



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

PREDIKTIVNÍ ÚDRŽBA S BEZDRÁTOVÝM PŘENOSEM DAT

PREDICTIVE MAINTENANCE WITH WIRELESS DATA TRANSFER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Michal Pernica

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Barnabás Dobossy

BRNO 2023

Zadání diplomové práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	Bc. Michal Pernica
Studijní program:	Mechatronika
Studijní obor:	bez specializace
Vedoucí práce:	Ing. Barnabás Dobossy
Akademický rok:	2022/23

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Prediktivní údržba s bezdrátovým přenosem dat

Stručná charakteristika problematiky úkolu:

V posledních letech se v průmyslových aplikacích stále častěji objevuje otázka využití prediktivní údržby k zefektivnění výroby a minimalizaci neplánovaných odstávek. Jedním z problémů stávajících zařízení je, že nemusí být k dispozici infrastruktura pro zavedení nástrojů pro sběr, vyhodnocování a přenos dat. V takových případech by mohla pomoci technologie bezdrátového přenosu dat. Tato práce se bude zabývat algoritmy pro detekci a izolaci poruch mechatronických soustav. Práce bude zahrnovat ověření vhodnosti bezdrátového komunikačního rozhraní NB-IoT pro účely prediktivní údržby.

Cíle diplomové práce:

1. Provedte rešerši algoritmů pro detekci a izolaci poruch a možný přenos dat pomocí bezdrátových technologií.
2. Vyberte alespoň dva výukové modely z Mechatronické laboratoře, na kterých bude možné vyvodit různé poruchové stavy, a použijte je k testování algoritmů pro detekci a izolaci poruch.
3. Na vybraných zařízeních implementujte mikrokontrolér a senzory, pomocí kterých budete schopni diagnostikovat poruchové stavy.
4. Vytvořte webové uživatelské rozhraní komunikující s mikrokontrolérem pomocí rozhraní NB-IoT.

Seznam doporučené literatury:

VALÁŠEK, Michael. Mechatronika. Vydavatelství ČVUT, 1995. ISBN 80-01-01276-X.

ISERMANN, Rolf. Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance. 1. Springer-Verlag Berlin Heidelberg, 2006. ISBN 978-3-540-24112-6.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2022/23

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jiří Hlinka, Ph.D.
děkan fakulty

Abstrakt

Tato práce se zabývá prediktivní údržbou a využitím sítě NB-IoT pro její účely. V první části práce jsou uvedené bezdrátové technologie pro přenos dat, následně jsou popsány různé druhy komunikačních protokolů a nakonec popsán postup vývoje prediktivní údržby. Ve druhé části práce je představena serverová aplikace pro účely prediktivní údržby, je zde představeno řízení soustav pomocí mikrokontroléru, NB-IoT modul SIM700E spolu s jeho specifikacemi, připojením k NB-IoT síti a implementací MQTT protokolu. Dále je popsán software serveru skládající se z MQTT brokera, Node-Red, Influx DB a Grafany. Následně je popsána aplikace prediktivní údržby na dvou zařízeních, konkrétně tepelná komora a pneustand. U těchto soustav je popsána jejich funkce, dále pak provedené úpravy. Poté jsou zmíněny poruchové stavy, které chceme detekovat a následně jsou natrénovány klasifikátory pro určení stavu. Nakonec je předěláno řízení soustav, tak aby to bylo možné pomocí mikrokontroléru. Nakonec jsou provedeny experimenty při provozu zařízení a ověření funkčnosti prediktivní údržby k identifikaci stavu.

Summary

This thesis deals with predictive maintenance and the use of NB-IoT network for its purpose. In the first part of the thesis, the wireless technologies for data transmission are presented, followed by a description of the different types of communication protocols, and lastly, the process of developing predictive maintenance is described.

In the second part of the thesis, a server application for predictive maintenance purposes is presented, system control using a microcontroller, the NB-IoT module SIM700E is introduced along with its specifications, connection to the NB-IoT network and implementation of the MQTT protocol. Also, the server software composed of the MQTT broker, Node-Red, Influx DB and Grafana is described.

The third section describes the application of predictive maintenance on two devices, namely a thermal chamber and a pneustand. The function of these systems is described, followed by the modifications applied. Afterwards, the fault conditions we want to detect are mentioned and then classifiers are trained to determine the condition. Finally, the control of the systems is redesigned so that it can be done using a microcontroller. Finally, experiments are performed while operating the device and verifying the functionality of predictive maintenance to identify the condition.

Klíčová slova

Prediktivní údržba, NB-IoT, MQTT, Grafana, mikrokontrolér, PIC32, Server, Raspberry Pi Pneustand, Tepelná komora

Keywords

Predictive Maintenance, NB-IoT, MQTT, Grafana, Microcontroller, PIC32, Server, Raspberry Pi Pneustand, Thermal Chamber

Bibliografická Citace

PERNICA, M. *Prediktivní údržba s bezdrátovým přenosem dat*. Brno, 2023. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/149617>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce: Ing. Barnabás Dobossy.

Prohlašuji, že jsem diplomovou práci na téma Prediktivní údržba s bezdrátovým přenosem dat vypracoval samostatně pod vedením Ing. Barnabáse Dobosyho a uvedl v ní všechny použité literární a jiné odborné zdroje.

Michal Pernica

Brno

.

Rád bych tímto poděkoval Ing. Barnabáši Dobossymu za věnovaný čas, vstřícný přístup a cenné připomínky, které mi v průběhu psaní diplomové práce poskytl. Dále bych chtěl poděkovat mé rodině a kamarádům, kteří mě podporovali po celou dobu studia.

Michal Pernica

Obsah

1	Úvod	9
2	Rešerše	11
2.1	Bezdrátový přenos dat	11
2.1.1	LoRaWan	11
2.1.2	ZigBee	12
2.1.3	LTE-M	13
2.1.4	NB-IoT	14
2.1.5	GSM	14
2.1.6	4G síť	15
2.1.7	Porovnání přenosů	15
2.2	Protokoly	16
2.2.1	TCP	16
2.2.2	UDP	16
2.2.3	HTTP	17
2.2.4	FTP	18
2.2.5	MQTT	18
2.2.6	Zhodnocení	19
2.3	Prediktivní údržba	19
2.3.1	Sběr dat	21
2.3.2	Předzpracování dat	21
2.3.3	Vytvoření atributů dat	21
2.3.4	Určení klasifikačních atributů	22
2.3.5	Trénování modelu	23
2.3.6	Nasazení v provozu	24
3	Serverová aplikace	25
3.1	Mikrokontrolér	25
3.2	Modul SIM7000E	26
3.2.1	Specifikace modulu	27
3.2.2	Připojení k NB-IoT síti	27
3.2.3	Protokol MQTT	28
3.3	Server	28
3.3.1	MQTT Broker - Mosquitto	29
3.3.2	Databáze InfluxDB	30
3.3.3	Node-Red	31
3.3.4	Grafana	32

4	Prediktivní údržba	34
4.1	Tepelná komora	34
4.1.1	Popis soustavy	34
4.1.2	Úprava komory	35
4.1.3	Poruchové stavy	36
4.1.4	Realizace algoritmů pro prediktivní údržbu - MATLAB	37
4.1.5	Implementace řízení a prediktivní údržby na mikrokontroléru	41
4.1.6	Experiment - ověření funkčnosti	43
4.2	Pneustand	44
4.2.1	Popis soustavy	45
4.2.2	Úprava soustavy	47
4.2.3	Poruchové stavy	48
4.2.4	Realizace algoritmů pro prediktivní údržbu - MATLAB	49
4.2.5	Implementace řízení a prediktivní údržby na mikrokontrolér	53
4.2.6	Experiment - ověření funkčnosti	55
5	Závěr	57
	Literatura	59
	Seznam zkratk	64
	Seznam obrázků	65
	Seznam tabulek	67
	Seznam příloh	68

1 Úvod

Prediktivní údržba je součástí čtvrté průmyslové revoluce, která klade důraz na digitalizaci a propojenost jednotlivých zařízení a strojů. Jejím cílem je za pomoci nejrůznějších senzorů, kterých se z důvodu větší komplexnosti instaluje stále více, snažit se o předpovězení poruch ještě před tím, než k samotné poruše dojde. K tomu využívá zpracování a analýzu dat získaných z již zmiňovaných senzorů. Díky včasnému předpovězení budoucí poruchy je možné snížit náklady na opravy, zvýšit spolehlivost stroje a maximalizovat jeho životnost.

Jelikož se zařízení instalují na různých místech a často bývají bez obsluhy či dostupné infrastruktury pro snadnou kontrolu, je dobré mít o jejich stavu alespoň nějaký přehled, aby např. došlo k minimalizaci neplánovaných odstávek. K tomu by měla právě sloužit prediktivní údržba spolu s bezdrátovým přenosem dat. Včasnou identifikací blížící se poruchy a využití vzdáleného monitoringu zařízení je možné se o tomto stavu snadno dozvědět a včas naplánovat kontrolu zařízení či případný servisní úkon, což může výrazně snížit šance pro nefunkčnost zařízení.

Šetření nákladů pomocí prediktivní údržby dokazuje např. článek od společnosti *Deloitte* [1], ve kterém se na základě zkušeností s prediktivní údržbou v projektech potvrzuje, že je pomocí prediktivní údržby možné zkrátit čas věnovaný plánování údržby o 20 – 50 %, zvýšit efektivní využití strojů a zařízení o 10 – 20 % a nakonec snížit celkové náklady na údržbu o 5 – 10 %. Právě kvůli těmto pozitivním dopadům prediktivní údržby jsou nadále prozkoumávány možnosti jejího využití v praxi.

Shrnutí cílů práce

Práce se bude věnovat především ověření funkčnosti algoritmů prediktivní údržby. Testování proběhne na dvou výukových soustavách s ověřením funkčnosti prediktivní údržby při reálném použití. Dále bude cílem práce doplnit prediktivní údržbu o vzdálený monitoring stavu pomocí bezdrátové technologie přenosu NB-IoT a ověřením její vhodnosti pro reálné použití.

Struktura práce

V rešeršní části v kap. 2 budou představeny dostupné bezdrátové technologie, které je možné použít pro vzdálený monitoring, dále pak popsány komunikační protokoly pro přenos dat přes internet a posledním bodem je postup vývoje a implementace prediktivní údržby na zařízeních.

1 ÚVOD

Praktická část práce se zabývá především algoritmem prediktivní údržby v praxi a přenosem dat pomocí NB-IoT. Úkolem je vybrat experimentální zařízení, navrhnout možné poruchové stavy, v případě potřeby upravit zařízení, a následně celý algoritmus implementovat na mikrokontrolér s následným bezdrátovým odesláním dat a zobrazením v aplikaci. V Mechatronické laboratoři bylo několik dostupných modelů, pro tuto práci byly vybrány tyto dva:

- **Tepelná komora**
- **Pneustand**

Nejprve bude v kap. 3 popsána serverová aplikace včetně úvodu k řízení pomocí mikrokontroléru. Následně budou v kap. 4 představeny jednotlivé výukové modely včetně úprav, které byly potřebné pro účely této práce. Dále pak bude popsána samotná implementace algoritmů prediktivní údržby a následně ověření na reálném zařízení.

2 Rešerše

2.1 Bezdrátový přenos dat

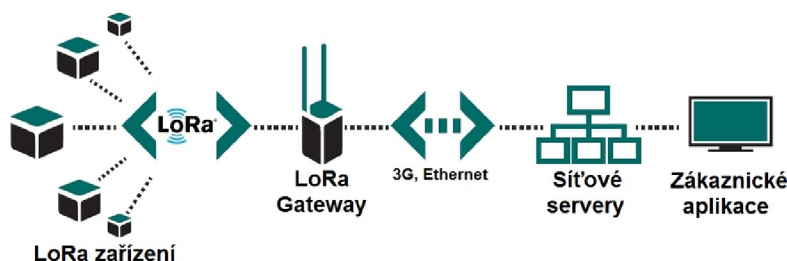
S Průmyslem 4.0 a s ním spojeným Internetem of Things (IoT) vzrostl zájem o sběr dat z nejrůznějších zařízení a stanic. Odesílání dat může probíhat různými způsoby, nejpraktičtější ale pro většinu použití je bezdrátový přenos dat. Ten můžeme rozdělit na 2 kategorie a to na technologie s krátkým a dlouhým dosahem. S IoT také vznikl požadavek na snížení spotřeby při přenosech dat, proto vznikly nové technologie přenosu spadající pod Low-Power Wide-Area Network (LPWAN). V následujících kapitolách budou stručně popsány vybrané technologie a doplněné o výčet hlavních vlastností.

2.1.1 LoRaWan

LoRaWAN (Long Range Wide Area Network) je LPWAN protokol sloužící pro přenos dat na velké vzdálenosti s nízkou spotřebou energie. Je postaven na bezdrátové technologii LoRa (Long Range), která přenáší data pomocí rádiových vln. K tomu využívá modulaci s rozprostřeným spektrem, konkrétně Chirp Spread Spectrum (CSS). Která funguje na principu generování Chirp signálu, který lineárně zvyšuje svoji frekvenci, od spodního do horního pásma (chirp-up), či naopak (chirp-down). Zpráva je následně vložena do signálu pomocí skoků mezi frekvencemi. [2, 3]

Topologii sítě LoRaWan (obr. 2.1) [4] můžeme rozdělit na 4 části:

- **End-node** - Zařízení nebo senzor odesílající data pomocí LoRa technologie.
- **Gateway** - Zařízení, které přijímá LoRa signál z okolních End-Nodů a dokáže ho dekódovat. Je připojené k internetu pro předání dat síťovému serveru.
- **Síťový server** - Zpracovává a dešifruje data z gatewayů a filtruje přijaté zprávy z vícero gatewayů, aby nebyly duplicitní.
- **Aplikační server** - Server pro práci s daty z odesílajících zařízení. Webová stránka nebo aplikace.



Obrázek 2.1: Topologie sítě LoRaWan [5]

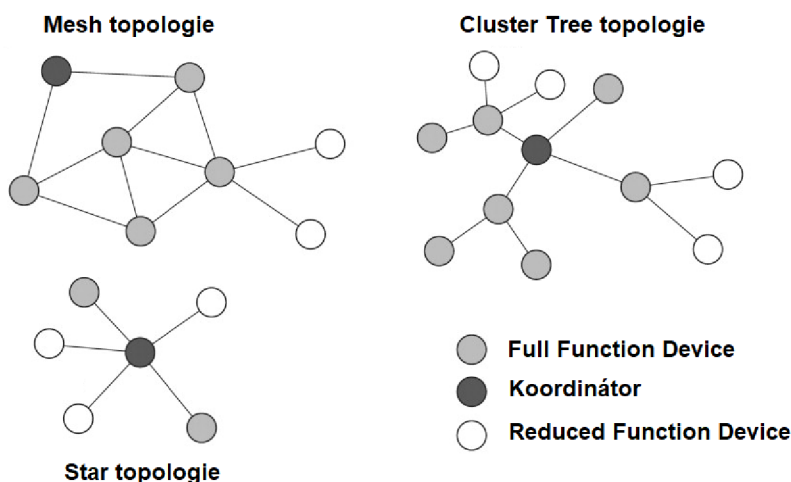
Hlavní vlastnosti LoRaWAN [5, 2]:

- Dlouhý dosah - ve městech kolem 3 km, mimo město až do 15 km.
- Nízká spotřeba energie - díky nízké spotřebě energie v režimu spánku.
- Velký počet připojených zařízení - schopnost obsluhovat velký počet zařízení, což umožňuje masivní nasazení IoT.
- Bezpečnost: různé mechanismy pro zabezpečení přenosu dat, včetně šifrování a autentizace.
- Jednoduchá infrastruktura: vyžaduje pouze několik brán, které propojují zařízení s internetem nebo lze použít i komerční řešení jako např. The Things Network.
- Rychlost - v rozmezí od 0,3 do 50 Kbit/s.

2.1.2 ZigBee

ZigBee [6] je bezdrátový komunikační protokol WPAN (Wireless Personal Area Network) vhodný pro domácí a průmyslovou automatizaci, nebo sběr dat ze senzorů v rámci IoT. Může být provozována v různých topologiích jako je topologie star, tree nebo mesh. S těmito možnostmi je možné vytvořit téměř libolné uspořádání sítě. Zařízení v síti se ještě dále dělí podle typu funkce následovně:

- **Koncové zařízení** - Nesměřují přenosy, mohou být provozována ve sleep módu, komunikují pouze přes směrovač nebo koordinátor, zpravidla bližšího zařízení.
- **Směrovač** - zodpovědné za směrování provozu mezi uzly, zodpovědné za příjem a ukládání zpráv určených jejich "dětem". Zodpovědné za povolení vstupu nových uzlů do sítě.
- **Koordinátor sítě** - speciální typ směrovače, který přiděluje adresu jednotlivých zařízení, určuje vhodný kanál a identifikátor PAN sítě.



Obrázek 2.2: Topologie v síti ZigBee [7]

Dále jsou zařízení rozdělena podle implementace ZigBee protokolu:

- Full Functional Device (FFD) - mají implementovaný kompletní protokolový rámec
- Reduced Functional Device (RFD) - pouze nezbytné části protokolu pro omezení hardwarových požadavků, mohou pracovat pouze jako koncová zařízení

Hlavní vlastnosti ZigBee [2, 6, 8]:

- Nízká spotřeba - možnost provozovat koncová zařízení z baterie
- Krátký dosah - dosah přenosu do 70 m
- Několik přenosových rychlostí - 20, 40 a 250 kbit/s
- Provoz v bezlicenčních pásmech - 868, 902-928 MHz a 2,4 GHz
- Možnost probouzet koncová zařízení pro odeslání dat pomocí synchronizačních sekvencí v čase od 15 ms až do 15 minut
- Různé druhy aplikačních profilů (např. ZigBee Building Automation, ZigBee Remote Control, atd.)
- Nutné vybudování vlastní sítě

2.1.3 LTE-M

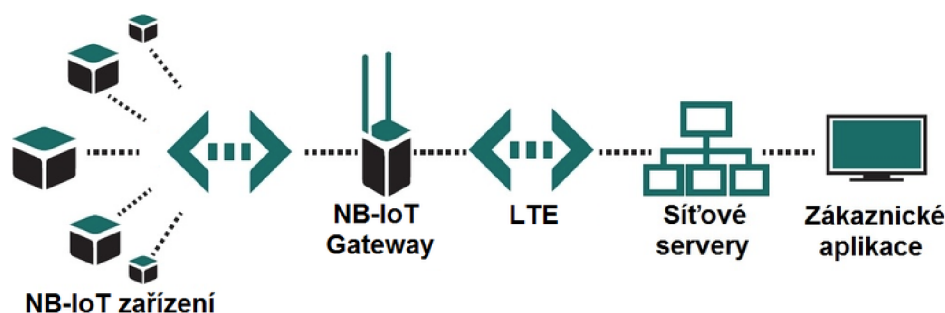
LTE-M (Long Term Evolution Machine Type Communication) je LPWAN technologie vytvořená v projektu 3GPP pro IoT a M2M (Machine To Machine). Je postavena na technologii 4G, aby umožnila připojení širokého spektra zařízení a služeb, a také ji lze díky tomu spustit na komunikační infrastruktuře mobilních operátorů pouhou softwarovou změnou. Díky úspornému režimu a přerušovanému příjmu mohou zařízení využívající tuto technologii pracovat při napájení z baterie po poměrně dlouhou dobu oproti klasické 4G síti. [9, 10]

Hlavní vlastnosti LTE-M [2, 9]:

- Rychlost přenosu - 1 Mbit/s pro verzi 13 a až 4 Mbit/s pro verzi 14
- Hand-Over - Možnost přepínání z jednoho vysílače na druhý - možnost použít pro mobilní zařízení
- Podporuje Voice Over LTE (Vo-LTE)
- Nízká spotřeba
- Malá latence přenosu - 10 - 15 ms
- Placená služba mobilních operátorů
- Roamingový přenos dat většinou bez poplatků navíc

2.1.4 NB-IoT

NB-IoT (Narrowband Internet of Things) je také LPWAN technologie speciálně navržená pro IoT aplikace. Obdobně jako LTE-M ji lze spustit již na stávající infrastruktuře pouhou softwarovou změnou. Využívá totiž standardu LTE, ale je omezena šířkou pásma na jediné úzké pásmo 200 kHz. Topologie sítě je zobrazena na obr. 2.3 [11, 12].



