně se během namotávání vlasce postupně přibližuje špičkou vodní hladině. Výsledkem detekce zařízení je tedy nahození a několik milisekund poté je chybně rozpoznáno stažení. Tento jev je společně se všemi ostatními zachycen na následujícím obrázku.



Obrázek 10.2: Výsledky

Řešení tohoto problému s přidruženými chybami je však naprosto jednoduché. Stačí pouze zavést na přijímači několik pravidel. Tím prvním by mohlo být krátké ignorování přijatých zpráv po obdržení některého z pohybů, což by vyřešilo problém nahození. Druhé pravidlo by pak vycházelo ze znalosti lovu a říkalo, že po nahození může následovat jakýkoliv z pohybů kromě nahození, a zároveň po jakémkoliv z pohybů kromě nahození může následovat jen nahození. Další možností je převedení sdružených dvojic na jeden správně detekovaný pohyb. Obě tyto uvedené úpravy by měly zajistit očekávanou správnou funkčnost detekčního zařízení.

Poslední otázkou zůstává, jestli na tento druh problému s přidruženými chybami nemá zásadní vliv samotná datová sada, protože při její tvorbě byly odstraněny všechny koncové části pohybů, které se nevešly do prvních dvou sekund. Další možností by bylo tyto části označit jako klidový stav a následně použít při trénování modelu. Vzhledem k tomu, že jde ale pouze o konce pohybů, který je chybně klasifikován jen u nahození, tak by tato změna pravděpodobně nevedla k rapidnímu zlepšení. Přesto je šance, že by mohlo dojít alespoň k pozitivnímu posunu v případě náhozu. Na druhou stranu je také možné, že touto změnou bude vnesen do tréninku větší chaos a celková úspěšnost naopak klesne. Každopádně se jedná o další možnost modifikace, která by mohla být předmětem budoucího vývoje.

11 Možnosti budoucího rozšíření

Již v průběhu celé práce byly zmiňovány možnosti případných modifikací, různá jiná řešení, prostory pro další rozšíření a podobně. Tato kapitola představuje seznam těch nejzásadnějších úprav, které by měly mít největší pozitivní vliv na funkčnost celého produktu.

S tou první v podobě optimalizace spotřeby bylo počítáno již v při samotném hardwarovém návrhu, a proto byly k ESP32 připojeny piny inerciální měřící jednotky označené jako přerušení. Zvolená inerciální měřicí jednotka BMI160 totiž disponuje speciálními integrovanými funkcemi, kde jednou z nich je i vygenerování impulzu přerušení v případě detekce pohybu [50]. Tento impulz by mohl být využíván k buzení ESP32, které by tak po dobu, kdy nedochází k žádnému pohybu, mohlo být přepnuto do režimu deep sleep s téměř nulovou spotřebou. Problém této myšlenky je doba spuštění ESP32. Ta se pohybuje v řádu stovek milisekund, což by způsobilo, že by značná část pohybu nebyla zachycena. Řešení by však mohlo být využití integrovaného bufferu v inerciální měřící jednotce [50]. Tento koncept s nečinností zařízení během doby, kdy nedochází k žádnému pohybu, by v případě feederového lovu, kdy je prut většinu času v klidu, představoval zásadní redukci spotřeby a tím pádem i kapacity baterie. Výsledkem by tedy bylo mnohem menší zařízení, což je vzhledem k jeho použití naprosto zásadní výhodou.

Další případné rozšíření by mohlo být věnováno expanzi datové sady, která je přímo odpovědná za úspěšnost detekce celého produktu. Především by mělo být provedeno snímání několika různých rybářů, což by zamezilo výskytu případných problémů ve chvíli, kdy má jedinec zásadním způsobem odlišnou techniku manipulace s prutem, což je situace, která může nastat.
S budoucí výrobou většího množství detekčních zařízení by se nabízelo rozšíření v podobě využití původně vybraného a následně zavrhnutého pohybu, a sice odložení prutu. Konkrétní představa je taková, že rybář má pro lov připraveno několik prutů, ze kterých je v jednu chvíli používán pouze jeden, maximálně dva. Cílem detekčního zařízení by pak bylo automatické uspání na základě rozpoznání pohybu odložení. Díky tomu by nedocházelo ke zbytečné spotřebně baterie a zároveň by byla vygenerována další důležitá informace o tom, že uživatel změnil prut A za prut B.

