



FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

KATEDRA  
KYBERNETIKY

## BAKALÁŘSKÁ PRÁCE

---

# Automatické ovládání žaluzií s využitím strojového učení

---

*Autor:*  
Vojtěch Breník

*Vedoucí práce:*  
Ing. Martin Bulín, M.Sc.

22. dubna 2022



## Prohlášení

I, Vojtěch Breník, declare that this thesis titled, „Automatické ovládání žaluzií s využitím strojového učení“ and the work presented in it are my own.  
I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed:

---

Date:

---



ZÁPADOČESKÁ UNIVERZITA

Fakulta aplikovaných věd

Katedra kybernetiky

## *Abstrakt*

Bakalářská práce

**Automatické ovládání žaluzií s využitím strojového učení**

Vojtěch Breník

Your abstract goes here...



## *Poděkování*

Your acknowledgements go here...

# Obsah

<b>Abstrakt</b>	<b>v</b>
<b>1 Úvod</b>	<b>1</b>
1.1 State of the Art . . . . .	1
1.2 Thesis Objectives . . . . .	1
1.3 Thesis Outline . . . . .	1
<b>2 Současný stav problematiky</b>	<b>3</b>
2.1 Neuronové sítě . . . . .	3
2.1.1 Aktivační funkce . . . . .	3
2.1.2 Dopředná neuronová síť . . . . .	3
2.1.3 Rekurentní neuronová síť, LSTM . . . . .	3
2.2 Komunikační protokoly . . . . .	3
2.2.1 Protokol HTTP . . . . .	3
2.2.2 Protokol WebSockets . . . . .	3
2.2.3 Protokol MQTT . . . . .	3
<b>3 Hardwarové komponenty</b>	<b>5</b>
3.1 Zařízení pro měření teploty vzduchu a intenzity osvětlení . . . . .	5
3.1.1 Zapojení . . . . .	6
3.1.2 Deska plošných spojů (DPS) . . . . .	7
3.1.3 Modul ESP-12E s MCU a WiFi . . . . .	8
3.1.4 Teplotní senzor DS18B20 . . . . .	8
3.1.5 Senzor intenzity osvětlení TSL2591 . . . . .	9
<b>4 Sběr dat a komunikace komponent</b>	<b>11</b>
4.1 Použití komunikačního protokolu MQTT . . . . .	12
4.2 Komunikace s pohonem žaluzie . . . . .	12
4.3 Využití služby OpenWeather . . . . .	14
4.4 Databáze . . . . .	14
4.5 Webový server pro komunikaci s GUI . . . . .	15
<b>5 Příznaky pro strojové učení</b>	<b>17</b>
<b>6 Regresory</b>	<b>21</b>
6.1 Regresor založený na pravidlech (If-else) . . . . .	21
6.2 Regresor využívající dopřednou neuronovou síť (FFNN) . . . . .	23
6.3 Regresor využívající rekurentní neuronovou síť (LSTM) . . . . .	23
6.4 Příprava dat pro strojové učení . . . . .	24
<b>7 Grafické uživatelské rozhraní a retraining</b>	<b>25</b>
7.1 Stránka Data . . . . .	25

7.2	Stránka Simulator . . . . .	26
7.3	Stránka Control . . . . .	27
7.4	Stránka Live . . . . .	27
7.5	Retraining neuronových sítí regresorů . . . . .	28
<b>8</b>	<b>Vyhodnocení</b>	<b>29</b>
8.1	Referenční data . . . . .	29
8.2	Vliv příznaků na predikci . . . . .	29
8.3	Porovnání regresorů . . . . .	29
8.4	Vyhodnocení predikce jednotlivých regresorů na vybraných datech . . . . .	29
8.5	Vyhodnocení vlivu retraingu . . . . .	29
<b>9</b>	<b>Možnosti nasazení systému</b>	<b>31</b>
<b>10</b>	<b>Diskuze</b>	<b>33</b>
10.1	Recapitulation of Methods . . . . .	33
10.2	Shrnutí výsledků . . . . .	33
<b>11</b>	<b>Závěr</b>	<b>35</b>
11.1	Future Work . . . . .	35
	<b>Bibliografie</b>	<b>37</b>
	<b>A1 Structure of the Workspace</b>	<b>39</b>