Obrázek 2.3: Topologie sítě NB-IoT [11]

Hlavní vlastnosti NB-IoT [2]:

- Nízká rychlost - 26 Kbit/s pro verzi NB1 a 127 Kbit/s pro NB2
- Nízká spotřeba energie - zajištěno podobně jako v LTE-M (kap. 2.1.3)
- Velké pokrytí a prostupnost - dosah i na zařízení umístěné hluboko v budovách nebo na odlehlých místech
- nepodporuje Hand-Over - není vhodná pro mobilní zařízení
- Škálovatelnost - navržen pro velký počet zařízení v síti, až 100 tisíc na jeden vysílač
- Velká latence přenosu - 1,6-10 s

2.1.5 GSM

GSM (Global System for Mobile Communications) je standard pro mobilní telefonování, avšak v průmyslových aplikacích se také využívá pro různé aplikace. Technologie podporuje řadu protokolů pro přenos dat jako např. GPRS nebo EDGE. Nebo může přenášet data ve formě SMS. Oproti novým technologiím jako jsou 4G nebo 5G sítě, ale dosahuje velmi malého přenosu dat. Proto se hodí spíše pouze pro monitorování systému např. pro sběr výstražných a diagnostických dat. [13]

Hlavní vlastnosti GSM:

- Nízká rychlost - 33 Kbit/s, při využití protokolu EDGE až 473, Kbit/s
- Velké pokrytí a spolehlivost - signál i na odlehlých místech
- Nížší pořizovací náklady modulů

2.1.6 4G síť

Síť 4G je v současné době nejrozšířenější a nejrychlejší síť, která je dostupná v ČR. Pro zajištění vyšších přenosových rychlostí využívá pokročilé anténní technologie, jako je MIMO (Multiple Input Multiple Output). Tato technologie zvyšuje kvalitu signálu, pokrytí a kapacitu tím, základní stanice komunikující s mobilním zařízením je schopna přidělit zařízení více antén k vysílání i příjmu dat. [14]

Hlavní vlastnosti 4G sítí [2, 14]:

- Vysoká rychlost - až 100 Mbit/s pro pohyblivá zařízení, nepohyblivá až 1 Gbit/s
- Malá latence přenosu
- Velké pokrytí
- Vyšší pořizovací cena modulů
- Připojení ke službám v reálném čase
- Malá latence přenosu - 20-30 ms
- Připojení 200-400 zařízení na mobilní věž

2.1.7 Porovnání přenosů

V předchozím kapitolách byly představeny různé druhy technologií, a v následující tabulce 2.1 je znázorněno porovnání podle klíčových vlastností vycházející ze zjištěných informací. Jelikož předpokládáme využití pro průmyslové aplikace, kde nemusí být zajištěn internet přímo ve výrobní hale, případně můžou být zařízení rozmístěná na rozlehlejší ploše. Bylo by vhodné využít některé z technologií které mají velký dosah nebo případně dobré pokrytí, čímž připadají v úvahu především technologie mobilních sítí. Dále víme, že se nebude jednat o velký přenos dat, jelikož budou odesílána pouze data ze senzorů po určitých časových intervalech, čímž by tedy bylo vhodné využít technologie LTE-M nebo NB-IoT, které navíc spadají technologií s malou spotřebou elektrické energie, tudíž je možné je provozovat pouze z baterie po dlouhou dobu.

Tabulka 2.1: Porovnání sítí

Název	Rychlost	Pokrytí ČR	Dosah	Počet připojených zařízení [ks]
LoRaWan	do 50 Kbit/s	kolem 50%	10 km	10 tis.
ZigBee	do 250 Kbit/s	-	70 m	35-100
LTE-M	do 4 Mbit/s	celá ČR	≈ 1 km	100 tis.
NB-IoT	do 127 Kbit/s	celá ČR	≈ 1 km	100 tis.
GSM	do 473 Kbit/s	90% ČR	≈ 35 km	1 tis.
4G	1 Gbit/s	90% ČR	≈ 6.5km	400

2.2 Protokoly

V této podkapitole budou představeny komunikační protokoly pro přenos dat pomocí internetu, a které je možné použít v NB-IoT síti. Budou popsány protokoly TCP, UDP, HTTP, FTP a MQTT. Budou popsány jejich vývody a nevýhody. V posledním bodě bude zdůvodnění výběru konkrétního protokolu, který bude dále použit v praktické části práce.

2.2.1 TCP

TCP (Transmission Control Protocol) je protokol transportní vrstvy. Jedná se o spojově orientovaný přenos a proto musí být před samotným odesláním dat navázáno spojení mezi klientem a serverem, to je nutné potvrdit tzv. trojcestným podáním ruky (Three-way handshake). Před odesláním jsou data rozdělena do paketů a ty jsou následně očíslovány na základě prvotního podání ruky. Očíslováním dat je zajištěna spolehlivost protokolu, jelikož po přijetí dat jsou na základě toho pakety seřazeny a pokud některý z paketů chybí, tak příjemce nepotvrdí příjem daného paketu a zařízení ho odešle znovu. [15]

Hlavní vlastnosti TCP:

- Spolehlivý potvrzovaný protokol;
- široké nastavení protokolu;
- regulace rychlosti sítě zabraňující zahlcení;
- nutné vytvořit vlastní server pro příjem a především zpracování dat;
- 1460 znaků je obvyklá maximální velikost zprávy v ethernetové síti.

2.2.2 UDP

UDP (User Datagram Protocol) je obdobně jako TCP protokolem transportní vrstvy a není spojově orientovaný. Jelikož nezaručuje spolehlivé doručení ani správné pořadí odesílaných paketů označuje se jako nespolehlivý. Jelikož tedy není spojově orientovaný, tak tím pádem nenavazuje spojení se serverem před odesláním dat a pracuje na principu otázka-odpověď (request-response). Je také označován jako nenáročný protokol a proto se

UDP využívá především u aplikací, kde je současně zapotřebí jednoduchost, nízká latence a neklade se důraz na příjem všech dat, což může být např. u online her, VoIP nebo je nasazen na DNS serverech. [16]

Hlavní vlastnosti UDP:

- Vysoká rychlost a nízká latence - nemusí se navazovat spojení a potvrzovat přijetí paketů.
- Možnost ztráty některých dat.
- Neobsahuje řízení toku dat, možné zahlcení.
- Nutné vytvořit vlastní server pro příjem a především zpracování dat.

2.2.3 HTTP

HTTP (Hyper Text Transfer Protocol) je síťový aplikační protokol, který slouží k přenosu dokumentů, především webových stránek. Protokol byl navržen pro komunikaci mezi webovými prohlížeči a webovými servery, ale může být použit také k jiným účelům. HTTP je protokol s modelem typu žádost-odpověď (request-response) v architektuře klient-server. K interakci se používají tzv. metody jako GET, POST a HEAD. Klient posílá požadavek pro získání nebo změnu zdroje na server. Po dobu vyřizování požadavku je udržováno spojení, dokud server neodpoví zprávou se stavem zpracování požadavku, za kterou následují vyžádaná data. Komunikace probíhá nad protokolem TCP a má rezervovaný port 80, ale může být zajištěna i jinými porty jako např. 8080 nebo 8008. [17]

• HTTP metody:

- GET - základní metoda, slouží k získání dokumentu, nejčastěji k HTML souboru;
- POST - slouží k získání dokumentu, který umožňuje nahrání dat na server;
- HEAD - velmi podobná metodě GET, ale získáme pouze informace o dokumentu, nikoliv samotný dokument;
- PUT/DELETE - požadavek na vytvoření/odstranění dokumentu ze serveru (příliš se nepoužívají).

Hlavní vlastnosti HTTP:

- Umožňuje posílání různého druhu dat podle nastavení v hlavičce;
- několik druhů požadavků (GET, POST, ...);
- je bez trvalého připojení;
- možnost zabezpečené komunikace pomocí HTTPS;
- nutné vytvoření webové stránky/serveru pro zpracování dat.

2.2.4 FTP

FTP (File Transfer Protocol) je síťový protokol sloužící především k přenosu souborů mezi klientem a serverem. Je založen na bázi TCP spojení a na modelu klient-server. Obvykle má vyhrazeny TCP porty 20 a 21. Port 21 obstarává navázání spojení a přenos příkazů. Přenos dat poté probíhá na portu 20. Uživatelé se ve většině případů přihlašují za pomoci uživatelského jména a hesla ve formátu prostého textu, ale mohou i anonymně, pokud je tak server nakonfigurován. Umožňuje uživatelům nahrávat, stahovat a spravovat soubory na FTP serverech. Podporuje operace jako výpis adresáře, přejmenování souborů, přesun souborů, kopírování. [18]

Hlavní vlastnosti FTP:

- Vyžaduje spojení se serverem;
- možnost zabezpečeného přenosu pomocí FTPS;
- příkazy pro manipulaci se soubory na serveru;
- 2 módy s různou odpovědností o vytvoření datového spojení - aktivní (server) a pasivní (klient);
- nutné vytvoření FTP serveru.

2.2.5 MQTT

MQTT (Message Queuing Telemetry Transport) je protokol určený k zasílání zpráv pro efektivní komunikaci mezi zařízeními, široce se používá v IoT aplikacích pro výměnu dat v reálném čase, vzdáleného monitorování a ovládání připojených zařízení. Vznikl hlavně kvůli nevhodnosti do té doby používaného protokolu HTTP. Jelikož využíval schéma žádost-odpověď, měl delší záhlaví zprávy a byl náročnější na implementaci. MQTT je tedy velmi jednoduchý a nenáročný protokol fungující na bázi TCP, nejčastěji na portu 1883. Řídí se modelem publish-subscribe, kdy jsou zprávy publikovány pod tzv. tématy (topic). Centrálním bodem je tzv. broker, ten zprávy přijímá, ukládá a třídí podle již zmíněných témat a případně je odesílá klientům, kteří dané téma odebírají. Protokol nabízí tři stupně ověřování QoS (Quality of Service). [19]

- QoS 0 - Maximálně jednou
Funguje podobně jako UDP, pošle zprávu a dál se nestará, jestli byla zpráva ztracena a nebo byla v pořádku doručena.
- QoS 1 - Alespoň jednou
Je zajištěno, že zpráva dorazí, ale je možné, že bude přijata vícekrát než jednou.
- QoS 2 - Přesně jednou
Je zajištěno, že zpráva dorazí, ale nestane se, že by mohla být duplikována.

Důležité je zmínit, že kromě QoS je možné nastavit tzv. retain flag. Ten zajistí, že broker si ukládá poslední zprávu a v případě připojení subscribera mu ji odešle.

Hlavní vlastnosti MQTT:

- Malý overhead přenosu;
- nastavení kvality služby (QoS);
- pomocí brokeru je možné přenést data několika uživatelům,
- umožňuje odběr dat pod různými tématy;
- možnost zabezpečení pomocí SSL;
- nutný MQTT broker.

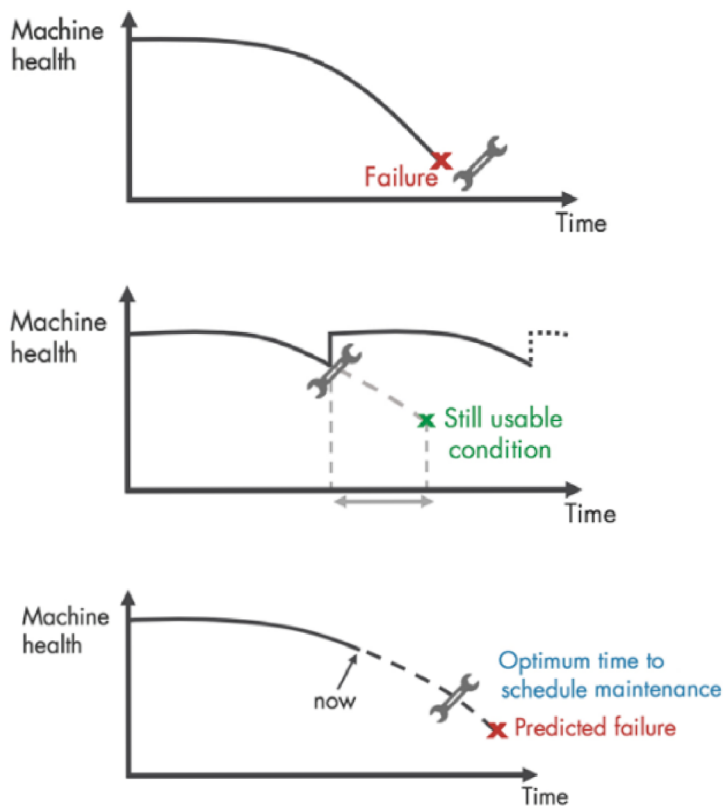
2.2.6 Zhodnocení

Jako nejlepší možný protokol pro přenos dat byl zvolen MQTT, jelikož je již doplněn o mnoho různých funkcionalit jako QoS, retain flag, rozdělení přijímaných zpráv do témat či jejich případné preposlání. Čehož by kombinací protokolů TCP a UDP šlo také dosáhnout, ale musely by být tyto funkcionality přímo naprogramovány v samotném TCP serveru. Proto je vhodnější již použít hotové řešení, které je navíc optimalizováno pro velké množství připojených zařízení.

2.3 Prediktivní údržba

Informace k následující podkapitole jsou čerpané především z: [20, 21]. Prediktivní údržba je technika pro efektivní údržbu zařízení a strojů. Pomocí této techniky se snažíme předvídat poruchu zařízení ještě dříve než k ní dojde. Předpovídání vznikajících poruch totiž má za následek ušetření nemalých finančních prostředků. Samotnou údržbu je možné rozdělit do třech kategorií dle toho, v jakém okamžiku k ní dochází:

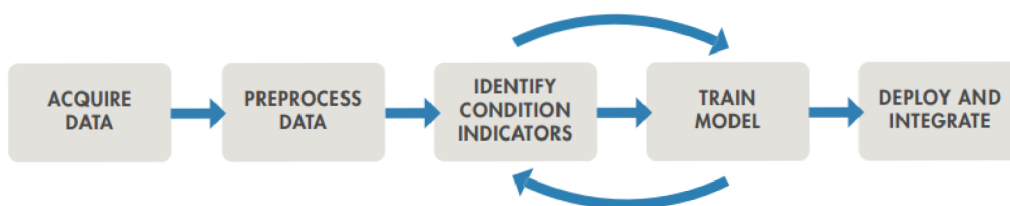
- **Reaktivní údržba** - údržba zařízení nastává až v době kdy došlo k jeho poruše, což u některých zařízení může být kritické.
- **Preventivní údržba** - části zařízení jsou měněny preventivně v předem daných časových intervalech nebo na základě překročení limitní hodnoty některé ze sledovaných veličin. Jelikož předem není známo kdy k poruše dojde, je zapotřebí stanovovat časové intervaly výměn součástí konzervativním způsobem.
- **Prediktivní údržba** - jak již bylo zmíněno jejím cílem je předpovědět možnou poruchu ještě dříve než k ní vůbec dojde, čímž je možné předem objednat nebo vyrobit potřebné díly a naplánovat opravy tak, aby co nejméně ovlivnily celý provoz. Dále se snaží předpovědět i dobu, za kterou k poruše dojde a identifikovat zdroje problému pro případnou optimalizaci zařízení. Díky tomu je možné pomocí ní ušetřit nemalé množství finančních prostředků, které je nutné vynaložit v případě předchozích dvou přístupů k údržbě.



Obrázek 2.4: Srovnání (shora) reaktivní, preventivní a prediktivní údržba [20]

Podstata prediktivní údržby spočívá ve sběru dat ze senzorů v zařízeních. Tyto data jsou následně zpracována a na základě analýzy zpracovaných dat jsou následně vyhodnocena. To bývá obvykle řešeno metodami umělé inteligence.

V následujících kapitolách budou jednotlivé body postupu prediktivní údržby blíže představeny. 2.5



Obrázek 2.5: Pracovní postup při implementaci prediktivní údržby [20]

2.3.1 Sběr dat

Prvním krokem celé implementace je sběr dat. Data mohou být získávány více přístupy, prvním z nich je:

- **Data Based** - Při tomto sběru dat je nutné identifikovat možné poruchové stavy a na základě analýzy těchto stavů určit potřebné senzory, kterými jsme schopni detekovat změny v chování zařízení. Dále při tomto sběru dat může hrozit riziko zničení zařízení.
- **Model Based** - Tento přístup lze využít v případě, že dokážeme zařízení matematicky správně popsat či případně využít metod identifikace parametrů konkrétně metody black box. Následně je možné simulovat zařízení za různých podmínek, např. i v takových, které by byly těžko dosažitelné. Tento přístup tedy může značně urychlit celý vývoj aplikace.

V obou případech ale platí, že je potřebné získat velké množství dat při různých provozních stavech. Přesnou velikost nelze empiricky stavět jelikož je třeba brát v úvahu i složitost problému a samotného klasifikačního algoritmu. Ale pouze s dostatečnou reprezentací klasifikovaných stavů v datech je možné natrénovat klasifikátor s dobrými zobecnujícími vlastnostmi. [22]

2.3.2 Předzpracování dat

Po získání surových dat je na řadě jejich předzpracování, to je důležité hlavně pro zlepšení jejich kvality. V tomto kroku je nutné získaná data upravit, tak aby z nich bylo možné určit tzv. příznaky. Předzpracování zahrnuje:

- převedení na fyzikální jednotky;
- odstranění offsetu, invalidních hodnot, detrendaci;
- filtraci signálu - redukce šumu;
- převedení signálu do frekvenční oblasti.

2.3.3 Vytvoření atributů dat

Po upravení dat přichází na řadu zpracování dat statistickou analýzou a tím jsou vytvořeny samotné atributy dat (features), čímž se snažíme kvantitativně popsat naměřená data, a tím je zobecnit pro následné trénování klasifikátoru. Ty tedy mohou být zpracovány buď na samotném získaném signále (časová závislost), nebo na předzpracovaném signálu převedeném do frekvenční závislosti. Mezi základní atributy dat v časové závislosti patří např. průměr, směrodatná odchylka, maximální hodnota signálu. Ve frekvenční závislosti je to např. výška špičky amplitudy, její frekvence nebo Výkon pásma. Všechny zbylé časové atributy, které je možné získat pomocí nástroje Diagnostic Feature Designer (DFD) dostupném v softwaru MATLAB, který byl použit pro jejich výpočet v praktické části, jsou znázorněny v tabulce 2.2.

Tabulka 2.2: Atributy dat pro časovou závislost dostupné v programu DFD

Feature	Atribut
RMS	Kvadratický průměr
Shape Factor	Koeficient tvaru signálu
Kurtosis	Koeficient špičatosti
Skewness	Koeficient šikmosti
Crest Factor	Činitel výkyvu
Impulse Factor	Činitel impulzu
Clearance Factor	Činitel vůle
Signal-to-Noise Ratio	Poměr signálu a šumu
Total Harmonic Distorsion	Zkreslení sinusového signálu
Signal to Noise and Distorsion Ratio	Poměr signálu k šumu a zkreslení

2.3.4 Určení klasifikačních atributů

Jakmile jsou atributy vypočítané, je zapotřebí je ohodnotit a seřadit podle toho, které atributy mají vypovídající hodnotu ke měnícímu se stavu soustavy. Protože dobrý výběr co možná nejmenšího počtu atributů má velký vliv na následný klasifikační proces. Jelikož z počtem atributů je klasifikátor větší, složitější, méně obecný a náročnější na výpočet. K výběru atributů je možné použít jednu ze 3 metod:

- **Filter**

Filter metody [23] jsou nezávislé na výběru modelu klasifikátoru. Pracují na principu ohodnocení jednotlivých atributů pomocí funkce a následném seřazení atributů od nejlepšího po nejhorší. Vhodné atributy jsou poté vybrány z prvních pozic v seznamu. Tato metoda je méně náročná oproti těm následujícím.

- **Wrapper**

Tato metoda [24] je založena na výběru atributů pro konkrétní model klasifikátoru. Prohledává stavový prostor možných atributů a snaží se najít optimální kombinaci atributů podle zadaného hodnotícího kritéria pro daný model.

- **Embedded**

Metoda Embedded [25] kombinuje vlastnosti již zmíněných metod. Vyznačuje se implementací algoritmu výběru atributů rovnou do klasifikátoru. Metoda natrénuje model, pro všechny atributy a následně porovnává vliv jednotlivých atributů na celkovou přesnost modelu. Příkladem této metody jsou Decision trees.