V neposlední řadě by také bylo vhodné přidat funkcionalitu pro uchování výsledků přímo v detekčním zařízení. Pokud by totiž nastala situace v podobě výpadku přijímacího zařízení, tak by byla odesílaná data ztracena. Řešení ale nepředstavuje pouze naivní uchování výsledků v RAM paměti detekčního zařízení do doby, než bude přijímač dostupný, protože v případě vypnutí by došlo opět ke ztrátě dat. Ideální by bylo sofistikovanější řešení trvalého uchování v podobě zápisu na flash paměť. Tam by mohl být vymezen prostor o určité velikosti s postupným přepisováním nejstarších záznamů. Mezi vysílačem a přijímačem by pak musela probíhat nějaká další komunikace pro zajištění synchronizace výsledků.

12 Závěr

Cílem této diplomové práce bylo vytvořit zařízení, které využívá metody strojového učení k rozpoznávání pohybů feederového rybářského prutu na základě dat z inerciální měřící jednotky.

Celý proces začal seznámením se základními aspekty rybolovné techniky feeder. Na základě obdržených znalostí pak bylo rozhodnuto, že nejlepší možné místo pro upevnění detekčního zařízení na prut představuje nevyužitá oblast mezi rukojetí a prvním očkem. Společně s tím byla vybrána pětice klíčových pohybů pro klasifikaci – nahození, krmení, zaseknutí, stažení a odložení. Následovalo vytvoření teoretického základu v oblasti strojového učení, kde byly představeny současné možnosti použití těchto metod na energeticky nenáročných zařízení. Na to navázal výběr konkrétních vhodných technik pro detekci jednotlivých pohybů prutu. Tato část zahrnovala specifikaci úlohy a seznámení s problematikou klasifikace, inerciální měřící jednotkou a úlohou rozpoznávání lidské činnosti. Výsledkem byl výběr dvojice metod hlubokého strojového učení v podobě konvoluční neuronové sítě a rekurentní neuronové sítě typu LSTM.

Práce pokračovala praktickou částí, konkrétně tedy návrhem a tvorbou detekčního zařízení. To bylo nejprve použito jako snímač, který ve spolupráci s bezdrátovým označováním pohybů pomocí dalšího zpracovaného zařízení zaznamenával data během lovu. Vhodným zpracováním obdržených výsledků pak byla vytvořena datová sada. Následovala implementace konvoluční neuronové sítě, proces jejího učení na vytvořené datové sadě, a nakonec převod nejúspěšnějšího modelu do podoby umožňující nasazení do detekčního zařízení. V další fázi byl vytvořen výsledný software pro detekční zařízení. Ten se skládá z pěti hlavních bloků – čtení a příprava dat pro klasifikaci založená na principu posuvného okna, nasazení modelu a klasifikace pohybů, distribuce výsledků přes ESP-NOW, detekce stavu baterie zařízení a informování uživatele pomocí jednoduchého rozhraní, které se skládá z osmi LED diod. Poté byla funkčnost a úspěšnost produktu otestována v praxi. To probíhalo tak, že detekční zařízení přidělané na prutu odesílalo během lovu data přes ESP-NOW na vývojovou desku s modulem ESP32, která přes připojený USB kabel přenášela obdržené informace do počítače, kde docházelo k jejich zobrazení v reálném čase pomocí grafu. Obdržené výsledky ukázaly, že zařízení rozpoznává jednotlivé pohyby s vysokou úspěšností. Společně s tím byla ale také odhalena přidružená chybná detekce během většiny pohybů. Předpokládaný důvod vzniku tohoto problému byl popsán a společně s tím i představen způsob řešení, který by měl všechny chyby eliminovat. Nakonec byly diskutovány možnosti budoucího vývoje vytvořeného produktu.

Výsledkem této práce je malé, bateriově napájené zařízení vytvořené na platformě ESP32, které po připevnění na libovolný feederový prut umožňuje pracovat v jednom ze dvou režimů. Tím prvním je záznam pohybu, kdy ve spolupráci s vytvořeným označovacím zařízením, následným stažením dat přes integrovaný webový server a zpracováním pomocí připravených programů je možné vytvářet nebo rozšiřovat datovou sadu. Druhý režim pak slouží k samotnému rozpoznávání pohybů prutu během lovu. V tomto směru poskytuje zařízení velmi vysokou úspěšnost detekce díky integrovanému modelu konvoluční neuronové sítě, který byl pro tuto úlohu specificky navržen a následně natrénován na vytvořené datové sadě. Navíc je zařízení vybaveno možností odesílat výsledky na vybrané přijímací zařízení pomocí bezdrátové komunikace v podobě ESP-NOW.