# Seznam obrázků

3.1	Skutečné umístění měřicího zařízení . . . . .	6
3.2	Schéma zapojení měřicích zařízení . . . . .	7
3.3	Návrh DPS měřicích zařízení . . . . .	8
4.1	Schéma komunikace komponent . . . . .	11
4.2	Schéma endpointů a přenášených struktur . . . . .	16
7.1	Záhlaví GUI . . . . .	25
7.2	Legenda tabulky s daty . . . . .	26
7.3	Vizualizace výstupu regresoru . . . . .	26

# Seznam tabulek

4.1	Příznaky a jejich MQTT téma	12
5.1	Příznaky pro automatické ovládání žaluzie	19
6.1	Prohledávání parametry dopředné neuronové sítě	23
6.2	Prohledávání parametry rekurentní neuronové sítě	24



# Acronyms

**API** aplikační rozhraní. 12

~~atp.~~ a tak podobně. 6, 14, 22, 23

**DPS** deska plošných spojů. viii, x, 5–7

**FFNN** feed-forward neural network. 

**GUI** Graphical User Interface. viii, 5, 13–15, 25, 27

**LSTM** long short-term memory. 23

**MCU** microcontroller unit. viii, 6–8

**MSE** střední kvadratická odchylka (mean squared error). 23, 24

**NS** neuronová síť. 21, 24

**URL** uniform resource locator. 13

**ČHMÚ** Český hydrometeorologický ústav. 14



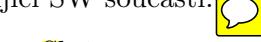


# 3 Hardwarové komponenty



K provozování určitých součástí systému je nutné použít fyzické technické vybavení, označované také jako hardware. Tato kapitola popisuje jeho funkční celky složené z jednotlivých použitých součástí a zmiňuje využití harwaru, které je blíže specifikováno v příslušných kapitolách (kapitoly 4 a 7).

K měření příznakových veličin intenzity osvětlení a teploty vzduchu ve vnějším a vnitřním prostředí se využívá 2 desek plošných spojů (DPS) vlastní konstrukce osazených modulem ESP-12E od společnosti Ai-Thinker (sekce 3.1). Na jednodeskovém počítači Raspberry Pi 2B, označme ho RPi1, běží následující SW součásti:



- Skript mqttDataCollector.py, který pomocí MQTT komunikuje s ostatními uzly a sbírá z nich data.
- Skript broker2mongo.py, který sesbíraná data ukládá do databáze MongoDB.
- MQTT broker<sup>1</sup> Mosquitto, ke kterému jsou připojeny všechny zdroje dat.

Posledním využívaným celkem je workstation označovaná jako KKY-PC, která:

- Hostuje databázový server MongoDB
- Hostuje webový server vlastního grafického uživatelského rozhraní (GUI).
- Provozuje vlastní REST a WebSocket backend implementovaný ve frameworku Tornado.
- Sloužila k trénování neuronových sítí regresorů.

Jedná se o výkonný osobní počítač s 6 jádrovým 64 bitovým procesorem Intel Core i7-7800X, 64 GiB operační paměti a grafickou kartou Nvidia GTX 1080. Pohání ji operační systém Ubuntu 20.04.3 a po dobu vývoje běžela na Katedře kybernetiky Západočeské univerzity v Plzni.