V praktické části práce bude využita metoda Filter, konkrétně její zástupce ReliefF, jeho použití vychází z článku [26], který se zabývá porovnáním metod pro volbu příznaků a na základě výsledků doporučuje používání této metody ohodnocení. Algoritmus ReliefF

je rozšířením Relief. Základní myšlenkou algoritmu je, že náhodně vybere příznak s určitou hodnotou a následně vyhledá nejbližší prvek stejné a opačné třídy. Následně vypočítá vzdálenost mezi nimi. Normalizované vzdálenosti stejné třídy jsou poté odečteny od váhy a naopak opačné třídy přičteny. Výsledná váha atributu je poté v intervalu od $\langle -1, 1 \rangle$. Čím vhodnější atribut je, tím jsou body stejné třídy blíže u sebe a naopak od opačných bodů dále. A váha se bude tím pádem bude přibližovat k 1. Rozšíření Relief přináší využití tohoto algoritmu pro data s více než dvěma třídami. [27]

2.3.5 Trénování modelu

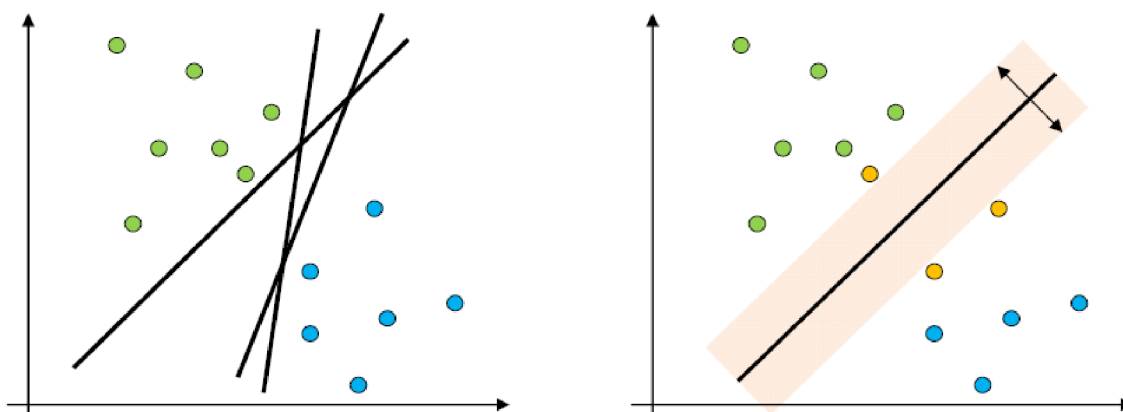
Po ohodnocení atributů je úkolem vytvořit hodnotící model pomocí metod strojového učení, který může mít různou funkcionalitu jako:

- detekci anomálií
- detekci různých typů závad pomocí klasifikace
- předpověď přechodu od zdravého stavu k poruše

Následně je možné určit, zda-li mají vybrané atributy rozlišovací schopnost k určení správného stavu. Toho je docíleno ověřením na validačních datech, které byly získány spolu s daty trénovacími, případně je to ověřeno až při zprovoznění na reálném systému. Pokud ale schopnost určit správně stav nemají, je zapotřebí přehodnotit správnost určených atributů a případně je změnit či zvolit jiný přístup.

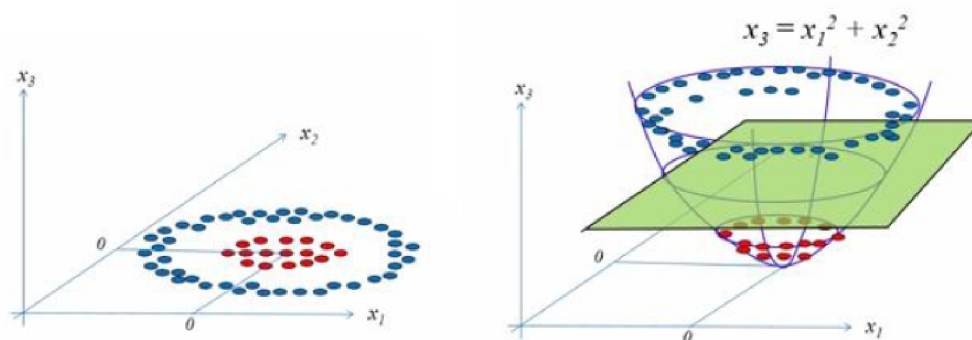
Trénování modelu může být realizováno již hotovým softwarových řešení jako je např. Classification learner app v programu MATLAB. Jedná se o grafické rozhraní pro vytváření a vyhodnocování klasifikačních algoritmů různých druhů (Např. diskriminační analýzy, rozhodovací stromy, algoritmy podpůrných vektorů nebo neuronových sítí). Výhodou této aplikace je právě rychlé natrénování mnoha různých algoritmů, výběr atributů pro jejich trénování, nastavení jednotlivých algoritmů, vytvoření Scatteru, což je graf rozptylu dat i s určením zda byl stav správně či špatně predikován, Confusion matrix, která vizualizuje klasifikační výsledky v tabulce pro každou třídu, či otestování na validačních datech. Nabízí také export vybraného modelu do prostředí MATLAB ve formě funkce, což nastává až ve chvíli, kdy vyhodnotíme, že daný model má dostatečnou přesnost pro použití ve vlastních skriptech nebo pro reálnou aplikaci. Pokud ale model nemá požadovanou přesnost musíme se opět vrátit k výběru atributů, případně nasbírat data z dalších senzorů, která budou lépe kvantifikovat stavy.

Z důvodu použití algoritmů podpůrných systémů neboli Support Vector Machines (SVM) [28, 29] v praktické části práce, z důvodu nejlepších výsledků v klasifikaci, zde budou ještě krátce popsány. Cílem této metody je rozdělení prostoru pomocí linií/rozhodovacích hranic tak, aby dokázaly rozdělit n -rozměrný prostor do tříd. Podle kterých můžeme následně nově vložené body do prostoru správně identifikovat. Pro lineárně separovatelná data ve 2D prostoru (obr. 2.6) je optimální hranicí přímka. Přímka by měla umístěna mezi body tak, aby byla maximalizována vzdálenost mezi nejbližšími body, které se nazývají podpůrnými vektory.



Obrázek 2.6: Dělicí přímka zjištěná pomocí SVM pro 2D prostor [28]

Pokud data nejsou lineárně separovatelná (obr. 2.7) využije se tzv. kernel trick, který transformuje data pomocí funkce, která je nelineární, do prostoru s vyšší dimenzí. Ve vyšší dimenzi jsou již následně data lineárně separovatelná. Kernelové funkce jsou např. polynomy, funkce s radiální bází.



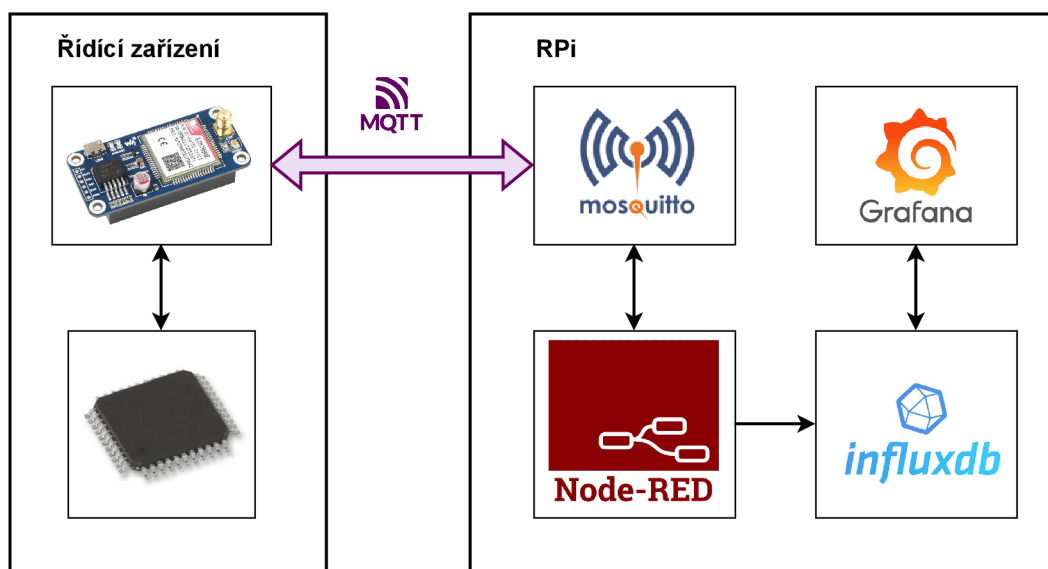
Obrázek 2.7: Transformace dat pomocí kernelovské funkce [28]

2.3.6 Nasazení v provozu

Poslední částí vývoje prediktivní údržby je její aplikace v reálném provozu. Nasazení může být realizované přímo na zařízení nebo vzdáleně odesláním dat na server. Po integraci na reálné zařízení je nutné algoritmus testovat, zda-li se chová tak jak má, případně přistoupit k upravení modelu klasifikátoru nebo začít s celým procesem znovu, po odhalení případných nedostatků.

3 Serverová aplikace

V rámci rešeršní části práce v podkapitolách 2.1 a 2.2 bylo rozhodnuto o využití NB-IoT sítě a protokolu MQTT k přenosu dat na server. Odůvodnění využití těchto dvou technologií je uvedeno v kap. 2.2.6. V této podkapitole bude popsán použitý mikrokontrolér, konkrétní modul pro přenos dat a vytvoření celého serverového systému. Serverovou část aplikace bude obstarávat několik druhů softwaru jako např. Node-Red, Grafana a další. Použitý software bude přiblížen v následujících kapitolách. Schéma struktury celé aplikace je na obr. 3.1.

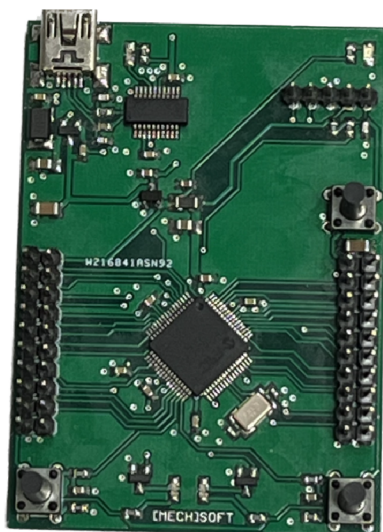


Obrázek 3.1: Schéma serverové aplikace

3.1 Mikrokontrolér

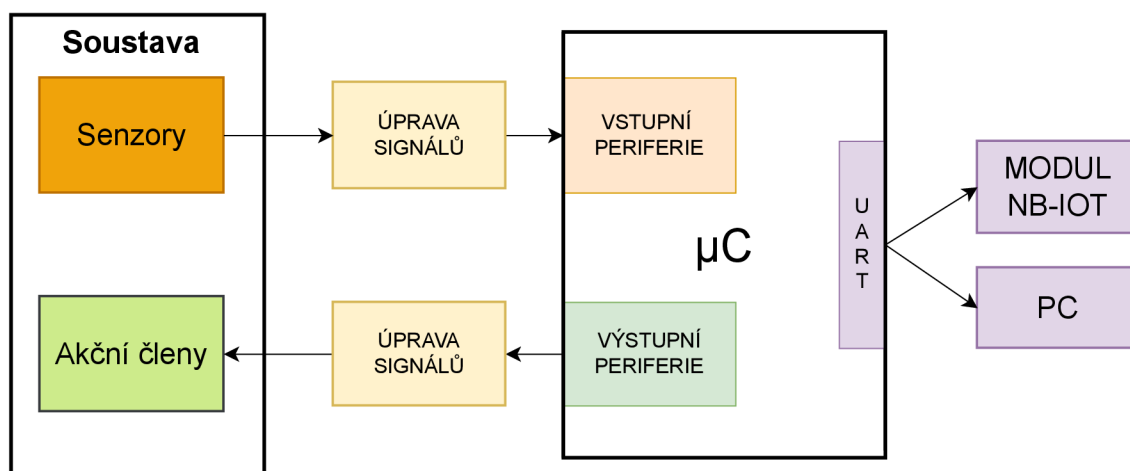
Jako mikrokontrolér určený k řízení soustav a komunikaci s modulem byl vybrán PIC32 od společnosti *Microchip*, a to konkrétně PIC32MK1024MCF064 [30]. Tento mikrokontrolér byl v laboratoři dostupný na vývojové platformě (obr. 3.2) a zároveň obsahoval potřebné druhy periférií a jejich počet, jako např. 5x AD převodník a 2x UART. Obsahuje dvě 20-pinové lišty pro připojení k pinům čipu a dvě vestavěná tlačítka.

Většina průmyslových zařízení bývá řízena pomocí řídicích karet, např. karty od National Instruments nebo PLC a ty jsou doplněny průmyslovými senzory a aktuátory, které pracují obvykle s napětím 24 V. Jelikož zařízení bude řízeno pomocí mikrokontroléru, který má maximální dovolené napětí 3,3 V, tak je potřeba výstupní napětí ze senzorů



Obrázek 3.2: Vývojová platforma (MECH)SOFT

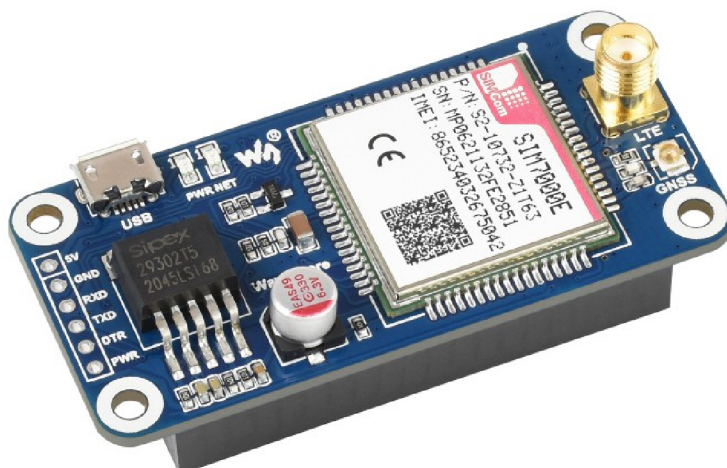
upravit, aby je bylo možné k mikrokontroléru připojit. Na výstupních perifériích je naopak nutné zajistit spínání vyššího napětí, např. tranzistory. Obecné schéma zapojení mikrokontroléru pro řízení vybraných soustav a odesílání dat pomocí modulu je znázorněno na obr. 3.3.



Obrázek 3.3: Obecné schéma zapojení mikrokontroléru

3.2 Modul SIM7000E

Pro připojení k NB-IoT síti byla vybrána již hotová vývojová platforma s čipem *SIM7000E* (obr. 3.4) od společnosti *Waveshare*. [31] Byla vybrána z toho důvodu, že je možné ji rychle integrovat do systému a také kvůli tomu, že čip disponuje potřebnými technologiemi pro naše účely. Vývojová platforma disponuje konektory pro připojení antén, micro-USB konektorem pro snadné připojení a komunikaci s počítačem a dále také 40-pinový konektor pro připojení Raspberry Pi.



Obrázek 3.4: Vývojová platforma s modulem SIMCom 7000E [31]

3.2.1 Specifikace modulu

SIM7000E vyráběný společností *SIMCom* je modul podporující bezdrátový přenos dat pomocí *NB-IoT/Cat-M/EDGE/GPRS*. Při využívání sítě *NB-IoT* může zajišťovat datový přenos s rychlostí stahování až 36kb/s a nahrávání až 66 kb/s. Pro přenos dat je možné využít protokoly TCP, UDP, HTTP, FTP, MQTT, SMS, atd. Dále je možné ho využít pro určování polohy díky *GPS/BeiDou/GLONASS/Galileo*. Komunikace s modulem z počítače či mikrokontroléru je zajišťována sběrnici UART, která může pracovat v rozsahu 300 bit/s až 3,68 Mbit/s, po té je možné ho ovládat pomocí AT příkazů. [31, 32]

3.2.2 Připojení k NB-IoT síti

Jelikož modul dokáže pracovat i s jinými druhy sítí je potřeba ho pomocí AT příkazů správně nastavit, aby se připojoval k *NB-IoT* síti. V tomto případě konkrétně k síti od společnosti *Vodafone*. Příkazy ke konfiguraci a připojení k internetu jsou uvedeny níže. [33, 34]

Pomocí prvních dvou příkazů modul vždy vypíše aktuální nastavení a poté všechny možnosti nastavení, třetím příkazem už se modul nastavuje. Ve třetí sekvenci příkazů je ukázka nastavení všech kanálů, je možné nastavit, ale jen některé. V ČR společnost *Vodafone* používá kanál č. 20.

AT+CNMP?	% zjisteni nastaveni
AT+CNMP=?	% moznosti nastaveni
AT+CNMP=38	% nastaveni LTE only
AT+CMNB?	% zjisteni nastaveni
AT+CMNB=?	% moznosti nastaveni
AT+CMNB=2	% nastaveni NB-IoT

```

AT+CBANDCFG?                               % zjisteni nastaveni
AT+CBANDCFG=?                               % moznosti nastaveni
AT+CBANDCFG="NB-IOT",3,5,8,20,28           % nastaveni kanalu pro NB-IoT

AT+COPS?                                    % zjisteni nazvu pripojene site
// +COPS: 0,0,"Vodafone DATA ONLY",9     % odpoved modulu - 9 = NB-IoT

```

3.2.3 Protokol MQTT

Protokol MQTT na modulu disponuje malým overheadem přenosu, nastavení QoS a retain flag, možnost šifrování pomocí SSL a nabízí i odběr MQTT topiců. Velikost odesílaných dat je ale limitována 512 znaky. Aplikační kód pro použití je zobrazen níže. [35]

```

AT+CNECT=1,"lpwa.vodafone.com"             % pripojeni k internetu

AT+SMCONF="URL","broker.hivemq.com","1883" % nastaveni MQTT
AT+SMCONF="KEEPTIME",60
AT+SMCONF="CLIENTID","clientID"
AT+SMCONF="USERNAME","uzivatelske jmeno"
AT+SMCONF="PASSWORD","heslo"
AT+SMCONN                                  % pripojeni klienta

%AT+SMPUB=<topic>,<contentlength>,<qos>,<retain> % odeslani zpravy
AT+SMPUB="test_nbiot","10",0,0

%AT+SMSUB=<topic>,<qos>                     % prijem zprav
AT+SMUNSUB="test_nbiot"

AT+SMDISC                                  % odpojeni klienta

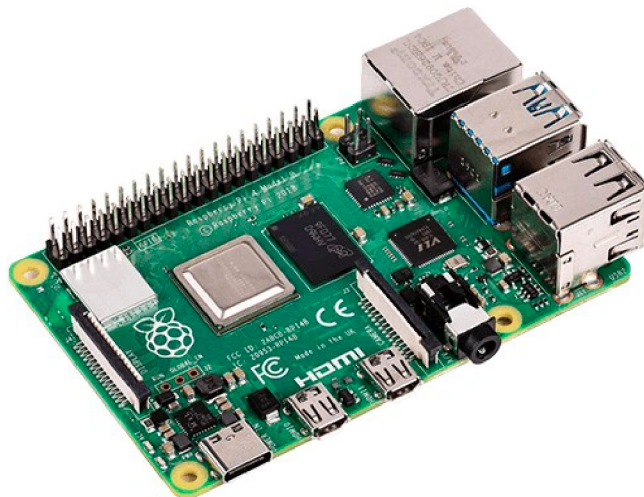
```

3.3 Server

Server je možné zařídit několika způsoby, např. hostováním serveru u soukromých společností nebo vytvořením vlastního serveru. V tomto případě bylo přistoupeno k vytvoření vlastního serveru. K tomu bylo využito Raspberry Pi 4B (RPi) v této specifikaci [36]:

- 64-bitový čtyřjádrový ARM-8 Cortex-A72 1,5 GHz
- 4 GB paměti RAM
- úložiště - 32 GB SD karta
- Připojení k síti - ethernet konektor RJ-45 nebo Wi-Fi

Tyto specifikace by pro naše účely měly dostačovat, jelikož se nebude jednat o žádné komerční řešení nýbrž pouze o ověření funkčnosti pro několik zařízení. V případě nedostatku paměti se dají připojit externí disky pomocí dvou USB.