Jedná se o produkt, který je ve své současné podobě připravený pro použití v praxi ať už jako samostatný prvek v kombinaci s přijímačem, tak jako koncové zařízení, které je možné integrovat do nějakého komplexnějšího ekosystému. Bonusem k tomu všemu je navíc dostatečný prostor pro případná další rozšíření, která mají velký potenciál zásadním způsobem zlepšit některé vlastnosti tohoto produktu.

Seznam použitých zdrojů

[1] 3.7v Lithium Polymer Battery Cell. [Online] 7. 1 2022. Dostupné z: https://www.dnkpower.com/products/3-7v-lithium-polymer-battery-cell/.

[2] Asiri, Sidath. Machine Learning Classifiers. Towards Data Science. [Online] 11. 6 2018. [Citace: 10. 12 2021.] Dostupné z: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623.

[3] Authors, TensorFlow. *all_ops_resolver.cc.* GitHub. [Online] [Citace: 3. 5 2022.] Dostupné z: https://github.com/tensorflow/tflite-micro/blob/main/tensorflow/lite/micro/all_ops_resolver.cc.

[4] Build and convert models. TensorFlow. [Online] [Citace: 27. 4 2022.] Dostupné z: https://www.tensorflow.org/lite/microcontrollers/build_convert.

[5] ÇELİK, Özer. *A Research on Machine Learning Methods and Its Applications.* Journal of Educational Technology & Online Learning. [Online] 30. 8 2018. [Citace: 11. 11 2021.] Dostupné z: https://dergipark.org.tr/en/download/article-file/532680.

[6] Circular Queue Data Structure. Programiz. [Online] [Citace: 22. 3 2022.] Dostupné z: https://www.programiz.com/dsa/circular-queue.

[7] Custom 18650 Battery Packs. [Online] [Citace: 7. 1 2022.] Dostupné z: https://www.dnkpower.com/li-ion-18650-battery/.

[8] Česenek, David. *Inertial measurement unit modeling*. [Online] 5 2019. [Citace: 10. 12 2021.] Dostupné z: https://dspace.cvut.cz/bitstream/handle/10467/83404/F3-DP-2019-Cesenek-David-master_thesis_imu_modeling_cesenek_final-merged.pdf?sequence=-1&isAllowed=y.

[9] Designing Applications with Li-ion Batteries. [Online] [Citace: 7. 1 2022.] Dostupné z: https://www.richtek.com/battery-management/en/designing-liion.html.

[10] douce, FIPSED - FÉDÉRATION INTERNATIONALE DE LA PÊCHE SPORTIVE en eau. *9th Feeder Fishing World Championship – final results*. FIPSED - FÉDÉRATION INTERNATIONALE DE LA PÊCHE SPORTIVE en eau douce. [Online] 18. 2 2019. [Citace: 18. 10 2021.] Dostupné z: http://www.fips-ed.com/fipsed/index.php/en/homepage/news/500-feeder-2019-final.

[11] Drábek, Pavel. Recenze: *Prut Preston Tyson Carp Feeder 10 ft*. NaFeeder.cz. [Online] 23. 5 2019. [Citace: 23. 10 2021.] Dostupné z: https://nafeeder.cz/recenze-prut-preston-tyson-carp-feeder-10-ft/.

[12] Education, IBM Cloud. *Overfitting*. IBM. [Online] 3. 3 2021. [Citace: 21. 12 2021.] Dostupné z: https://www.ibm.com/cloud/learn/overfitting.

[13] Energy, Office of Nuclear. *INFOGRAPHIC: How Much Power Does A Nuclear Reactor Produce?* Office of Nuclear Energy. [Online] 31. 3 2021. [Citace: 10. 12 2021.] Dostupné z: https://www.energy.gov/ne/articles/infographic-how-much-power-does-nuclear-reactor-produce.

[14] ESP32 Hardware Design Guidelines. [Online] 9. 8 2021. [Citace: 12. 1 2020.] Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32_hardware_design_guidelines_en.pdf.

[15] ESP32 Series Datasheet. [Online] 10 2021. [Citace: 5. 1 2022.] Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[16] ESP-NOW. *ESP-IDF Programming guide*. [Online] Espressif. [Citace: 10. 2 2022.] Dostupné z: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_now.html.