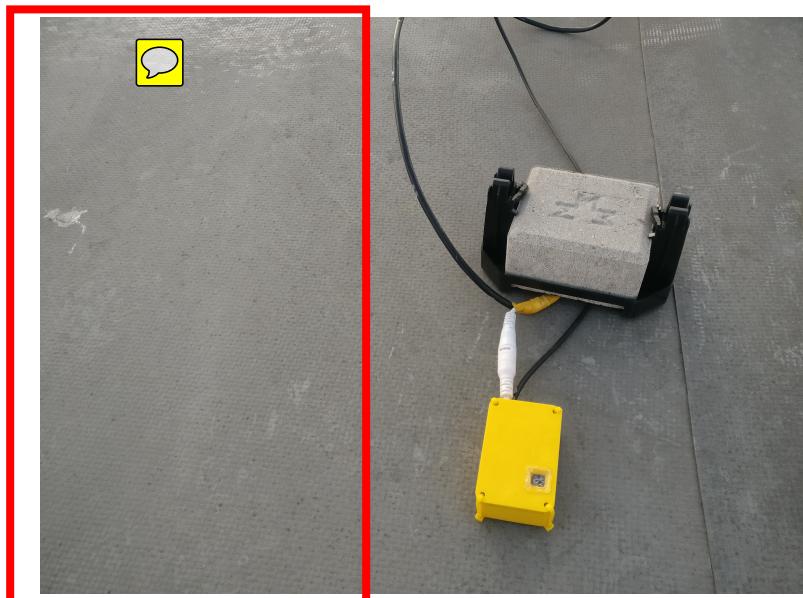
## 3.1 Zařízení pro měření teploty vzduchu a intenzity osvětlení



Čtyři z příznaků pro regresory tvoří teplota vzduchu a intenzita osvětlení, obojí v interiéru i exteriéru a je tedy nutné obě veličiny měřit a jejich hodnoty periodicky a také v případě zásahu uživatelem předávat dalším součástem systému. K tomu slouží již v úvodu této kapitoly zmíněné, osazené DPS.

---

<sup>1</sup>MQTT broker je centrální uzel, který zprostředkovává přenos zpráv od klienta, který zprávu odesal, ke klientům, kteří se přihlásili k jejich odběru. Více v sekci 4.1



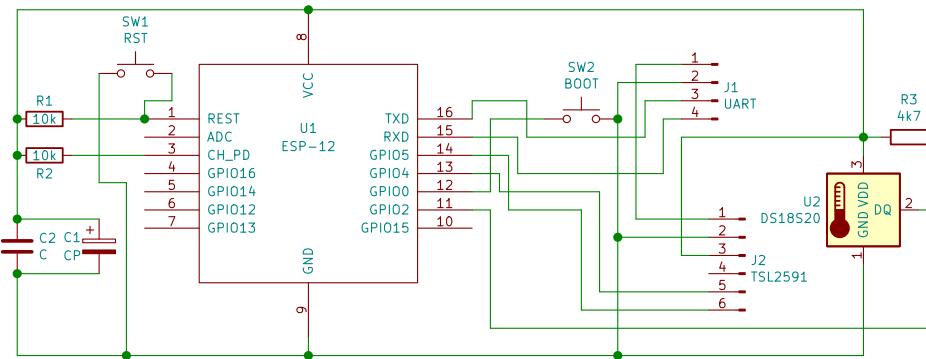
OBRÁZEK 3.1: Skutečné umístění měřicího zařízení na střeše rodinného domu.

Jedna z nich je umístěna v žaluzií zatemňované místnosti na polici tak, aby na připojené senzory nedopadalo přímé sluneční světlo, druhá pak na střeše domu, ve kterém je tato místnost, uvnitř krabičky s okénkem tak, aby byla elektronika krytá před povětrnostními vlivy, ale na senzor intenzity osvětlení i tak dopadalo světlo. Voděodolnost krabičky zajišťuje přišroubované víko se silikonovým těsněním a silikonem utěsněné kabelové průchody. Umístění na střeše bylo zvoleno tak, aby senzor intenzity osvětlení nebyl zastíněn dalšími nesouvisejícími předměty na střeše, jako jsou antény, jejich stožáry, komíny atp. Teplotní senzor vyvedený na kabelu o délce 0,75 m je umístěn za severní hranou střechy, kde je krytý před dopadajícím slunečním zářením. Fotografie skutečného umístění zařízení je na obrázku 3.1.

Hlavními součástmi těchto zařízení, které jsou podrobněji popsány dále, jsou: DPS, modul ESP-12E s MCU ESP8266, modul se senzorem intenzity osvětlení TSL2591 a vodotěsný senzor teploty DS18B20 na kabelu. Na desce je dále přítomno 5 pasivních součástek, z toho 3 rezistory a 2 kondenzátory, a 2 tlačítka. Napájeny jsou síťovými zdroji stejnosměrného proudu o velikosti až 0,5 A a napětí 5 V.