Obrázek 3.5: Raspberry PI 4B [36]

Na SD kartu pro RPi byl pomocí programu Raspberry Pi Imageru nainstalován OS Linux, konkrétně jeho distribuce Ubuntu server verze 20.04.5 LTS. Po takto přeformátované kartě a zapojení RPi je již možné se k serveru připojit. Následně byly na RPi nainstalovány software dle schéma na obr. 3.1:

- MQTT broker - Mosquitto
- Node-Red
- databáze InfluxDB
- vizualizační software Grafana

Pro správnou funkci celé aplikace při vlastním hostování serveru je nutné mít veřejnou IP adresu, buď přímo k samotnému serveru či alespoň k routeru. Následně pokud má router veřejnou IP adresu je nutné přeměrovat dále zmiňované porty na adresu serveru. Pokud není nutné přistupovat ke Grafaně a Node-Redu z vnějšku sítě postačí přeměrovat pouze port MQTT brokera. Grafana a Node-Red budou dostupné pomocí lokální IP adresy serveru a portů.

3.3.1 MQTT Broker - Mosquitto

MQTT broker Eclipse Mosquitto byl vybrán především z důvodu, že se jedná o open-source brokera.[38] Tudíž je jeho použití zdarma, dále je také nenáročný na výkon systému, proto ho lze instalovat i na jednodeskové počítače s nízkou spotřebou energie, ale je možné ho využít i na enterprise serverech. Je dostupný pro mnoho operačních systémů, a podporuje i formy autorizace a autentizace uživatelů. Instalaci na linuxovém serveru provedeme pomocí následujících příkazů. [39] Brokera je možné využít ihned po nainstalování

bez žádné změny konfigurace, ale pomocí následujících příkazů je k serveru přiřazeno uživatelské jméno a heslo pro alespoň částečné zabezpečení. Vytvořený broker je provozován na IP adrese serveru a portu 1883, ale ten je možné libovolně nastavit.

```

sudo apt install mosquitto mosquitto-clients      % instalace brokera

echo "mqtt_username:mqtt_password" > pwfile      % vytvoreni souboru s uziva-
                                                    % telskym jmenem a heslem

mosquitto_passwd -U pwfile                        % zasifrovani souboru

sudo mv pwfile /etc/mosquitto/                   % presun souboru do konfi-
                                                    % guracni slozky

sudo nano /etc/mosquitto/mosquitto.conf          % otevreni souboru v editoru

allow_anonymous false                            % na konec souboru pred
password_file /etc/mosquitto/pwfile              % include_dir vlozime cestu
                                                    % k zasifrovanemu souboru

sudo /etc/init.d/mosquitto restart                % po ulozeni souboru
                                                    % provedeme restart brokera

```

3.3.2 Databáze InfluxDB

Influx DB [40] je open-source databáze, kterou je vyvíjena společností *InfluxData*. Pro využití v serverové aplikaci byla vybrána z toho důvodu, že je optimalizována pro rychlé vyhledávání a ukládání dat ve formě časových řad. Z toho důvodu se také hojně využívá v oblastech monitorování provozu, IoT a analýz v reálném čase. Její výkon dosahuje až ukládání stovek tisíc bodů za sekundu. Pro instalaci InfluxDB je možné použít její oficiální repozitář. [37] Instalaci na server je možné provést pomocí následujících příkazů. Přístup k databázi je pomocí IP adresy(localhost) a portu 8086.

```

% zadani repozitare
wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-key add -

% je nekolik verzi lisici se podle OS a jeho verze - pro Ubuntu 20.04.LTS
echo "deb https://repos.influxdata.com/ubuntu focal stable" | sudo tee
/etc/apt/sources.list.d/influxdb.list

sudo apt update                                  % aktualizace balicku
sudo apt install influxdb                        % instalace

```

```

sudo systemctl unmask influxdb           % pridani influxdb na seznam
sudo systemctl enable influxdb           % pro start po spusteni

sudo systemctl start influxdb            % manualni zapnuti

influx                                    % pripojeni
% vytvoreni uzivatele se vsemi pravy
CREATE USER admin WITH PASSWORD 'adminpassword' WITH ALL PRIVILEGES

```

Následně je zapotřebí vytvořené uživatelské jméno a heslo pro přístup do databáze vložit do konfiguračních souborů, jinak by byl vytvořený uživatel ignorován. Po vložení stačí restartovat InfluxDb a přihlásit se. Následně už jen stačí vytvořit prázdnou databázi pomocí následujícího příkazu. Jakmile do budou posílány data automaticky se uloží.

```
CREATE DATABASE <nazev databaze>
```

3.3.3 Node-Red

Node-RED [41] je grafický vývojový nástroj uzpůsobený pro propojování hardwarových zařízení, rozhraní API a online služeb. Vyvinut byl společností IBM a od roku 2016 je provozován jako open-source projekt. Od té doby si jej také automatizační průmysl osvojil jako snadný způsob pro vytváření průmyslových aplikací IoT, které shromažďují, zpracovávají a sdílejí data prostřednictvím internetu. V našem případě bude sloužit k propojení MQTT brokera s Influx DB. Instalaci se provede pomocí následujících příkazů [37], před ní je ale důležité nainstalovat ještě nástroj pro vytváření binárních modulů pomocí NPM. Dále je potřeba nainstalovat modul pro připojení k Influx DB. Pro větší bezpečnost je dobré i u této služby nastavit přihlašovací údaje. Pokud se chceme ke službě připojit stačí do webového prohlížeče zadat IP adresu serveru s portem 1880. Následně pomocí nastavení bloků a jejich pospojování vytvoříme potřebnou funkci.

```

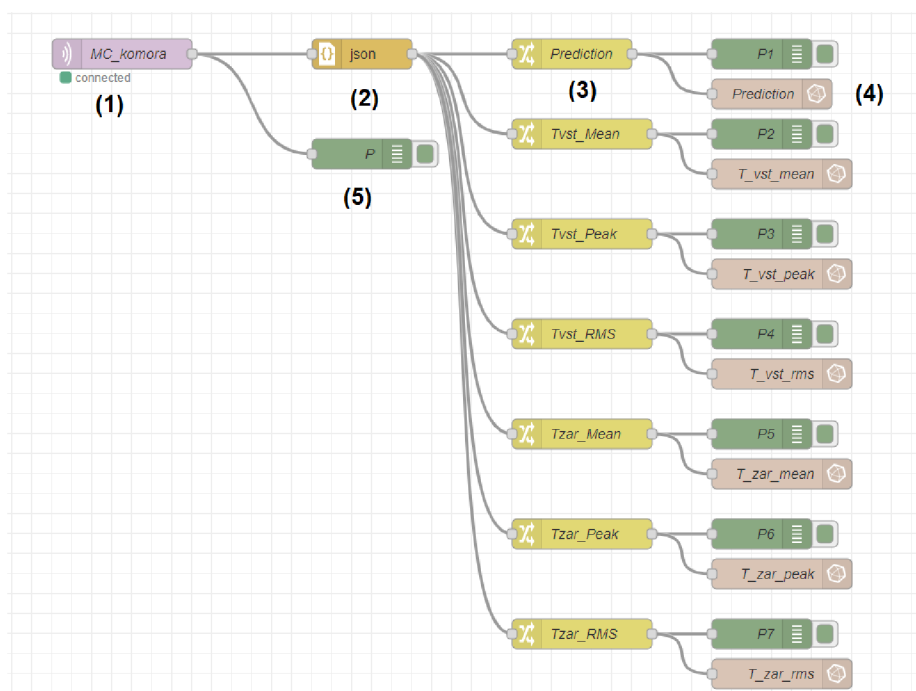
sudo apt install build-essential git      % instalace nástroje pro NPM
npm install node-red-contrib-influxdb    % instalace node-red
sudo systemctl enable nodered.service    % pridani influxdb na seznam
                                           % pro start po spusteni

sudo systemctl start nodered.service     % zapnuti node-red

```


Na obr. 3.6 je zobrazeno schéma pro ukládání dat pro tepelnou komoru, jednotlivé bloky mají následující funkci:

- **MC_komora (1)** - Slouží pro připojení k MQTT brokerovi, nastavují se v něm údaje k serveru (IP adresa, port, uživatelské údaje, atd.), odebíraný topic (MC_komora), QoS (0) a jaký je formát výstupu (a String).
- **json (2)** - slouží k převedení vstupující zprávy do formátu JSON
- **Prediction (3)** - slouží k vybrání určité proměnné z objektu, je v něm specifikován přesný typ proměnné (Např z payload vezmi pouze payload.MS02.Pred).
- **T_vst_mean (4)** - slouží k uložení dat do InfluxDB. Nastavují se v něm opět údaje k serveru (IP adresa, port, uživ. údaje), dále i název databáze (NB_IoT_DB) a název měření pod kterým budou data ukládána (measurement/MS02/Prediction).
- **P (5)** - tento blok slouží pro debugging, vypsání zprávy do konzole.



Obrázek 3.6: Schéma příjmu, zpracování a uložení zprávy v Node-Red

3.3.4 Grafana

Grafana [42] je open-source platforma vyvinutá společností Grafana Labs umožňující vizualizaci dat pomocí diagramů a grafů. Umožňuje vytvářet interaktivní panely pro analýzu a monitorování dat z různých zdrojů. Je pomocí ní možné nastavit různá upozornění vyplývající ze zobrazovaných dat. Interaktivní panely se mohou skládat z různých možností grafů jako např. grafy, měřidla, tabulky či heatmapy. Instalace [37] je také možná pomocí oficiálního repozitáře a probíhá následujícími příkazy. Grafana je jako Node-Red dostupná z webového prohlížeče pomocí IP adresy serveru a portu 3000.

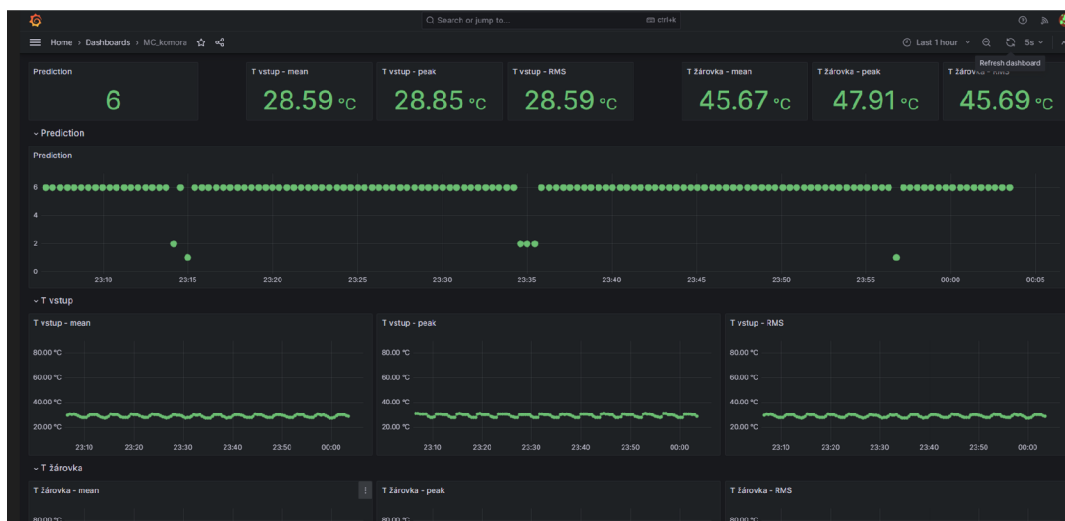
```
% nacteni repozitare
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -

echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a
/etc/apt/sources.list.d/grafana.list

sudo apt update                                % aktualizace balicku
sudo apt install grafana                        % instalace
sudo systemctl enable grafana-server           % pridani grafany na seznam
                                                % pro start po spusteni

sudo systemctl start grafana-server           % manualni zapnuti
```

Před vytvořením interaktivní panelu je nutné v nastavení v záložce *Connections* nastavit připojení k InfluxDB pro vyčítání dat do grafu. Následně byl vytvořen interaktivní panel pro zobrazování dat z tepelné komory (obr. 3.7) pomocí dvou druhů vizualizací pro veškeré posílané hodnoty (Time-Series Chart a Stats). Poté stačí ve vizualizacích nastavit zdroj dat a nastavit grafiku dle potřeby užití.

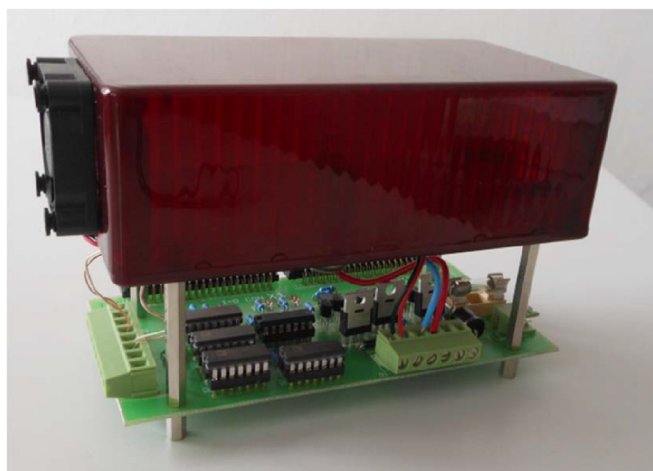


Obrázek 3.7: Ukázka interaktivního panelu pro tepelnou komoru v Grafaně

4 Prediktivní údržba

4.1 Tepelná komora

Výukový model Tepelná komora byl vytvořen v rámci bakalářské práce Ing. Tomáše Gregora pod vedením doc. Ing. Radka Vlacha, Ph.D. v roce 2013 [43]. Cílem bylo navrhnout a zrealizovat výukový model tepelné komory (obr. 4.1). Hotové zařízení slouží jako pomůcka pro ukázky regulace teploty ve výuce.



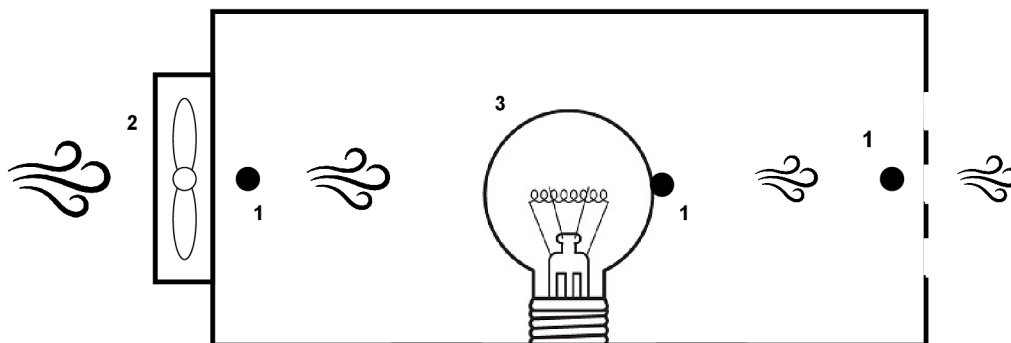
Obrázek 4.1: Tepelná komora [43]

4.1.1 Popis soustavy

Komora je vyrobena z průsvitné plastové krabičky se dvěma otvory na protilehlých stranách. Jeden otvor slouží pro nasávání vzduchu, u něho je umístěn ventilátor, a druhý otvor, kterým vhněný vzduch uniká z komory pryč. Uprostřed krabičky je osazena běžná autožárovka s instalovaným výkonem 21 W . Při průchodu krabičkou se vstupující vzduch ohřívá od zdroje tepla (autožárovky) a vytváří rozdíl teplot ΔT . Regulace teploty žárovky nebo vzduchu na výstupu z komory je zajišťována změnou otáček ventilátoru nebo změnou výkonu žárovky. Zjednodušené schéma modelu komory je na obr. 4.2.

Výkonová část komory je realizována pomocí tranzistorů a optočlenů. Je napájena z laboratorního zdroje o napětí $U = 12\text{ V}$ a řízena PWM signály z karty MF624. [44]

Měřicí část komory se skládá ze tří termočlánků typu K (1 v obr. 4.2), které jsou umístěny na vstupu a výstupu vzduchu z komory, a jeden je přilepen přímo na žárovce. Termočlánek je připojen na kompenzační obvod AD595, který napětí linearizuje, zesiluje

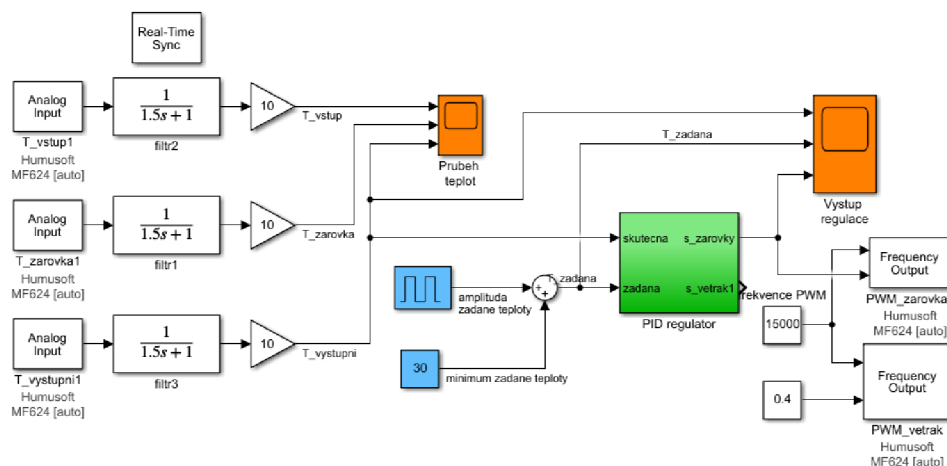


Obrázek 4.2: Schéma tepelné komory schéma, 1 - termočlánky, 2 - ventilátor, 3 - tepelný zdroj

na $10 \text{ mV}/^\circ\text{C}$ a navíc také zajišťuje kompenzaci studeného konce termočlánku. Výstup z obvodu AD595 je následně ještě zesílen pomocí operačního zesilovače kvůli využití celého rozsahu AD převodníku. Napájení měřicí části je zajištěno výstupním 12 V pinem karty MF624.

Měřicí karta je připojena pomocí dvou 40-pinových konektorů a je možné ji ovládat pomocí softwaru Simulink. Využity jsou dva PWM kanály s frekvencí 15 kHz pro řízení ventilátoru a žárovky. Dále jsou využity tři AD převodníky pro měření teplot v komoře.

V rámci BP [43] byly vytvořeny 2 simulinkovské modely. V této práci je využit model pro regulaci výstupní teploty komory při konstantních otáčkách ventilátoru a regulace je zajištěna změnou střídání žárovky pomocí PID regulátoru. Požadovaná výstupní teplota komory je měněna s periodou $T = 125 \text{ s}$ vždy na $T_{\text{výstupní}1} = 30^\circ\text{C}$ nebo $T_{\text{výstupní}2} = 32^\circ\text{C}$. Celá řídicí smyčka je spouštěna s frekvencí $f_{\text{smyčky}} = 100 \text{ Hz}$.



Obrázek 4.3: Regulace výstupní teploty komory v programu Simulink [43]

4.1.2 Úprava komory

Na komoře byla upravena regulace v Simulinku tak, že nejsou po celou dobu konstantní otáčky ventilátoru, ale pokud je požadována velká změna teploty, otáčky ventilátoru se zvýší či sníží změnou střídání PWM v rozmezí $< 0,25; 0,95 >$.

Dále byly vymodelovány 4 druhy krytek (obr. 4.4) se 7 otvory s různým průměrem, které byly vytisknuty na 3D tiskárně v dostupné laboratoři.

Jelikož byla komora navržena na řízení pomocí karty MF624, tak výstupní napětí termočlánku bylo zesilováno až na 10V viz podkap. 4.1.1. Pro řízení komory pomocí mikrokontroléru bylo nutné snížit výstupní napětí viz kap. 3.1. K tomu byly využity OP v zapojení napěťového sledovače a následně bylo výstupní napětí zmenšeno pomocí odporového děliče.