[17] Feeder, Fishing. *World Championship Feeder 2011 - Italia*. Fishing Feeder. [Online] 2. 9 2011. [Citace: 18. 10 2021.] Dostupné z: http://www.fishing-feeder.cz/cs/mdl/info/world-championship-feeder-2011---italia.

[18] Frank Rosenblatt. Wikipedia. [Online] [Citace: 11. 11 2021.] Dostupné z: https://en.wikipedia.org/wiki/Frank_Rosenblatt.

[19] Get started with microcontrollers. TensorFlow. [Online] [Citace: 27. 4 2022.] Dostupné z: https://www.tensorflow.org/lite/microcontrollers/get_started_low_level.

[20] GRZYBOWSKÁ, Bc. MARTINA. *HUMAN MO-CAP SYSTEM BASED ON INERTIAL*. [Online] 2021. Dostupné z: https://www.fit.vut.cz/study/thesis-file/20602/20602.pdf.

[21] Han, Odongo Steven Eyobu and Dong Seog. *Feature Representation and Data Augmentation for Human Activity Classification Based on Wearable IMU Sensor Data Using a Deep LSTM Neural Network*. [Online] 31. 8 2018. [Citace: 10. 12 2021.] Dostupné z: https://www.mdpi.com/1424-8220/18/9/2892/htm#sec5-sensors-18-02892.

[22] Hitchcock, Ed. *Fishing Rod Action & Power – Choose the Best Rod for You*. Tailored Tackle. [Online] 1. 10 2018. [Citace: 23. 10 2021.] Dostupné z: https://tailoredtackle.com/fishing-rod-action-and-power/#prettyPhoto.

[23] Horáček, Miroslav. *Základy feederu – 3. díl - Feederový prut*. chytej.cz. [Online] 28. 5 2011. [Citace: 25. 10 2021.] Dostupné z: https://www.chytej.cz/clanky/1081/zaklady-feederu-3-dil-feederovy-prut/.

[24] Horáček, Miroslav. *Základy feederu – 4. díl - Feederové špičky*. chytej.cz. [Online] 3. 10 2011. [Citace: 25. 10 2021.] Dostupné z: https://www.chytej.cz/clanky/1099/zaklady-feederu-4-dil-feederove-spicky/.

[25] Chauvin, Yves, and David E. Rumelhart, eds. *Backpropagation: theory, architectures, and applications*. Lawrence Erlbaum Associates, 1995.

[26] Chen, Yuwen, et al. *LSTM networks for mobile human activity recognition*. [Online] 2016. [Citace: 10. 12 2021.] Dostupné z: https://scholar.google.com/scholar?hl=cs&as_sdt=0%2C5&q=LSTM+Networks+for+Mobile+Human+Activity+Recognition&btnG=.

[27] Jindong Wang, et al. *Deep Learning for Sensor-based Activity Recognition: A Survey*. [Online] [Citace: 10. 12 2021.] Dostupné z: https://arxiv.org/pdf/1707.03502.pdf.

[28] M., Jun. *Build your own neural network classifier in R*. Towards Data Science. [Online] 28. 4 2019. [Citace: 21. 12 2021.] Dostupné z: https://towardsdatascience.com/build-your-own-neural-network-classifier-in-r-b7f1f183261d.

[29] Machine Learning: *How to Prevent Overfitting*. Start it up. [Online] 30. 1 2021. Dostupné z: https://medium.com/swlh/machine-learning-how-to-prevent-overfitting-fdf759cc00a9.

[30] Matthew Stewart, PhD. *Tiny Machine Learning: The Next AI Revolution*. Towards Data Science. [Online] 2. 10 2020. [Citace: 10. 12 2021.] Dostupné z: https://towardsdatascience.com/tiny-machine-learning-the-next-ai-revolution-495c26463868.

[31] Maximum Element Sliding Window. Techie Me. [Online] [Citace: 22. 3 2022.] Dostupné z: http://techieme.in/maximum-element-sliding-window/.

[32] Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar. *Foundation of Machine Learning*. second. MIT Press, 2018.

[33] MICROCHIP. MCP73830/L Datasheet. [Online] 14. 5 19. [Citace: 11. 1 2020.] Dostupné z: https://ww1.microchip.com/downloads/en/DeviceDoc/MCP73830-L-Data-Sheet-DS20005049E.pdf.