Obrázek 4.4: Krytky na otvory

4.1.3 Poruchové stavy

Na tepelné komoře byly uměle vytvořeny následující poruchové stavy, které budeme chtít pomocí algoritmů prediktivní údržby určovat jsou uvedeny v tab. 4.1:

Tabulka 4.1: Poruchové stavy tepelné komory

Stav	Název stavu	Popis
1	Referenční stav	měření za normálních podmínek
2	Zaizolování komory	izolace vrchní a bočních stran polystyrenem o tloušťce $t = 1 \text{ cm}$
3	Omezení průtoku vzduchu 1	omezen přívod vzduchu krytkou s otvory o $\varnothing 4 \text{ mm}$
4	Omezení průtoku vzduchu 2	omezen přívod vzduchu krytkou s otvory o $\varnothing 5 \text{ mm}$
5	Omezení průtoku vzduchu 3	omezen přívod vzduchu krytkou s otvory o $\varnothing 8 \text{ mm}$
6	Omezení průtoku vzduchu 4	omezen odvod vzduchu krytkou s otvory o $\varnothing 4 \text{ mm}$

4.1.4 Realizace algoritmů pro prediktivní údržbu - MATLAB

Měření dat

Pro měření dat byl vytvořen skript k automatizování celého měření z důvodu časové náročnosti. Skript zajišťuje měření daného počtu stavů a jejich opakování. Při jednotlivém měření je zavolán simulinkový model pro regulaci výstupní teploty, který řídí komoru po dobu celého měření a snímá data z AD převodníku. Model byl oproti tomu zmíněnému v kap. 4.1.1 a 4.1.2 doplněn o vyčítání hodnot teplot v komoře do MATLAB Workspace.

Naměřené průběhy teplot v komoře (Vstupní, Žárovky a Výstupní) byly ořezány pouze na čas měření. To je zapotřebí kvůli ustálení teploty na výstupu z komory před zahájením měření. Je to i z toho důvodu, že signál z AD převodníků je softwarově filtrován filtry typu dolní propust s mezní frekvencí $f = 0,11 \text{ Hz}$. Filtry byly do modelu zahrnuty z důvodu velkého rušení TČ výkonovými obvody.

Následně jsou data upravena do formy tabulky obsahující časovou tabulku (Timetable) pro každou z měřených teplot a aktuální měřený stav komory. Časová tabulka obsahuje naměřený signál i s časem naměření. To je nutné z důvodu následného použití DFD, který takto vložená data vyžaduje. Nakonec se tabulka uloží na disk a pojmenuje dle měřeného stavu a aktuálního cyklu.

Data byla měřena pro každý stav 30 krát tak, abychom měli dostatek dat pro natrénování klasifikátoru i následnou validaci. Proto byla data následně rozdělena na 25 a 5 měření.

Vytvoření a určení klasifikačních atributů

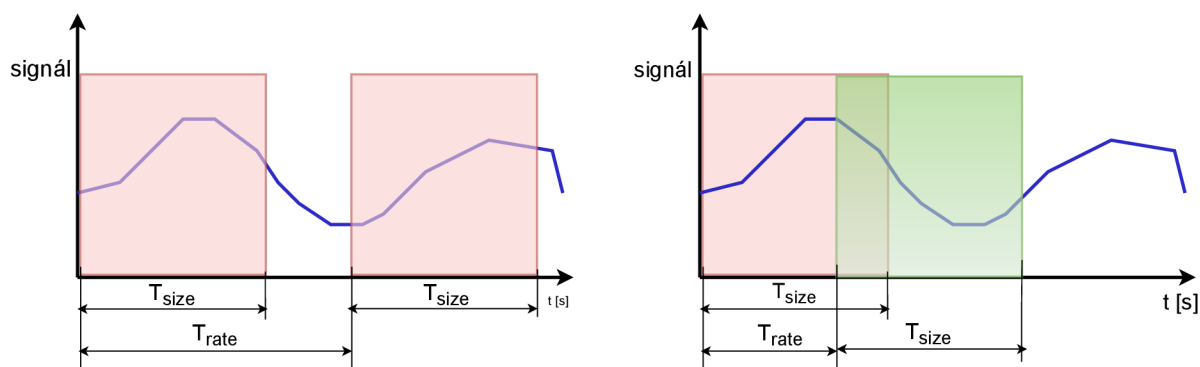
Z toho důvodu, že data již byla filtrována při měření. Byla bez dalších úprav vložena pomocí skriptu do jedné souhrnné tabulky, vhodné pro vložení do Diagnostic Feature Designeru. Pomocí něho byla vygenerována funkce k vytváření atributů vkládaných dat. Vytvořená funkce převádí vstupní signály z časové oblasti do frekvenční a vypočítává všechny dostupné příznaky v časové oblasti, tak i ve frekvenční.

Po vytvoření příznaků a ohodnocení pomocí funkce ReliefF zmíněné v kap.2.3.4. Výsledné ohodnocení pomocí této funkce je v tab. 4.2.

Tabulka 4.2: Nejlepších 10 atributů dat po ohodnocení - $T = 250s$

Pořadí	Atribut
1	$T_{vstup} - STD$
2	$T_{vstup} - Mean$
3	$T_{vstup} - RMS$
4	$T_{vstup} - CrestFactor$
5	$T_{vstup} - ImpulseFactor$
6	$T_{vstup} - ClearanceFactor$
7	$T_{vstup} - PeakValue$
8	$T_{vystup} - THD$
9	$T_{vstup} - ShapeFactor$
10	$T_{zarovka} - Mean$

Dle tabulky 4.2 byl následně natrénován klasifikátor (kap. 4.1.4). Ačkoliv klasifikátor dosahoval dobrých výsledků, Bylo rozhodnuto, že perioda výpočtu a klasifikace stavu 1x za 250 s je nedostatečná a bude snaha ji zkrátit.



Obrázek 4.5: Vytvoření časových úseků v DFD

K tomu byla opět využita vygenerovaná funkce pomocí DFD, kde lze před vygenerováním nastavit časový úsek pro rozdělení dat (T_{size}) a časový úsek pro start nového výpočtu atributu (T_{rate}) viz. obr. 4.5. Po vygenerování byla funkce použita na datech pro různou periodu výpočtu atributů s tím, že se vždy $T_{size} = T_{rate}$. Následně byly pomocí atributů natrénovány klasifikátory (kap. 4.1.4), po zhodnocení výsledků byla určena perioda výpočtu $T = 25 s$, jejíž nejlepší atributy jsou zobrazeny v tab. 4.3.

Tabulka 4.3: Atributy dat po ohodnocení - $T = 25s$

Pořadí	Atribut
1	$T_{vstup} - RMS$
2	$T_{vstup} - Mean$
3	$T_{vstup} - PeakValue$
4	$T_{zarovka} - Mean$
5	$T_{zarovka} - RMS$
6	$T_{zarovka} - PeakValue$

Klasifikátor

K trénování klasifikátoru byla využita aplikace Classification Learner dostupná v programu MATLAB (kap. 2.3.4). V aplikaci byly pomocí grafického rozhraní natrénovány všechny dostupné klasifikátory (jako Decision trees, Discriminant analysis, SVM, atd.) pro atributy s rozdílnými intervaly výpočtu. Klasifikátory byly vždy trénovány na šestici nejlepších atributů s nastavením křížové validace na 5, SVM multiclass method: One-vs-One a zapnutou standardizací dat. Počet 6 nejlepších atributů byl stanoven vzhledem k tomu, že se výpočet atributů a predikce budou provádět přímo na mikrokontroléru, který má omezené výpočetní možnosti, ale pro výpočet 6 atributů by měl být dostatečný. Porovnání dosažených přesností je uvedeno v tabulce 4.4. Z ní je patrné, že nejlepší klasifikace stavů bylo dosaženo s periodou výpočtu atributů $T = 25 s$. Pro tuto periodu jsou zobrazeny matice zmatenosti pro měřená (obr. 4.6), tak testovací data (obr. 4.7). Klasifikátor na validačních datech dosahuje přesnosti 91.9 %.

Tabulka 4.4: Porovnání klasifikátorů pro periody výpočtu atributů

Perioda výpočtu [s]	Klasifikátor	Přesnost [%]
250	Quadratic SVM	92.0
50	Fine Gaussian SVM	89.7
25	Fine Gaussian SVM	92.2
15	Fine Gaussian SVM	89.5
10	Fine Gaussian SVM	89.2
5	Fine Gaussian SVM	88.6
3	Fine Gaussian SVM	88.8

True Class	1	587	10			28	
	2	5	563			51	6
	3	3		621			1
	4			1	624		
	5	79	79			467	
	6		4		6		615
		1	2	3	4	5	6
	Predicted Class						

Obrázek 4.6: Matice zmatenosti pro klasifikaci Fine Gaussian SVM pro $T = 25 s$ - trénovací data

True Class	1	116				9	
	2		111			14	
	3			125			
	4				125		
	5		29			96	
	6				9		116
	1	2	3	4	5	6	
	Predicted Class						

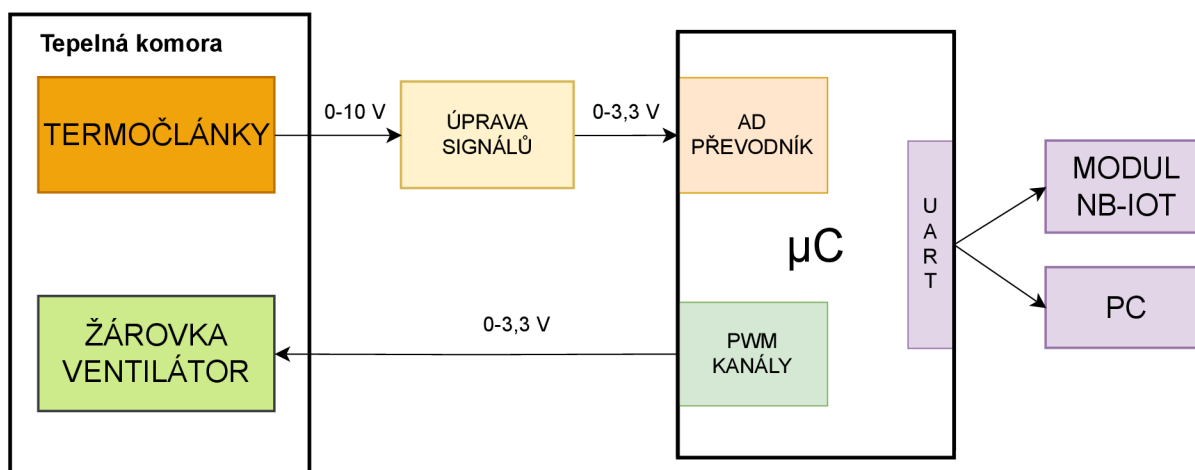
Obrázek 4.7: Matice zmatenosti pro klasifikaci Fine Gaussian SVM pro $T = 25s$ - testovací data

4.1.5 Implementace řízení a prediktivní údržby na mikrokontroléru

Pro implementaci na mikrokontrolér bylo potřeba upravit signály z termočlánku (kap. 3.1 a 4.1.2). Pro řízení komory byly zapotřebí konkrétně tyto periferie:

- 3x AD převodník
- 2x UART
- 3x Timer
- 2 PWM kanály

Zapojení jednotlivých částí hardwaru jako mikrokontrolér, bezdrátový modul, tepelná komora, napájecí zdroje, obvodů pro úpravu signálu z termočlánku je zobrazeno na obr. 4.8 a vychází z obecného schéma na obr. 3.3 v kap. 3.1.



Obrázek 4.8: Blokové schéma hardware

Většina kódu byla naprogramována pomocí bloků (funkcí), proměnných typu struktura a předávání proměnných pomocí adres většinou na předchozí blok. Schematické řazení bloků celého softwaru je zobrazeno na obr. 4.9. Celý program v mikrokontroléru byl rozdělen na 4 části podle toho, kdy je potřeba daný kód vykonávat, volání funkce probíhalo většinou pomocí přerušování od jednotlivých časovačů:

- **Řízení komory - Timer 1**

Řídicí smyčka komory se příliš neliší od smyčky v měřícím modelu (obr. 4.3), je pouze doplněna o sběr dat pro výpočet atributů (obr. 4.9). Smyčka je taktéž volána s frekvencí $f_{smyčky} = 100 \text{ Hz}$, jak bylo uvedeno v kap. 4.1.1.

- **Výpočet atributů - Timer 2**

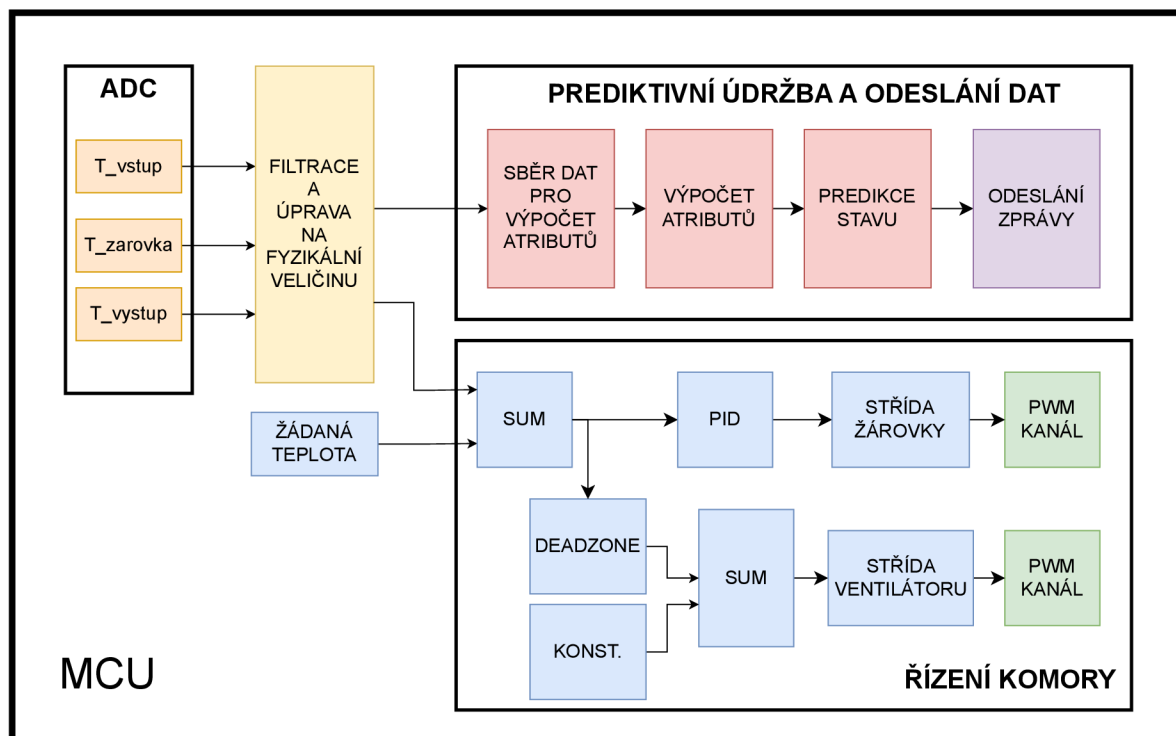
Funkce pro výpočet atributů je volána každých 25s, hlavním cílem je přepsat si výsledné hodnoty ze sběru dat pro výpočet atributů a následně je vymazat pro další běh programu. Poté je proveden výpočet atributů, následovaný funkcí predikce stavu. Poté je vytvořena zpráva pro odeslání na server a dán pokyn pro vykonání v hlavní smyčce programu.

- **Změna žádané teploty - Timer 3**

Funkce pro změnu žádané teploty je volána 1x za 125s a přepíná mezi požadovanými teplotami.

- **Odeslání zprávy na server - Main**

Odeslání dat na server pomocí modulu. Obstarává komunikaci s modulem pomocí sběrnice UART, aby proběhl přenos dat. Vykonává se v hlavní smyčce programu po nastavení příznaku pro odeslání v po predikci stavu.



Obrázek 4.9: Blokové schéma softwaru na MCU

Funkce pro predikci stavu byla získána exportem Compact modelu z Classification Learner app. Následně byl exportovaný model uložen pomocí následujících příkazu [45]:

```
saveLearnerForCoder(trainedModel.ClassificationSVM, '
ClassificationLearnerModel')
```

Pro takto exportovaný model byla vytvořena funkce pro volání modelu a vytvoření predikce. Tato funkce byla následně vložena do aplikace Matlab Coder, která dokáže kód napsaný v jazyce MATLAB převést do jazyka C. Po vyzkoušení funkce a ověření predikce ale bylo zjištěno, že nahrání vygenerovaných funkcí není možné z důvodu nedostatku místa v paměti. Z toho důvodu bylo přistoupeno na přetypování vstupujících hodnot do klasifikátoru z hodnot *double* na *float*, což snížilo přesnost dat, a vystupujících z *double* na *int32*. Na nově natrénovaném modelu, ale nebyl patrný žádný rozdíl. Přesnost modelu nepatrně snížila, a to na 91,9 %, což vychází z matice zmatenosti pro validační data. Poté byla funkce opět vygenerována do jazyka C a již se do paměti mikrokontroléru vešla.

4.1.6 Experiment - ověření funkčnosti

Po zprovoznění celého systému již zbývalo otestovat jeho funkci. Toho bylo docíleno provozem systému v různých stavech podle kap. 4.1.3. Zkoušení tedy proběhlo otestováním každého stavu vždy provozem po dobu zhruba 1 hodiny. Za tu dobu bylo provedeno okolo 140 predikcí a ty jsou uvedeny na obr. 4.10.

True Class	1	117				41	
	2	5	127			10	
	3			134	10		
	4				130	14	
	5	17	2			125	
	6						143
		1	2	3	4	5	6
		Predicted Class					

Obrázek 4.10: Matice zmatenosti - reálný provoz

True Class	1	92.8%				7.2%	
	2		88.8%			11.2%	
	3			100.0%			
	4				100.0%		
	5		23.2%			76.8%	
	6				7.2%		92.8%
		1	2	3	4	5	6
		Predicted Class					

(a) Validační data

True Class	1	74.1%				25.9%	
	2	3.5%	89.4%			7.0%	
	3			93.1%	6.9%		
	4				90.3%	9.7%	
	5	11.8%	1.4%			86.8%	
	6						100.0%
		1	2	3	4	5	6
		Predicted Class					

(b) Reálný provoz

Obrázek 4.11: Porovnání matic zmatenosti ověřovacího experimentu na tepelné komoře

Přesnost predikce na reálném zařízení dosáhla 88,7 %, vypočítána byla pomocí obr. 4.10. Jak ale můžeme vidět z obr. 4.25 je patrné, že např. ve stavu 1, tedy v normálním

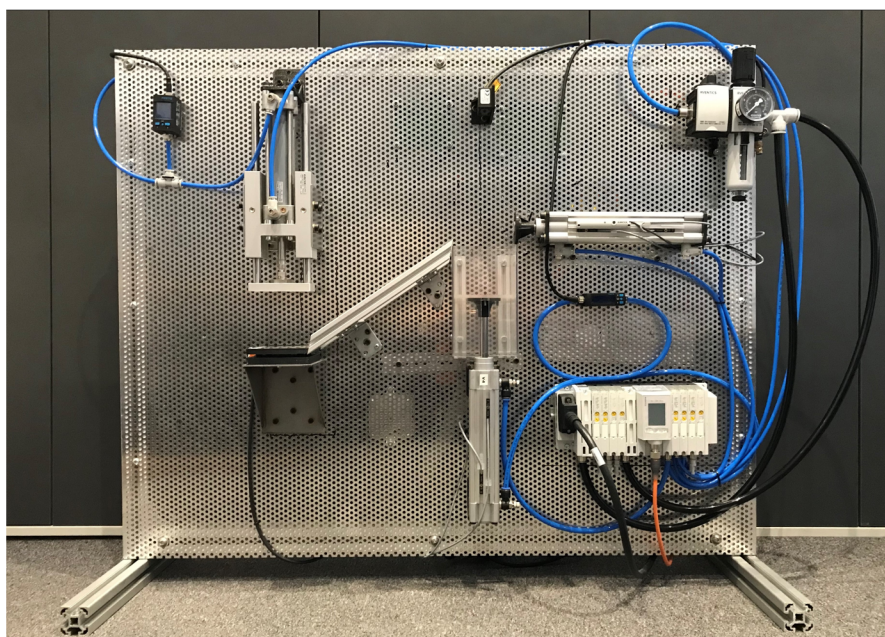
provozu, je mnohem častěji predikován stav 5 oproti než v případě predikce na validačních datech. Když se podíváme na graf s jednotlivými predikcemi (obr. 4.12), je z něho patrné, že predikce se střídá se správným a špatným stavem. Když ale porovnáme všechna data i vypočítané atributy, zjistíme, že špatná predikce nastává v případě ochlazování komory (žádaná teplota $T_{\text{žádaná}} = 30^{\circ}\text{C}$). Když porovnáme hodnoty atributů mezi validačními a získanými daty, zjistíme, že teplota žárovky příliš klesne, což má za následek již zmíněnou špatnou predikci stavu. Na přílišné klesnutí teploty žárovky měl pravděpodobně vliv jiný průběh ochlazování komory, což mohlo být způsobeno rozdílným prouděním vzduchu kolem komory, či mírně rozdílnou teplotou v místnosti. Na druhou stranu přesnost predikce u jiných stavů se díky odlišnému proudění zlepšila. Celkově se ale na soustavě povedlo predikovat stavy za pomoci prediktivní údržby.