[34] Model training APIs. Keras. [Online] [Citace: 20. 4 2022.] Dostupné z: https://keras.io/api/models/model_training_apis/.

[35] Moor, James H. *The Turing Test: The Elusive Standard of Artificial Intelligence*. Kluwer Academic Publishers, 2003.

[36] Moor, James. *The Dartmouth College Artificial Intelligence Conference: The Next Fifty Years*. AI Magazine, 2006. 4.

[37] Olah, Christopher. *Understanding LSTM Networks*. [Online] 27. 8 2015. [Citace: 10. 12 2021.] Dostupné z: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[38] Osička, Petr. *Strojové učení a deep learning*. Přírodovědecká fakulta UP, Olomouc. [Online] Dostupné z: https://mfi.upol.cz/files/30/3002/mfi_3002_133_145.pdf.

[39] Osisanwo, F. Y., et al. *Supervised Machine Learning Algorithms: Classification and Comparison*. [Online] International Journal of Computer Trends and Technology (IJCTT), 3. 11 2017. [Citace: 10. 12 2021.] Dostupné z: https://www.researchgate.net/profile/J-E-T-Akinsola/publication/318338750_Supervised_Machine_Learning_Algorithms_Classification_and_Comparison/links/596481dd0f7e9b819497e265/Supervised-Machine-Learning-Algorithms-Classification-and-Comparison.pdf

[40] Ouředníček, Jiří. *FEEDER ZÁKLADY 1 – od úplného začátku*. O rybách a rybaření. [Online] 14. 12 2019. [Citace: 14. 10 2021.] Dostupné z: https://rybarina.fousuvcancak.cz/20520/feeder-od-zacatku-001/.

[41] Papert, Marvin L. Minsky and Seymour A. *Perceptrons: expanded edition*. 1988.

[42] PCB width trace calculator. Advanced Circuits. [Online] Dostupné z: https://www.4pcb.com/trace-width-calculator.html.

[43] Machine Learning: How to Prevent Overfitting. Start it up. [Online] 30. 1 2021. Dostupné z: https://medium.com/swlh/machine-learning-how-to-prevent-overfitting-fdf759cc00a9.

[44] Pra, Marco Del. *Time Series Classification with Deep Learning*. Towards Data Science. [Online] 8. 8 2020. [Citace: 10. 12 2021.] Dostupné z: https://towardsdatascience.com/time-series-classification-with-deep-learning-d238f0147d6f.

[45] PRESTON INNOVATIONS MONSTER WANDZEE REVIEW. ANGLING TIMES. [Online] 16. 7 2018. [Citace: 23. 10 2021.] Dostupné z: https://bauer-xcel-8aey.squarespace.com/fishing-tackle/rods/articles?offset=1531739795680&category=Rods.

[46] Přehled nejdůležitějších ustanovení zákona č. 99/2004 Sb. a vyhlášky č. 197/2004 Sb., ve znění pozdějších předpisů. ČRS - Český rybářský svaz. [Online] 1. 1 2022. [Citace: 1. 5 2022.] Dostupné z: https://www.rybsvaz.cz/beta/index.php.

[47] ROBU.IN. Lithium-ion Battery vs Lithium-polymer Battery. ROBU.IN. [Online] 20. 4 2020. [Citace: 11. 1 2021.] Dostupné z: https://robu.in/lithium-ion-battery-vs-li-po-battery/.

[48] Rogers, Andrew. *Introduction to USB Type-C*. [Online] 2015. Dostupné z: https://cdn.sparkfun.com/assets/e/b/4/f/7/USB-C_Datasheet.pdf.

[49] Samsung. *Introduction of INR18650-25R*. [Online] 2013. [Citace: 11. 1 2020.] Dostupné z: https://www.powerstream.com/p/INR18650-25R-datasheet.pdf.

[50] Sensortec, Bosch. *BMI160 Datasheet*. [Online] 25. 11 2020. [Citace: 11. 1 2020.] Dostupné z: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi160-ds000.pdf.

[51] Seznam registrovaných závodníků. ČRS - Český rybářský svaz. [Online] 13. 10 2021. [Citace: 14. 10 2021.] Dostupné z: https://www.rybsvaz.cz/beta/index.php/o-crs/sport-a-mezinarodni-cinnost/seznam-registrovanych-sportovcu.