Obrázek 4.12: Graf predikce - testování - stav 1

4.2 Pneustand

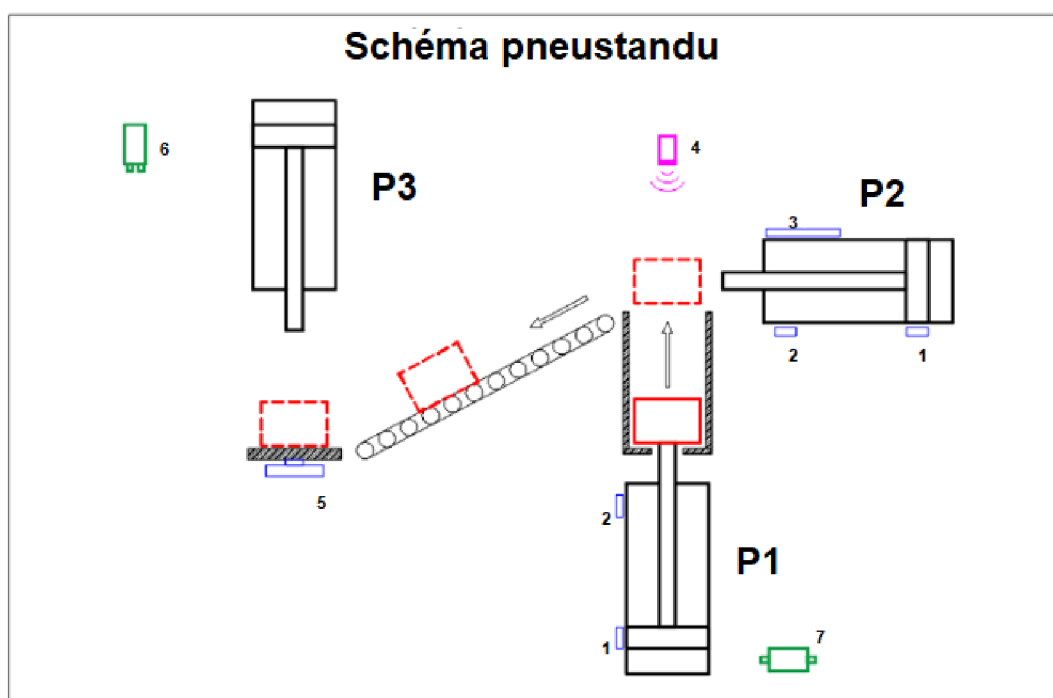
Výukový model Pneustandu byl vytvořen v rámci bakalářské práce Davida Javorka pod vedením Ing. Jana Najmana, Ph.D. v roce 2023. [46] Cílem bylo navrhnout a zrealizovat výukový model s pneumatickými aktuátory (obr. 4.13), jelikož se v laboratoři žádný nevykytoval. K němu bylo také vytvořeno ovládání z PC a fyzikální počítačový model.



Obrázek 4.13: Pneustand [46]

4.2.1 Popis soustavy

Celá soustava pneustandu reprezentuje dopravní linku s lisem, jejíž hlavním úkolem je přesun dřevěné kostky ze "zásobníku" pod "lis" a následné zalisování součásti požadovanou silou. Soustava se skládá ze 3 pneumatických dvojčinných válců a řady senzorů popsaných dále v této podkapitole. Popis funkce (obr. 4.13) je následující: První píst (P1) dopravuje dřevěné kostky vertikálně směrem do výšky, poloha pístu je snímána laserovým snímačem (5) a magnetickými snímači (1,2). Píst je doplněn o snímač průtoku vzduchu (7) na vstupu pro vyjetí pístu. Píst 2 (P2), orientovaný horizontálně, kostku posune z pístu 1 a ta následně po skluzavce sjede k pístu 3 (P3). Píst 2 je osazen lineárním magnetickým snímačem polohy (3) a magnetickými koncovými snímači (1,2). Pod pístem 3, který na pneustandu reprezentuje lis, je umístěna podložka obsahující tenzometr (4), pro snímání přítláčné síly pístu. Dále je píst 3 osazen senzorem tlaku (6) na vstupu pro vyjetí. Specifikace použitých pneumatických válců a senzorů jsou uvedeny v tabulkách 4.6 a 4.5. Většina senzorů je napájena pomocí 24 V, pouze tenzometr je napájen 5 V.



Obrázek 4.14: Schéma pneustandu [46]

Tabulka 4.5: Sensorická část pneustandu

Senzor	Typ	Rozsah	Výstup	ID (obr.)
Magnetické koncové snímače	SMT-8M-A-PS-24V-E-2,5-OE [47]	-	24 V	1,2
Magnetický lineární snímač polohy	SM6 R412010144 [48]	0 – 64 mm	0 – 10 V	3
Laserový snímač	optoNCDT ILD 1220-500 [49]	100 – 600 mm	4 – 20 mA	4
Tenzometr	FX293X-100A-0100-L [50]	0 – 100 lb	0,5 – 4,5 V	5
Tlakový snímač	SPAU-P10R-W-G18FD-L-PNLK-PNVBA-M8U [51]	0 – 10 bar	0 – 10 V	6
Průtokoměr	SFAH-50U-Q8 [52]	0 – 50 l/min	0 – 10 V	7

Tabulka 4.6: Specifikace pneumatických valců

	Píst 1	Píst 2	Píst 3
výrobce	Festo		Aventics
označení	DSBC-U-32-130-D3-PA-N3 [53]	DSBC-32-130-D3-PPSA-N [53]	TRB-32-100/R480600272 [54]
průměr pístu [mm]	36		32
zdvih pístu [mm]	130		100
průměr pístnice [mm]	12		12
Další informace	obsahuje magnety pro snímání polohy magnetickými snímači		možnost nastavení vlastního odpružení v koncových polohách pístu
ID (obr.)	P1	P2	P3

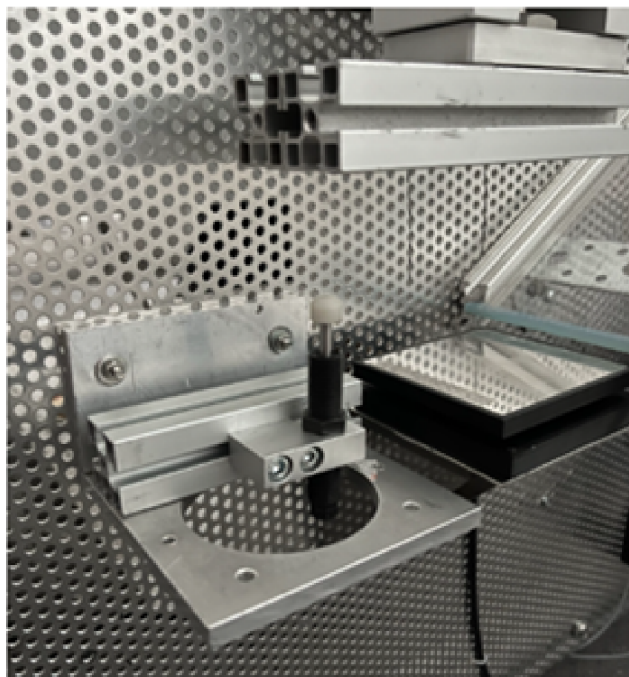
Celá soustava je řízena pomocí karet MF634 a NI-9474. Karta MF-634 zajišťuje měření dat ze snímačů s frekvencí vzorkování $f_{vz} = 1000 \text{ Hz}$. Karta NI-9474 od společnosti *National Instruments* disponuje 8 výstupními piny pro napětí v rozsahu 5 – 30 V pro spínání dvoucestných ventilů v pneumatickém rozvaděči. Výstupní piny jsou ovládány s frekvencí $f_{vz} = 100 \text{ Hz}$. Následně je k celému řízení využit model v prostředí Simulink, pomocí kterého se dají měřecí karty ovládat. Je v něm graficky naprogramována řídicí

logika. Dále je model doplněn o samostatné řízení pístů.

Jelikož je k řízení použita karta MF634 a v tab. 4.5 ve sloupci Výstup je vidět, že některé senzory nemohou být ke kartě připojené, takže je nutné jejich výstup upravit. Jedná se o magnetické koncové snímače jejichž výstupní napětí musí být sníženo z 24 V na 5 V. Toho je docíleno pomocí optočlenů. Dále pak není možné ke kartě připojit laserový snímač, který má proudový výstup, ten musí být před vstupem do karty upraven pomocí převodníku s potenciometrem.

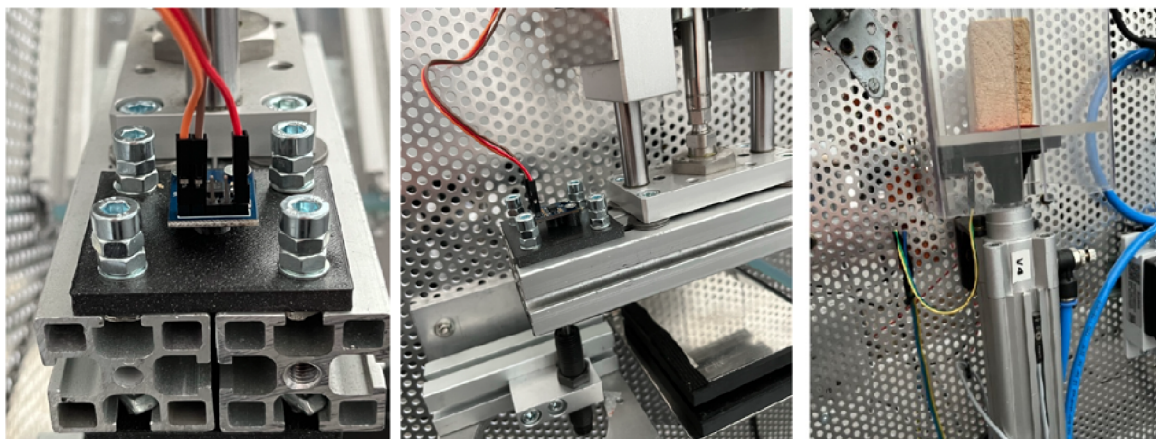
4.2.2 Úprava soustavy

Soustava byla doplněna o nastavitelný tlumič, který u pístu 3 simuluje zalisování součástí. K tomu byl využit konkrétně tlumič MA 150 M-B od společnosti ACE [55], s vratnou silou 3 – 5 N a maximální absorbovanou energií na jeden cyklus 17 Nm. Ten byl na soustavu přidělán pomocí hliníkové konstrukce zobrazené na obr. 4.15. Poté byly přidělané hliníkové profily na píst 3 pro dosažení k tlumiči, doplněné o podložku vytištěnou na 3D tiskárně.



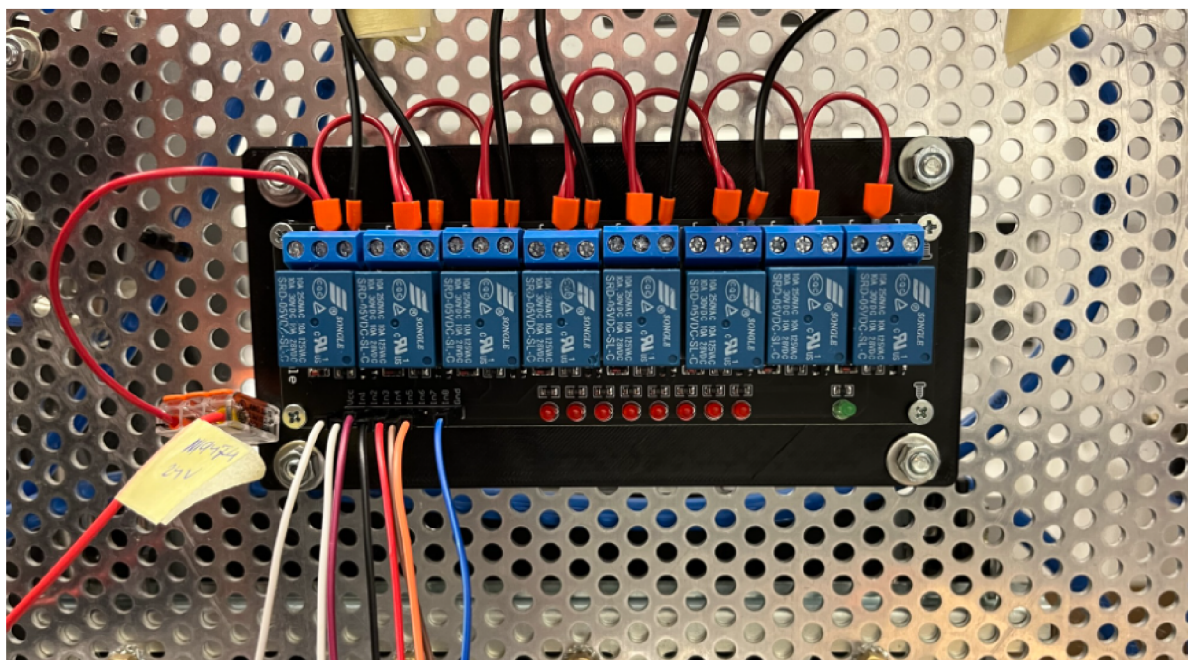
Obrázek 4.15: Přidělení tlumiče na pneustand

Na hliníkové profily byl následně pomocí držáku připevněn akcelerometr GY-61 ADXL335 [56] (obr. 4.16), který je napájen ze zdroje napětí 5 V a připojen na měřicí kartu MF634. Jeho rozsah je $\pm 3 g$ a citlivost $300 mV/g$. Stejný typ akcelerometru byl přidělán i na píst 1 (obr. 4.16). Jako poslední byl na pneustand přidělán uzavírací ventil ke vstupu na píst č. 2, přidělán na přípravku pro zajištění vždy stejného uzavření.



Obrázek 4.16: Přidělení akcelerometrů na písty

Pro implementaci na řízení pomocí mikrokontrolerů musely být napěťové signály upraveny na maximální hodnotu 3,3V. Toho bylo docíleno opět použitím napěťových sledovačů a odporových děličů. Pro řízení pomocí mikrokontroleru musela být ještě nahrazena karta NI-9474. To bylo vyřešeno pomocí modulu s 8 kusy relé [57], které jsou schopné i pomocí 3,3V logiky spínat zátěž o napětí do 30 VDC a 2 A. Následně byl tento modul přidělán na pneustand a připojen k řízení rozvaděče (obr. 4.17).



Obrázek 4.17: Přidělení relé modulu

4.2.3 Poruchové stavy

Uměle vytvořené poruchové stavy, které budeme chtít na pneustandu identifikovat pomocí prediktivní údržby, jsou uvedeny v tab. 4.7.

Tabulka 4.7: Poruchové stavy pneustandu

Stav	Název stavu	Popis
1	Referenční stav	nastavený tlak 3 bary, tlumič pol. č. 5
2	Těžší kostka	kostka o $m = 40\text{ g}$ nahrazena těžší kostkou o $m = 80\text{ g}$
3	Bez kostky	zalisování bez součásti na dopravní lince
4	Tlumič poloha č. 2	tlumič simulující zalisování součásti je nastaven na lehčí stupeň
5	Tlumič poloh č. 8	tlumič simulující zalisování součásti je nastaven na těžší stupeň
6	Snížení tlaku	snížený vstupní tlak do soustavy, $p = 1,3\text{ bar}$
7	Zavřený ventil u P2	simulace zanesení/zaškrcení přívodu vzduchu do pístu

4.2.4 Realizace algoritmů pro prediktivní údržbu - MATLAB

Měření dat

Pro měření dat byl opět vytvořen skript pro automatizování celého měření, podobný tomu použitému v kap. 4.1.4. Skript tedy zajišťuje měření daného počtu stavů a jejich opakování, spouštění měřicího modelu pneustandu, který má na starosti celé řízení. Model z BP[46] byl akorát doplněn o snímání všech senzorů aplikovaných na pneustandu, i těch přidanych, a vyčítání hodnot po ukončení měření do MATLAB workspace.

Po naměření hodnot jsou signály ořezány pouze na čas měření z důvodu inicializace celého systému před a po spuštění. Po oříznutí signálu jsou data upravena do formy tabulky obsahující 11 časových tabulek (Timetable) pro každý z měřených signálů a aktuální měřený stav pneustandu. Nakonec se tabulka uloží a pojmenuje dle měřeného stavu a aktuálního cyklu. Data byla naměřena pro každý ze stavů 30krát, abychom měli dostatek dat pro natrénování klasifikátoru i následnou validaci. Proto byla data následně rozdělena na 25 a 5 měření.

Určení klasifikačních atributů a klasifikátor

Po dokončení měření byla data ještě dále zpracována. K tomu bylo přistoupeno z důvodu zobrazení dat v grafu, kde bylo zjištěno, že je možné některá data dále oříznout. To bylo možné např. u dat ze senzorů u pístů 1,2 a u pístu 3, kde bylo patrné, že při práci pístů 1,2 jsou všechny hodnoty senzorů u pístu 3 nulové a naopak. Proto byly vybrány pouze ty úseky, které budou mít vypovídající hodnotu pro další zpracování. Data byla opět filtrována filtrem typu dolní propust již při měření a z toho důvodu nebyla již dále filtrována.

Byla pouze odečtena konstantní složka ze signálu z akcelerometrů z toho důvodu, že se jednalo o absolutní snímače zrychlení a v tomto případě neměla gravitační složka zrychlení žádný vliv na naši soustavu, naopak zkreslovala signál pro výpočet atributů. K provedení úprav je v příloze dostupný skript. Všechna naměřená data byla vložena do tabulky pro vložení do DFD.

Pomocí DFD byla vytvořena funkce pro vytvoření atributů dat. Vytvořená funkce převádí vstupní signály z časové oblasti do frekvenční a vypočítává všechny dostupné atributy jak v časové oblasti, tak i ve frekvenční. Po vypočítání atributů a jsou ohodnoceny pomocí funkce ReliefF zmíněné v 2.3.3. Seřazené příznaky po ohodnocení jsou zobrazeny v tabulce 4.8.

Tabulka 4.8: Pořadí atributů dat po ohodnocení

Pořadí	Atribut
1	<i>Pist 1 zajet – Mean</i>
2	<i>Pist 1 zajet – RMS</i>
3	<i>Pist 1 zajet – STD</i>
4	<i>Pist 1 zajet – Skewness</i>
5	<i>Pist 1 zajet – CrestFactor</i>
6	<i>Pist 1 zajet – ShapeFactor</i>
7	<i>Pist 1 zajet – ImpulseFactor</i>
8	<i>Pist 1 zajet – Kurtosis</i>

Jak lze z tabulky 4.8 vidět, všechny atributy vychází pouze z jednoho měřeného signálu. Na základě ohodnocení tedy proběhlo natrénování všech dostupných klasifikátorů v aplikaci Classification learner app jako v kap. 4.1.4. Výsledná matice zmatenosti z nejlepšího klasifikátoru, kterým byl Medium Gaussian SVM, je zobrazena na obr. 4.18. Je z něho patrné, že tyto určené parametry jsou vhodné pouze pro určení stavu 6 a 7.

1	7	5	1	6	5	1	
2	3	14		5	3		
3	6	7	1	7	4		
4	7	5	1	8	4		
5	5	6		9	5		
6						25	
7							25
	1	2	3	4	5	6	7

Predicted Class

Obrázek 4.18: Klasifikátor SVM - ohodnocení pro všechny stavy

Proto byla provedena další ohodnocení vždy pouze pro některé stavy. Další ohodnocení bylo provedeno pouze pro vypočtené atributy pro všechny stavy kromě 7. Nadále bylo provedeno pro ohodnocení pro stavy 1,2 a 3, a nakonec ještě ohodnocení pouze pro první 2 stavy. Výsledné pořadí prvních třech atributů u každého ohodnocení je zobrazeno v tabulce 4.9.