[52] Schwartz, Ocsar. *In the 17th Century, Leibniz Dreamed of a Machine That Could Calculate Ideas*. IEEE Spectrum. [Online] 4. 11 2019. [Citace: 11. 11 2021.] Dostupné z: https://spectrum.ieee.org/in-the-17th-century-leibniz-dreamed-of-a-machine-that-could-calculate-ideas.

[53] SMTECH. *18650 Battery How to Charge – 5 Simple Solutions*. SMTECH. [Online] [Citace: 11. 1 2020.] Dostupné z: https://somanytech.com/18650-battery-how-to-charge-18650-battery-with-charger-and-without-charger/.

[54] Spector, Lee. *Evolution of artificial intelligence. ScienceDirect*. [Online] 7. 11 2006. [Citace: 11. 11 2021.] Dostupné z: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.482.1549&rep=rep1&type=pdf.

[55] ST. STM32 High Performance MCUs. ST. [Online] [Citace: 5. 1 2022.] Dostupné z: https://www.st.com/en/microcontrollers-microprocessors/stm32-high-performance-mcus.html.

[56] svaz, Český rybářský. Soutěžní řád. ČRS - Český rybářský svaz. [Online] 3. 4 2021. [Citace: 14. 10 2021.] Dostupné z: https://www.rybsvaz.cz/beta/index.php/o-crs/sport-a-mezinarodni-cinnost/lru-feeder/category/22-soutezni-a-zavodni-rad.

[57] Systems, Espressif. *ESP32¬WROOM¬32E/ESP32¬WROOM¬32UE Datasheet*. Espressif Systems. [Online] 8. 11 2021. [Citace: 5. 1 2022.] Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf.

[58] tanakamasayuki. *TensorFlowLite_ESP32*. GitHub. [Online] [Citace: 27. 4 2022.] Dostupné z: https://github.com/tanakamasayuki/Arduino_TensorFlowLite_ESP32.

[59] TensorFlow Lite converter. TensorFlow. [Online] [Citace: 27. 4 2022.] Dostupné z: https://www.tensorflow.org/lite/convert/index.

[60] TensorFlow. *TensorFlow Lite for Microcontrollers*. TensorFlow. [Online] 1. 9 2021. [Citace: 5. 1 2022.] Dostupné z: https://www.tensorflow.org/lite/microcontrollers.

[61] Thuillier, Jules. *Handling IMU Drift*. VR Tracker. [Online] 10. 8 2018. [Citace: 21. 12 2021.] Dostupné z: https://vrtracker.xyz/handling-imu-drift/.

[62] TIMES, ANGLING. *Why are feeder matches so popular?* ANGLING TIMES. [Online] 15. 4 2021. [Citace: 23. 10 2021.] Recenze: Prut Preston Tyson Carp Feeder 10 ft.

[63] Torex. XC6210 Series Datasheet. [Online] [Citace: 12. 1 2020.] Dostupné z: https://www.torexsemi.com/file/xc6210/XC6210.pdf.

[64] tutorials, Random nerd. *ESP32 ADC – Read Analog Values with Arduino IDE*. Random nerd tutorials. [Online] [Citace: 12. 1 2020.] Dostupné z: https://randomnerdtutorials.com/esp32-adc-analog-read-arduino-ide/.

[65] VAROL, TOLGA. [Online] 1. 8 2019. [Citace: 11. 1 2021.] Dostupné z: https://www.diva-portal.org/smash/get/diva2:1331704/FULLTEXT01.pdf.

[66] Wang, Xianbin. *Towards Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions*. ResearchGate. [Online] 11 2018. [Citace: 21. 12 2021.] Dostupné z: https://www.researchgate.net/figure/Classification-of-existing-machine-learning-techniques_fig5_326928417.