Tabulka 4.9: Pořadí atributů dat po ohodnocení pro různé stavy

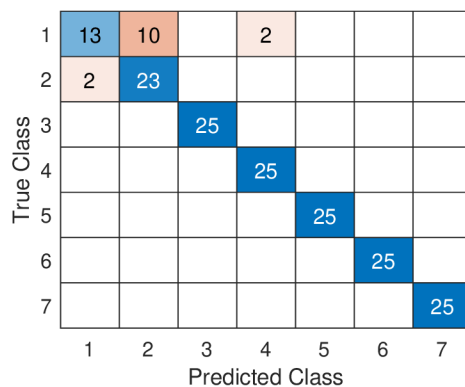
Pořadí	Atribut
Všechny stavy kromě 7	
1	<i>Píst 3 tlak – Kurtosis</i>
2	<i>Píst 3 tlak – Clearance Factor</i>
3	<i>Píst 3 tlak – Impulse Factor</i>
Stavy 1 - 3	
1	<i>Píst 1 laser – Mean</i>
2	<i>Píst 1 laser – Shape Factor</i>
3	<i>Píst 1 laser – RMS</i>
Stavy 1 a 2	
1	<i>Píst 2 poloha – magnet. – Crest Factor</i>
2	<i>Píst 1 průtokoměr – Shape Factor</i>
3	<i>Píst 2 poloha – magnet – Kurtosis</i>

Z tabulek 4.8 a 4.9 byl vybrán vždy první atribut. Na těchto attributech byly následně natrénovány klasifikátory. Z natrénovaných klasifikátorů a jejich matic zmatenosti bylo patrné, že je stále problém správně detekovat první dva stavy, jelikož v nich není příliš velký rozdíl. Z toho důvodu byl přidán další atribut pro klasifikátor, dle tabulky 4.9 pro stavy 1 a 2, konkrétně *Píst 1 průtokomer – Shape Factor*. Dále byl atribut *Kurtosis* z důvodu náročnosti jeho výpočtu na paměť mikrokontroléru nahrazen následujícím atributem *Píst 3 tlak – Clearance Factor*. Soupis finálních atributů pro klasifikátor následně implementovaný na mikrokontrolér je v tabulce 4.10.

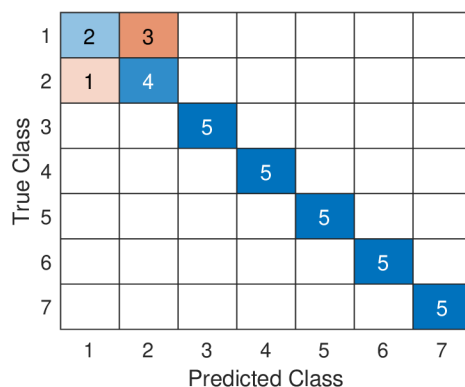
Tabulka 4.10: Finální výpis atributů pro klasifikaci

Atribut
<i>Píst 1 zajet – Mean</i>
<i>Píst 3 tlak – Clearance Factor</i>
<i>Píst 1 laser – Mean</i>
<i>Píst 2 poloha – magnet. – Crest Factor</i>
<i>Píst 1 průtokoměr – Shape Factor</i>

Nejlepším klasifikátorem byl v tomto případě Quadratic SVM, s nastavením křížové validace na 5, SVM multiclass method: One-vs-One a zapnutou standardizací dat. Ten ale při výsledném použití na mikrokontroleru a ověření funkčnosti nebyl schopný správně identifikovat stavy 1 a 2. Proto bylo přistoupeno k využití druhého nejlepšího klasifikátoru, kterým byl Medium Gaussian SVM se stejným nastavením jako Quadratic SVM. Jeho výsledné matice zmatenosti jsou zobrazeny na obr. 4.19 pro trénovací data a na obr. 4.20 pro data validační.



Obrázek 4.19: Klasifikátor SVM - finální verze - trénovací data



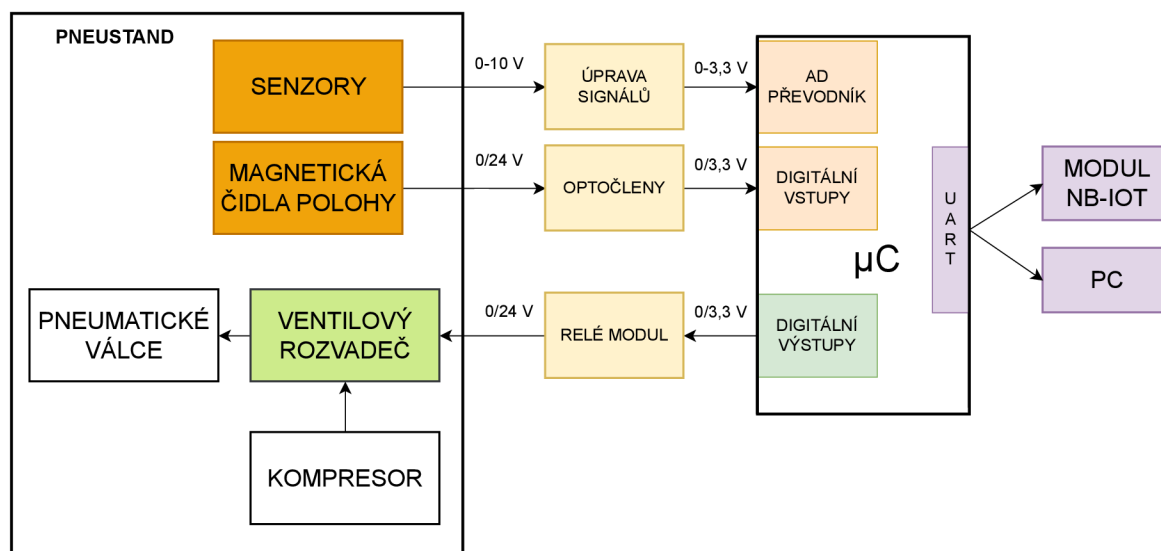
Obrázek 4.20: Klasifikátor SVM - finální verze - validační data

4.2.5 Implementace řízení a prediktivní údržby na mikrokontrolér

K řízení byla opět využita vývojová platforma (*MECH*)*SOFT* popsána v kap. 3.1, v tomto případě bylo zapotřebí těchto periférií:

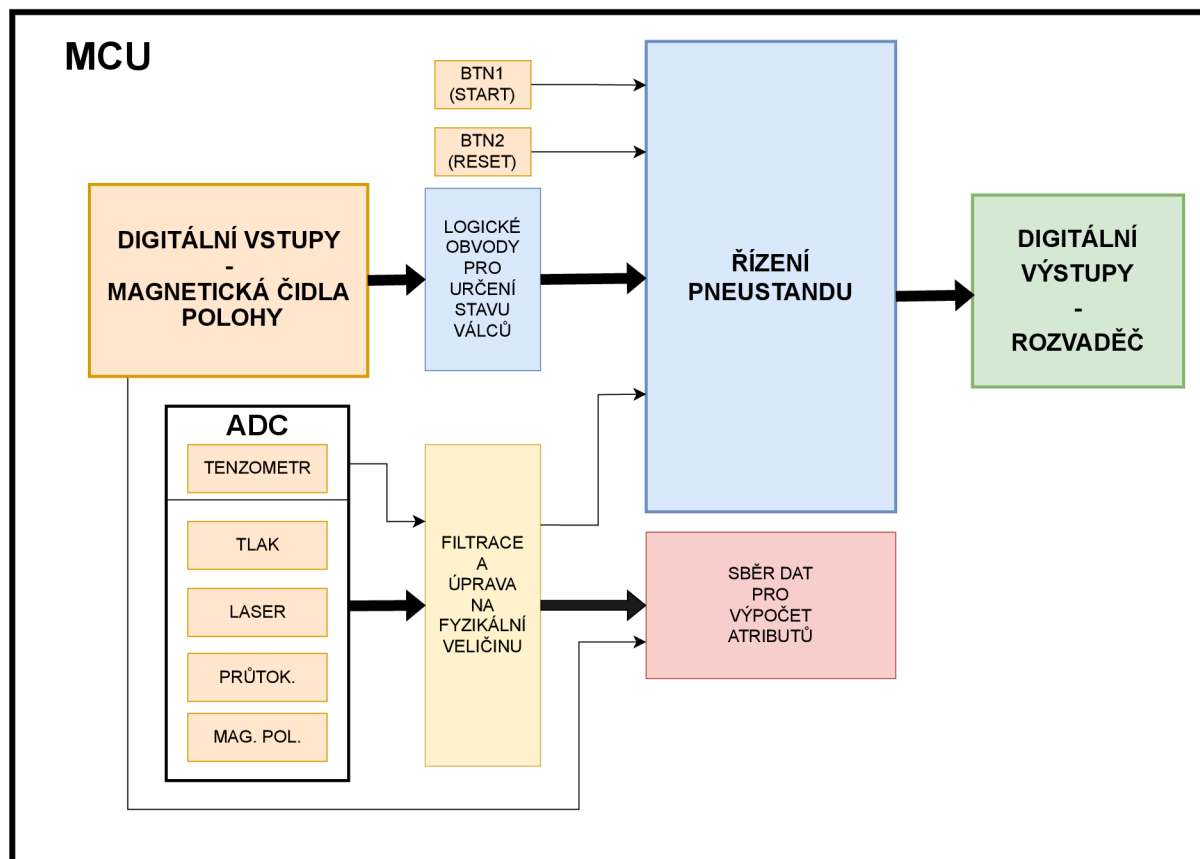
- 1x Timer
- 5x AD převodník
- 2x UART
- 6x digitální výstup
- 4x digitální vstup
- 2x digitální vstup - integrovaná tlačítka platformy

Zapojení opět vychází ze schéma v obr. 3.3, všechny potřebné úpravy pro řízení pomocí mikrokontroléru jsou popsány výše v kap. 4.1.2, na obr. 4.21 je schematicky znázorněno podrobnější zapojení všech komponent potřebných pro řízení pneustandu, jako jsou zdroje, mikrokontrolér, modul pro odesílání dat, elektronika pro úpravu signálu a relé modul pro spínání ventilů v rozvaděči.



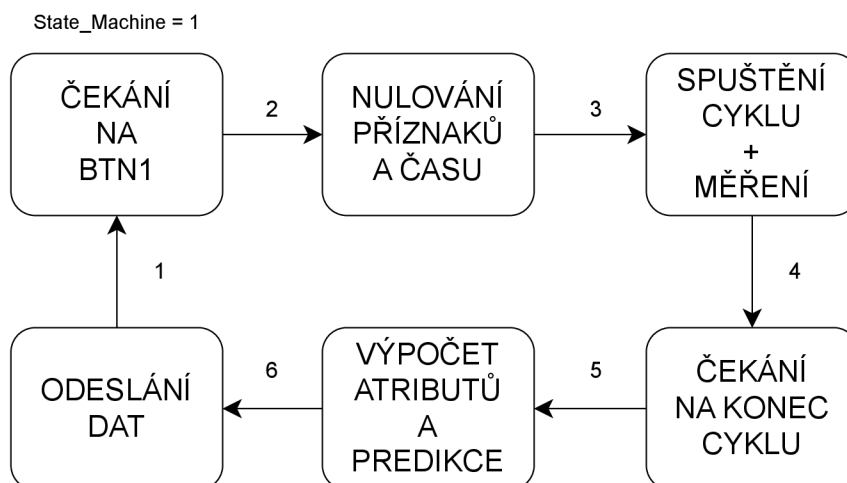
Obrázek 4.21: Pneustand - schéma komponent

Většina kódu byla opět naprogramována pomocí bloků a proměnných typu struktura. V přerušení od Timeru 1 byla umístěna funkce pro sběr dat pro výpočet atributů a řídicí algoritmus pneustandu, schéma funkce je znázorněno v obr. 4.22.



Obrázek 4.22: Pneustand - schéma řízení na MCU

Celé rozhraní řízení a odesílání dat bylo naprogramováno ve stavovém automatu podle schématu obr. 4.23. V případě potřeby (chyby systému řízení) je možné zmáčknutím druhého tlačítka (BTN 2) celý cyklus restartovat.



Obrázek 4.23: Pneustand - schéma stavového automatu

- Nejprve systém čeká na zmáčknutí tlačítka BTN 1.
- Dále jsou vynulovány bloky pro výpočet atributů a časovací proměnná.
- Provedení cyklu + měření veličin v definovaných časových úsecích.
- Výpočet atributů a predikce stavu.
- Odeslání dat na server.

Obdobně jako pro tepelnou komoru v kap. 4.1.5 byla knihovna pro predikci stavu vygenerována do jazyka C z modelu získaného pomocí Classification Learner app. Ze stejných důvodů došlo opět k přetypování dat vstupujících i vystupujících proměnných.

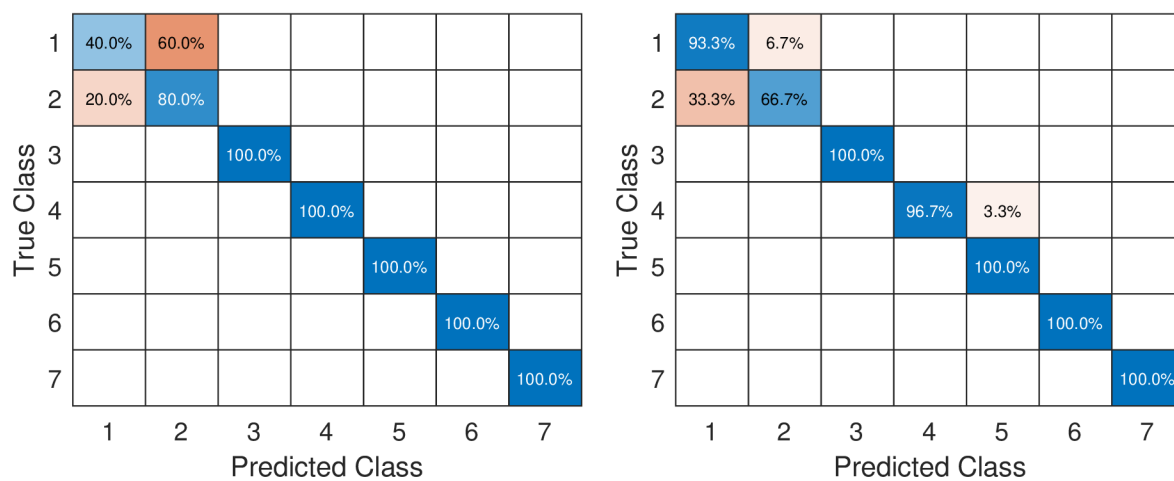
4.2.6 Experiment - ověření funkčnosti

Po natrénování klasifikátoru a implementaci řízení pomocí mikrokontroléru bylo zapotřebí ověřit, zda-li predikce stavů funguje správně. Toho bylo docíleno provozem pneustandu v jednotlivých poruchových stavech. Každý stav byl 30krát zopakován. Z těchto dat poté bylo možné sestavit matici zmatenosti predikce na reálném zařízení, ta je na obr. 4.24.

1	28	2					
2	10	20					
3			30				
4				29	1		
5					30		
6						30	
7							30
	1	2	3	4	5	6	7

Obrázek 4.24: Matice zmatenosti pneustandu - reálný provoz

Přesnost predikce na reálném zařízení dosáhla 93,8 %, vypočítána byla pomocí matice zmatenosti. Z obr. 4.24 je patrné, že identifikace stavů pomocí vybraných atributů v kap. 4.2.4 byla ve většině případů správná. Největší problém s identifikací byl ve stavu 1 a 2. Tyto stavy byly od sebe velmi málo odlišné, jelikož jediným rozdílem bylo 40 g ve váze jednotlivých kostek. Takový rozdíl byl při přesunu kostek pro pneumatické válce v podstatě zanedbatelný. Přesto jsme se je pokusili identifikovat, je u nich ve výsledcích není dosaženo tak vysoké úspěšnosti identifikace. Celkově se ale podařilo implementovat a otestovat algoritmy prediktivní údržby na tomto zařízení.



(a) Validační data (b) Reálný provoz
 Obrázek 4.25: Porovnání matic zmatenosti ověřovacího experimentu na pneustandu

5 Závěr

Tato práce se zabírala ověřením funkčnosti algoritmů prediktivní údržby na 2 zařízeních, které byly dostupné v Mechatronické laboratoři a také ověřením vhodnosti použití sítě NB-IoT pro její potřeby.

V rešeršní části jsou nejprve představeny různé technologie bezdrátového přenosu dat, které jsou vhodné především k využití v průmyslových aplikacích. Hlavním výstupem této části je v kap. 2.1.7 tabulka 2.1, kde jsou popisované technologie srovnány podle rychlosti přenosu, dosahu, podle pokrytí v ČR a počtu připojených zařízení, což může být využito při návrhu konkrétního zařízení pro IoT.

Pro využití IoT aplikací je nutné propojení zařízení se serverem, k čemuž slouží komunikační protokoly a v souvislosti s tím byly také představeny základní druhy protokolů. Dále pak je popsán samotný algoritmus vývoje aplikace prediktivní údržby včetně jeho jednotlivých kroků.

Praktická část práce byla rozdělena na 2 dílčí části a to konkrétně Serverová aplikace a Prediktivní údržba. V Serverové aplikaci je popsána celá struktura přenosu dat pomocí NB-IoT ze zařízení na server s následnou možností vizualizace dat uživateli. Nejprve je obecně popsáno řízení soustav pomocí mikrokontroléru, poté je popsán vybraný NB-IoT modul SIM7000E od společnosti SIMCom. K tomu je dále doplněn aplikační kód pro připojení modulu k NB-IoT síti a to, jakým způsobem je na modulu implementován protokol MQTT, který byl pro přenos dat vybrán v rámci rešeršní práce v kap. 2.2.6.

Dále jsou v kap. 3.3 představeny specifikace serveru, v tomto případě jednodeskového počítače Raspberry Pi 4B. Součástí je také vysvětlení, jakým způsobem je server a na něm používaný software zprovozněn. Konkrétně se jedná o služby MQTT Mosquitto broker, Node-Red, InfluxDB a vizualizační platformu Grafana, které dohromady s modulem a přenášenými daty pomocí NB-IoT zajišťují zobrazování měřených veličin uživateli a demonstují využití této sítě pro potřeby prediktivní údržby.

Zbytek praktické části práce byl věnován prediktivní údržbě, a to konkrétně aplikací na dva výukové modely, na které byly aplikován algoritmus prediktivní údržby a následně proběhlo ověření prediktivní údržby při řízení soustav mikrokontrolérem. Nejprve byl v kap. 4.1 zkoumán výukový model tepelné komory. Na něm byly navrženy různé umělé poruchové stavy, především různá omezení průtoku vzduchu komorou. Poté byly pomocí senzorů získána data k vývoji predikce stavu soustavy. To bylo provedeno především v softwaru Matlab pomocí doplňků kterými byly Simulink, Diagnostic Feature Designer a Classification Learner app. Pomocí těchto aplikací byla naměřená data zpracována a následně bylo možné natrénovat klasifikátor.

5 ZÁVĚR

Po připravení klasifikátoru bylo zařízení upraveno, aby mohlo být řízeno pomocí mikrokontroléru. Následně byla pomocí experimentu na samotném zařízení ověřena funkčnost klasifikátoru, který při vývoji prediktivní údržby dosahoval přesnosti klasifikace 91,8 %, a tím byl Fine Gaussian SVM. Ten byl implementovaný na mikrokontrolér a poté při provozu dosahoval přesnosti 88,7 %. Dle výsledků, které jsou v kap. 4.1.6, ale bylo zjištěno, že komora je velmi náchylná na okolní proudění vzduchu a teplotu, proto by bylo vhodné ji doplnit ještě o 1 termočlánek, který by měřil i teplotu okolí komory pro ještě přesnější predikci.

V kap. 4.2 byl představen výukový model pneustandu znázorňující výrobní linku s lisem. Pro něj byly opět navrženy umělé poruchové stavy. V tomto případě již ale došlo k větším úpravám soustavy, a to přidělení nastavitelného tlumiče k lisu, který simuloval různé zalisování součásti do podložky. Dále pak byl doplněn uzavírací ventil pro jeden pneumatický válec a dvojice akcelerometrů. Na takto upraveném zařízení následně proběhlo měření a vývoj prediktivní údržby v softwaru Matlab, jako tomu bylo u tepelné komory. V tomto případě ale bylo složité určení klasifikačních parametrů pro správnou funkci predikce. Proto bylo určení klasifikátorů provedeno pro různé skupiny stavů, které byly na sobě nějakým způsobem závislé. Po takto určených stavech v tab. 4.10 byl natrénován klasifikátor, ten ale při ověřování funkčnosti experimentem nebyl schopný správně identifikovat poruchové stavy 1 a 2, ve kterých byla zaměněna referenční kostka za jinou (těžší). Proto muselo být přistoupeno k úpravě výsledného klasifikátoru, jak bylo popsáno v kap. 2.3.6 v rešeršní části práce a proto byl klasifikátor Quadratic SVM nahrazen druhým nejlepším klasifikátorem (Medium Gaussian SVM) s přesností 88,6 %. Následně proběhlo opětovné ověření funkčnosti klasifikátoru pomocí experimentu, kde zmiňovaný klasifikátor dosáhl celkové úspěšnosti predikce 93,8 %.

Literatura

- [1] *Prediktivní údržba* [online]. [cit. 2023-05-26]. Dostupné z: <https://www2.deloitte.com/cz/cs/pages/deloitte-analytics/solutions/predictive-maintenance.html>
- [2] *Low Power Wireless Technologies for Your Future Device: Selection Guide* [online]. 8.9.2022 [cit. 2023-05-26]. Dostupné z: <https://promwad.com/news/low-power-wireless-technologies>
- [3] *What are LoRa and LoRaWAN?* [online]. [cit. 2023-05-13]. Dostupné z: <https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/>
- [4] *An In-depth look at LoRaWAN® Class A Devices* [online]. [cit. 2023-05-13]. Dostupné z: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lorawan-class-a-devices/>
- [5] *Přehled sítí pro IoT v Čechách* [online]. [cit. 2023-05-26]. Dostupné z: <https://eshop.sectron.cz/prehled-siti-pro-iot-v-cechach/a-6398/>
- [6] *Zigbee* In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2023-05-26]. Dostupné z: <https://en.wikipedia.org/wiki/Zigbee>
- [7] [online]. In: . [cit. 2023-05-26]. Dostupné z: <https://ars.els-cdn.com/content/image/3-s2.0-B9780123735805500545-f20-05-9780123735805.jpg>
- [8] *Introduction to the ZigBee Wireless Sensor and Control Network* [online]. 2.12.2009 [cit. 2023-05-26]. Dostupné z: <https://www.informit.com/articles/article.aspx?p=1409785&seqNum=4>
- [9] *IoT přichází i k nám!* [online]. In: . [cit. 2023-05-26]. Dostupné z: <https://eshop.sectron.cz/iot-prichazi-i-k-nam/a-6255/>
- [10] PAPANIKOLAOU, Kalliopi. *What Is LTE-M? Long Term Evolution for Machines Explained* [online]. In: . [cit. 2023-05-26]. Dostupné z: <https://www.emnify.com/iot-glossary/lte-m>
- [11] LOM, Michal, Ing. a Ondřej PŘIBYL. *Sítě pro internet věcí v České republice* [online]. In: . 6.11.2017 [cit. 2023-05-26]. Dostupné z: <https://elektro.tzb-info.cz/informacni-a-telekomunikacni-technologie/16519-site-pro-internet-veci-v-ceske-republice>
- [12] ROSENCRANCE, Linda. *Narrowband IoT (NB-IoT)* [online]. 04.2020 [cit. 2023-05-26]. Dostupné z:

- [https://www.techtarget.com/whatis/definition/narrowband-IoT-NB-IoT#:text=Narrowband%20IoT%20\(NB%2DIoT\)%20is%20a%20wireless%20internet%20of,three%20main%203GPP%20LPWAN%20standards](https://www.techtarget.com/whatis/definition/narrowband-IoT-NB-IoT#:text=Narrowband%20IoT%20(NB%2DIoT)%20is%20a%20wireless%20internet%20of,three%20main%203GPP%20LPWAN%20standards).
- [13] *Mobilní sítě GSM – mobilní sítě 2. generace* [online]. [cit. 2023-05-26]. Dostupné z: <https://publi.cz/books/236/03.html>
- [14] GILLIS, Alexander S., Dan JONES a Kevin BEAVER. *LTE (Long-Term Evolution)* [online]. [cit. 2023-05-26]. Dostupné z: [https://www.techtarget.com/searchmobilecomputing/definition/Long-Term-Evolution-LTE#:text=LTE%20\(Long%2DTerm%20Evolution\)%20is%20a%20fourth%2Dgeneration,%2Dgeneration%20\(3G\)%20technology](https://www.techtarget.com/searchmobilecomputing/definition/Long-Term-Evolution-LTE#:text=LTE%20(Long%2DTerm%20Evolution)%20is%20a%20fourth%2Dgeneration,%2Dgeneration%20(3G)%20technology).
- [15] BRZOBOHATÝ, Lukáš. *Modul pro měření teploty a vlhkosti s rozhraním Ethernet* [online]. 2012. Brno, [cit. 2023-05-26]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=118276
- [16] ROSENCRANCE, Linda, George LAWTON a Chuck MOOZAKIS. *UDP (User Datagram Protocol)* [online]. 2020. [cit. 2023-05-26]. Dostupné z: [https://searchnetworking.techtarget.com/definition/UDP-User-Datagram-Protocol#:text=UDP%20\(User%20Datagram%20Protocol\)%20is,provided%20by%20the%20receiving%20party](https://searchnetworking.techtarget.com/definition/UDP-User-Datagram-Protocol#:text=UDP%20(User%20Datagram%20Protocol)%20is,provided%20by%20the%20receiving%20party)
- [17] ZAPLETAL, Lukáš. *Protokol HTTP 1.1 pod lupou* [online]. 2001. [cit. 2023-05-26]. Dostupné z: <https://www.root.cz/clanky/protokol-http-1-1-pod-lupou/>
- [18] BRÁZDA, Libor. *Instalace, konfigurace a testování serverových služeb typu DHCP, FTP, VPN, NAT a SNMP v prostředí IPv4 a IPv6* [online]. 2012. Brno, 2020 [cit. 2023-05-26]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=56911
- [19] MALÝ, Martin. *Protokol MQTT: komunikační standard pro IoT* [online]. 2016 [cit. 2023-05-26]. Dostupné z: <https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/>
- [20] *Predictive Maintenance with MATLAB* [online]. [cit. 2023-05-26]. Dostupné z: <https://www.mathworks.com/content/dam/mathworks/ebook/gated/predictive-maintenance-ebook-all-chapters.pdf>
- [21] *Overcoming Four Common Obstacles to Predictive Maintenance with MATLAB and Simulink* [online]. [cit. 2023-05-26]. Dostupné z: <https://www.mathworks.com/content/dam/mathworks/white-paper/gated/predictive-maintenance-challenges-whitepaper.pdf>
- [22] DOBOSSY, B.; FORMÁNEK, M.; ŠTASTNÝ, P.; SPÁČIL, T. *Detection and Identification on Pneumatic Production Machine*. In *Modelling and Simulation for Autonomous Systems. Lecture Notes in Computer Science*. 1. Virtual, Online: SPRINGER INTERNATIONAL PUBLISHING AG, 2022. s. 39-60. ISBN: 978-3-030-98260-7. ISSN: 1611-3349.

- [23] VACULÍK, Karel. *Selekce příznaků pomocí nekorelovaných charakteristik*. Brno, 2013. Dostupné také z: <https://dspace.vutbr.cz/bitstream/handle/11012/24928/final-thesis.pdf?sequence=-1&isAllowed=y>. Bakalářská práce. Vysoké učení technické. Vedoucí práce Petr Honzík.
- [24] VERMA, Vikas. A comprehensive guide to Feature Selection using Wrapper methods in Python [online]. 24.10.2020 [cit. 2023-05-26]. Dostupné z: <https://dspace.vutbr.cz/bitstream/handle/11012/24928/final-thesis.pdf?sequence=1&isAllowed=y>
- [25] ABHIGYAN. *Feature Selection For Dimensionality Reduction(Embedded Method)* [online]. 20.5.2020 [cit. 2023-05-26]. Dostupné z: <https://medium.com/analytics-vidhya/feature-selection-for-dimensionality-reduction-embedded-method-e05c74014aa>
- [26] BOLÓN-CANEDO, Verónica a SÁNCHEZ-MAROÑO. Bolón-Canedo, Verónica & Sánchez-Marño, Noelia & Alonso-Betanzos, Amparo: *A review of feature selection methods on synthetic data*. Knowledge and Information Systems. 2022. Dostupné z: doi:10.1007/s10115012-0487-8
- [27] *Relief (feature selection)* [online]. [cit. 2023-05-26]. Dostupné z: [https://en.wikipedia.org/wiki/Relief_feature_selection\)#Relief_Algorithm](https://en.wikipedia.org/wiki/Relief_feature_selection)#Relief_Algorithm)
- [28] KREJSA, Jiří. *Support Vector Machines (SVM)*. 2021.
- [29] AYLIN TOKUC, A. *Why Feature Scaling in SVM?* [online]. 2023 [cit. 2023-05-26]. Dostupné z: <https://www.baeldung.com/cs/svm-feature-scaling>
- [30] MICROCHIP. *PIC32MX Family Data Sheet: 64/100-Pin General Purpose and USB 32-Bit Flash Microcontrollers*. Dostupné také z: https://ww1.microchip.com/downloads/en/DeviceDoc/PIC32MX_Datasheet_v2_61143B.pdf
- [31] *SIM7000E NB-IoT / Cat-M / EDGE / GPRS HAT for Raspberry Pi, GNSS, for Europe, Africa, Australia, Southeast Asia*. Waveshare [online]. [cit. 2023-05-05]. Dostupné z: <https://www.waveshare.com/sim7000e-nb-iot-hat.htm>
- [32] *SIMCOM WIRELESS. SIM7000 Series SIMCom LPWA Module* [online]. 2020. 2020, 3 s. [cit. 2023-05-12]. v: 2020.03. Dostupné z: <https://www.simcom.com/service-1074.html>
- [33] *SIMCOM WIRELESS SOLUTIONS LIMITED. SIM7000 Series_ AT Command Manual*. 2020.
- [34] *How to use test NB-IOT with SIM7000 modules?*. M2msupport.net [online]. [cit. 2023-05-12]. Dostupné z: <https://m2msupport.net/m2msupport/how-to-use-test-nb-iot-with-sim7000-modules/>
- [35] *SIMCOM WIRELESS SOLUTIONS LIMITED. SIM7000 Series MQTT(S)_Application Note*. 2020. Dostupné také z: <https://www.simcom.com/service-1192.html>

- [36] *Mikropočítač Raspberry Pi 4 Model B* [online]. [cit. 2023-05-26]. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
- [37] OXER, Jonathan. *Datalogging with MQTT, Node-RED, InfluxDB, and Grafana* [online]. 10.12.2020 [cit. 2023-05-26]. Dostupné z: <https://www.superhouse.tv/41-datalogging-with-mqtt-node-red-influxdb-and-grafana/>
- [38] AGNIHOTRI, Nikhil. *The top MQTT brokers of 2022* [online]. [cit. 2023-05-26]. Dostupné z: <https://www.engineersgarage.com/top-mqtt-brokers-2022/>
- [39] *Install Mosquitto MQTT Broker on Raspberry Pi* [online]. 10. 5. 2022 [cit. 2023-05-26]. Dostupné z: <https://randomnerdtutorials.com/how-to-install-mosquitto-broker-on-raspberry-pi/>
- [40] *InfluxDB: Introduction* [online]. [cit. 2023-05-26]. Dostupné z: <https://www.stackhero.io/en/services/InfluxDB/documentations/Introduction>
- [41] LUNDBERG, MATT. *What is Node-RED?* [online]. In: . [cit. 2023-05-26]. Dostupné z: https://www.impulse-embedded.co.uk/info/what-is-node_red.htm
- [42] SHIVANG. *What is Grafana? Why Use It? Everything You Should Know About It* [online]. [cit. 2023-05-26]. Dostupné z: <https://scaleyourapp.com/what-is-grafana-why-use-it-everything-you-should-know-about-it/>
- [43] GREGOR, TOMÁŠ. *Tepelná komora- výukový model*. Brno, 2013, 35 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Vedoucí práce Doc. Ing. RADEK VLACH, Ph.D.
- [44] *MF634-PCI Express Multifunkční vstupně-výstupní karta*. Humusoft.cz [online]. Praha 8, Humusoft, © 1991 – 2022 [cit. 2023-05-26]. Dostupné z: <https://www.humusoft.cz/datacq/mf634/>
- [45] *Code Generation and Classification Learner App*. The MathWorks, Inc. [online]. [cit. 2023-05-26]. Dostupné z: <https://www.mathworks.com/help/stats/code-generation-and-classification-learner-app.html>
- [46] JAVOREK, David. *Návrh a realizace pneumatického výukového modelu*. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/140211>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Jan Najman
- [47] *SMT-8M-A-PS-24V-E-2,5-OE*. Farnell.com [online]. Anglie, An Avnet Company, © 2022 [cit. 2023-05-26]. Dostupné z: <https://cz.farnell.com/festo/574335/proximity-sensor-magneto-resistive/dp/3437186>
- [48] *Sensors, Series SM6 R412010144*. Emerson.com [online]. Praha, Emerson Electric Co, © 2022 [cit. 2023-05-26]. Dostupné z: <https://www.emerson.com/cs-cz/catalog/aventics-sku-r412010144-en-gb>
- [49] *optoNCDT 1220*. Micro-epsilon.cz [online]. Germany, [cit. 2023-05-26]. Dostupné z: <https://www.micro-epsilon.cz/download/manuals/man-optoNCDT-1220-en.pdf>

- [50] *FX293X-100A-0100-L*. Farnell.com [online]. Anglie, An Av-net Company, © 2022 [cit. 2023-05-26]. Dostupné z: https://cz.mouser.com/datasheet/2/418/5/ENG_DS_FX29_A5-1609196.pdf
- [51] *SPAU-P10R-W-G18FD-L-PNLK-PNVBA-M8U*. Festo.com [online]. Taiwan, Festo Corporation, © 2022 [cit. 2023-05-26]. Dostupné z: https://www.festo.com/tw/en/a/8001231/?siteUidfox_tw&siteNameFesto+TW
- [52] *Průtokoměry SFAH*. Festo.com [online]. [cit. 2023-05-26]. Dostupné z: https://www.festo.com/cat/en-gb_gb/data/doc_CS/PDF/CZ/SFAH_CZ.PDF
- [53] *Válce dle norem DSBC, ISO15552*. Festo.com [online]. [cit. 2023-05-26]. Dostupné z: <https://www.festo.com/media/pim/653/D15000100152653.PDF>
- [54] *Series PRA and TRB*. Sotekshop.com [online]. [cit. 2023-05-26]. Dostupné z: https://www.sotekshop.com/res/foto_pdf/PDFGENERICI/b001/TRB.pdf
- [55] *Tlumič nárazu MA 150 M-B ACE* [online]. [cit. 2023-05-26]. Dostupné z: <https://cz.rs-online.com/web/p/tlumice-narazu/3663345>
- [56] *GY-61 DXL335 3-AXIS ACCELEROMETER MODULE* [online]. [cit. 2023-05-26]. Dostupné z: <https://www.robotpark.com/GY-61-DXL335-3-Axis-Accelerometer-Module>
- [57] *Relé modul s 8 kanály* [online]. [cit. 2023-05-26]. Dostupné z: <https://dratek.cz/docs/produkty/0/953/1550859050.pdf>

Seznam zkratek

LoRaWAN Long Range Wide Area Network

WPAN Wireless Personal Area Network

FFD Full Functional Device

RFD Reduced Functional Device

LTE-M Long Term Evolution Machine Type Communication

Vo-LTE Voice Over LTE

NB-IoT Narrowband Internet of Things

GSM Global System for Mobile Communications

TCP Transmission Control Protocol

UDP User Datagram Protocol

FTP File Transfer Protocol

HTTP Hypertext Transfer Protocol

MQTT Message Queuing Telemetry Transport

QoS Quality of Service

SSL Secure Sockets Layer

M2M Machine To Machine

DFD Diagnostic Feature Designer

SVM Support Vector Machines

AD převodník Analogově digitální převodník

UART Universal Asynchronous Receiver/Transmitter

RPi Raspberry Pi 4B (RPi)

PWM Pulse-width modulation

TČ Termočlánek

OZ Operační zesilovač

Seznam obrázků

2.1	Topologie sítě LoRaWan [5]	11
2.2	Topologie v síti ZigBee [7]	12
2.3	Topologie sítě NB-IoT [11]	14
2.4	Srovnání (shora) reaktivní, preventivní a prediktivní údržba [20]	20
2.5	Pracovní postup při implementaci prediktivní údržby [20]	20
2.6	Dělicí přímka zjištěná pomocí SVM pro 2D prostor [28]	24
2.7	Transformace dat pomocí kernelovské funkce [28]	24
3.1	Schéma serverové aplikace	25
3.2	Vývojová platforma (MECH)SOFT	26
3.3	Obecné schéma zapojení mikrokontroléru	26
3.4	Vývojová platforma s modulem SIMCom 7000E [31]	27
3.5	Raspberry PI 4B [36]	29
3.6	Schéma příjmu, zpracování a uložení zprávy v Node-Red	32
3.7	Ukázka interaktivního panelu pro tepelnou komoru v Grafaně	33
4.1	Tepelná komora [43]	34
4.2	Schéma tepelné komory schéma, 1 - termočlánky, 2 - ventilátor, 3 - tepelný zdroj	35
4.3	Regulace výstupní teploty komory v programu Simulink [43]	35
4.4	Krytky na otvory	36
4.5	Vytvořáření časových úseků v DFD	38
4.6	Matice zmatenosti pro klasifikaci Fine Gaussian SVM pro $T = 25$ s - trénovací data	40
4.7	Matice zmatenosti pro klasifikaci Fine Gaussian SVM pro $T = 25$ s - testovací data	40
4.8	Blokové schéma hardware	41
4.9	Blokové schéma softwaru na MCU	42
4.10	Matice zmatenosti - reálný provoz	43
4.11	Porovnání matic zmatenosti ověřovacího experimentu na tepelné komoře	43
4.12	Graf predikce - testování - stav 1	44
4.13	Pneustand [46]	44
4.14	Schéma pneustandu [46]	45
4.15	Přidělení tlumiče na pneustand	47
4.16	Přidělení akcelerometrů na písty	48
4.17	Přidělení relé modulu	48
4.18	Klasifikátor SVM - ohodnocení pro všechny stavy	50
4.19	Klasifikátor SVM - finální verze - trénovací data	52
4.20	Klasifikátor SVM - finální verze - validační data	52

4.21 Pneustand - schéma komponent	53
4.22 Pneustand - schéma řízení na MCU	54
4.23 Pneustand - schéma stavového automatu	54
4.24 Matice zmatenosti pneustandu - reálný provoz	55
4.25 Porovnání matic zmatenosti ověřovacího experimentu na pneustandu	56

Seznam tabulek

2.1	Porovnání sítí	16
2.2	Atributy dat pro časovou závislost dostupné v programu DFD	22
4.1	Poruchové stavy tepelné komory	36
4.2	Nejlepších 10 atributů dat po ohodnocení - $T = 250s$	38
4.3	Atributy dat po ohodnocení - $T = 25s$	39
4.4	Porovnání klasifikátorů pro periody výpočtu atributů	39
4.5	Senzorická část pneustandu	46
4.6	Specifikace pneumatických valců	46
4.7	Poruchové stavy pneustandu	49
4.8	Pořadí atributů dat po ohodnocení	50
4.9	Pořadí atributů dat po ohodnocení pro různé stavy	51
4.10	Finální výpis atributů pro klasifikaci	52

Seznam příloh

- modul_SIM7000E** – obsahuje soubory .txt s AT příkazy k jednotlivým protokolům
- pneustand** – obsahuje složku **matlab** obsahující soubory k prediktivní údržbě pneustandu v MATLABu, dále pak složku **mikrokontroler** obsahující projekt s kódy řízení pneustandu z MPLABu pro MCU
- tepelna_komora** – obsahuje složku **matlab** pro obsahující soubory k prediktivní údržbě tepelné komory v matlabu, dále pak složku **mikrokontroler** obsahující projekt s kódy pro její řízení z MPLABu pro MCU