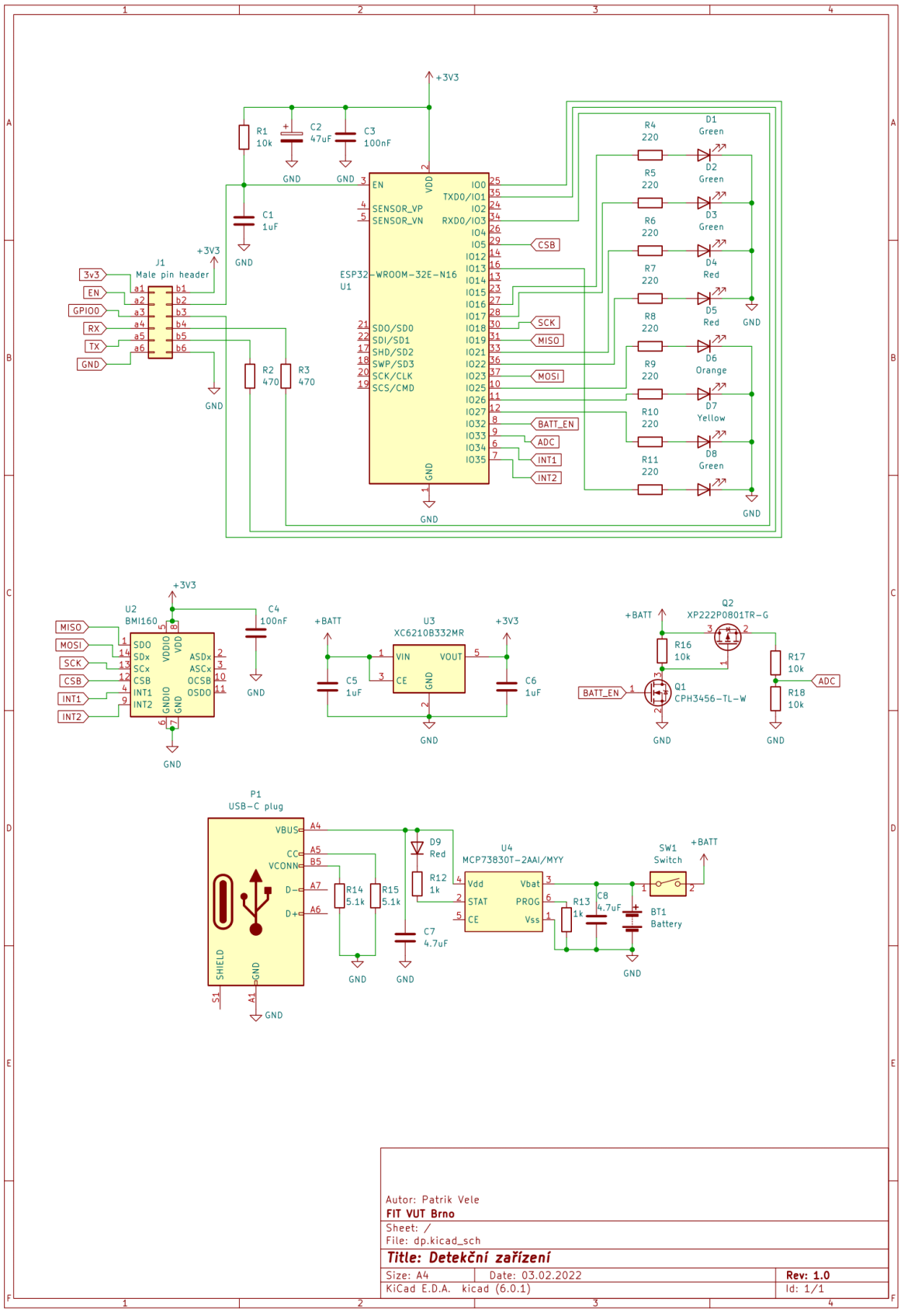
Přílohy

A Seznam součástek

Název	Označení	Popis	Počet kusů
ESP32-WROOM-32E (4MB)	U1	ESP32	1
INR 18650-25R SAMSUNG SDI	BT1	Li-ion baterie 18650	1
BMI160	U2	Inerciální měřící jednotka	1
MCP73830T-2AAI/MYY	U4	Nabíječka	1
XC6210B332MR	U3	LDO regulátor	1
KP-2012CGCK	D1, D2, D3, D8	LED zelená	4
KP-2012SRC-PRV	D4, D5, D9	LED červená	3
KP-2012SYCK	D7	LED žlutá	1
KPT-2012LVSECK-J4-PRV	D6	LED oranžová	1
TLJR476M006R3200	C2	C – Tantal 47 uF, 0805	1
CPH3456-TL-W	Q1	MOSFET – N	1
XP222P0801TR-G	Q2	MOSFET – P	1
PR20206HBNN	J1	Pravoúhlý konektor – 6 pinů	1
1044	-	Držák na baterii	1
USB4125-GF-A	P1	USB-C konektor	1
2 pinový konektor – samice	-	Připojení BT1 a SW1, rozteč 2,54 mm	2
2 pinový konektor – samec	-	Připojení BT1 a SW1	2
Kolébkový spínač	SW1	Zapínací tlačítko	1
Keramický kondenzátor 100 nF	C3, C4	Pouzdro 0805	2
Keramický kondenzátor 1 uF	C1, C5, C6	Pouzdro 0805	3
Keramický kondenzátor 4,7 uF	C7, C8	Pouzdro 0805	2
Rezistor 10 kΩ	R1, R16, R17, R18	Pouzdro 0805	4
Rezistor 5,1 kΩ	R14, R15	Pouzdro 0805	2
Rezistor 1 kΩ	R12, R13	Pouzdro 0805	2
Rezistor 470 Ω	R2, R3	Pouzdro 0805	2
Rezistor 220 Ω	R4 – R11	Pouzdro 0805	8

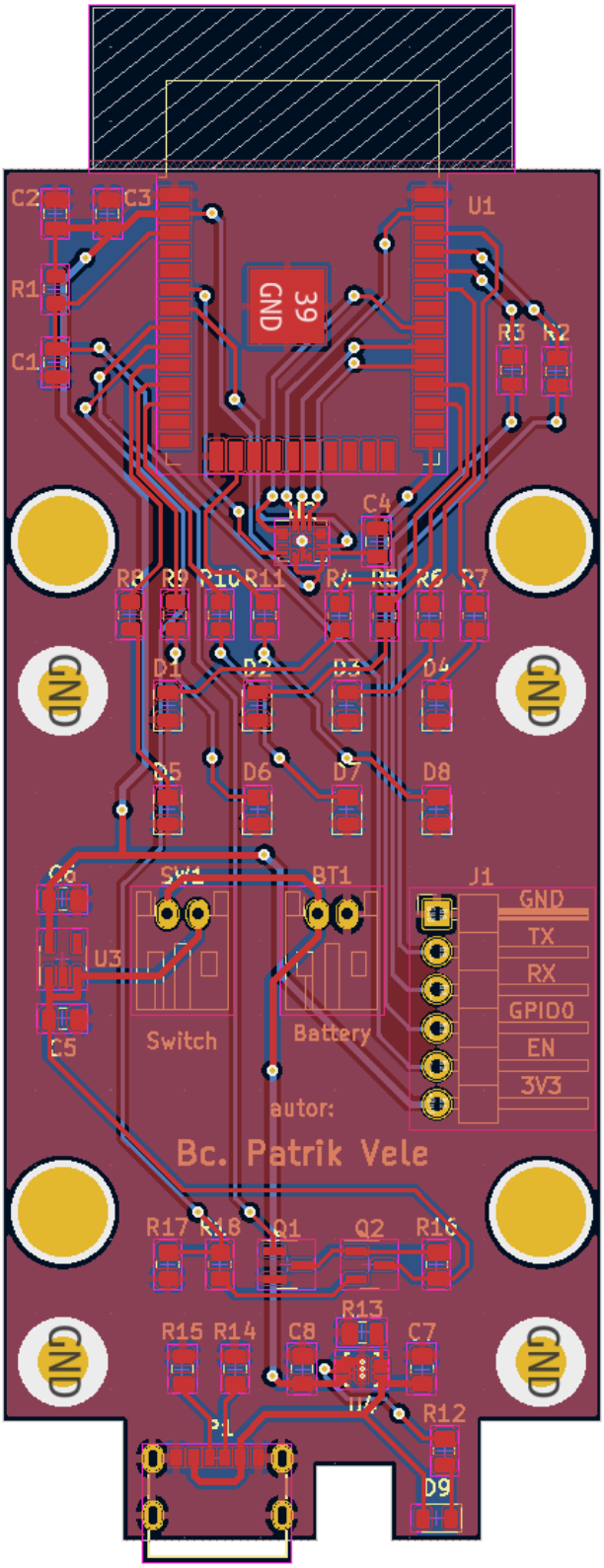
Tabulka A.1: Seznam součástek

B Schéma zapojení



Obrázek B.1: Schéma zapojení

C Deska plošných spojů



Obrázek C.1: Deska plošných spojů

D Obsah přiloženého CD

CD

- 3D - Zdrojové soubory pro 3D tisk.
- Dataset - Vytvořená datová sada.
- Dokumentace - Zdrojové soubory této práce.
- Modely - Vytvořené modely konvoluční neuronové sítě.
- PCB - Zdrojové soubory pro desku plošných spojů.
- Zdrojove kody - Zdrojové kódy.
- DP.pdf - Text této práce ve formátu PDF.