### 3.1.1 Zapojení

Zapojení vychází z dokumentace součástek a jeho schéma je na obrázku 3.2. Do paměti MCU je nutné nahrát firmware, nejjednodušší možnost je využít vyvedeného rozhraní UART na pinech 15 a 16, dále k tomu slouží tlačítko SW2, které připojí GPIO0 na 0 V a přepne tak MCU do režimu nahrávání firmwaru. Společně s napájecími piny je tedy toto rozhraní vyvedeno z desky na konektoru J1 se 4 piny. Tlačítko SW1 slouží k propojení REST pinu (1) MCU a 0 V, což vyvolá jeho reset. Jinak je přes pull-up rezistor R1 připojen na napájecí napětí, aby se předešlo náhodnému resetování. Rezistor R2 pak přivádí napětí na CH\_PD pin, který tak uvede do chodu interní regulátory napětí pro procesor. Kondenzátory C1 a C2 slouží k vyhlazení napájecího napětí, omezení rušení a překlenutí odběrových špiček. Pro připojení senzoru



OBRÁZEK 3.2: Schéma zapojení zařízení, která měří intenzitu osvětlení pomocí senzoru TSL2591 a teplotu vzduchu pomocí senzoru DS18B20. Hodnoty se přenáší přes WiFi.

teploty se využívá sběrnice 1-Wire, její datový vodič je připojen jednako na GPIO2, jednak přes rezistor R3 na napájecí napětí jako pull-up. Napájecí vodiče sběrnice jsou přímo spojené s napájecími vodiči MCU.

Na GPIO4 a GPIO5 jsou zapojeny po řadě datový a hodinový vodič sběrnice I<sup>2</sup>C, která slouží ke komunikaci se senzorem intenzity osvětlení. Ten dále slouží jako stabilizovaný zdroj napětí pro celé zařízení, součástí modulu je totiž lineární snižující zdroj napětí o velikosti 3,3 V.

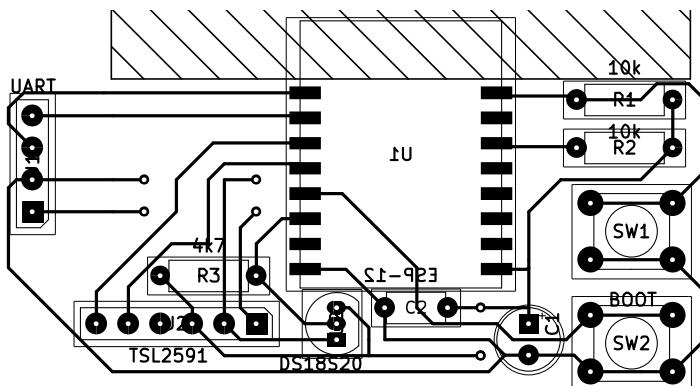
### 3.1.2 Deska plošných spojů (DPS)

DPS elektricky propjuje ostatní komponenty těchto zařízení a byla vyrobena po domácku. V opensource programu pro návrh elektronických zařízení KiCad byl sestaven obvod a návrh DPS (obrázek 3.3), jenž byl pomocí kancelářské tiskárny překlopeně vytištěn na běžný papír v měřítku 1:1.

Z jednostranně poměděného laminátu (cuprexit) byla pákovými nůžkami vystřížena deska o příslušných rozměrech dle návrhu, přebroušena brusným papírem a byla zabalena do papíru s návrhem tak, aby strana s mědí překryvala celou oblast návrhu, který byl ponechán na vnější straně. Kladivem a důlčíkem se do mědi poklepem na důlčík vyznačily budoucí díry pro jednotlivé součástky (v návrhu černá mezikruží, ze kterých mohou vést cesty) a krajní body pro osazení modulu ESP-12E (v návrhu černé obdélníky, ze kterých mohou vést cesty). Takové značky slouží jednak k orientaci při zakreslování vodivých cest, ale také při následném vrtání jako vodítka pro vrták.

Po vyznačení všech děr byla deska vyňata z papíru a lihovým fixem se zakresly pájecí body součástek (černá mezikruží a obdélníky) a cesty mezi nimi podle papírové předlohy (nejtenčí linie jsou ohrazení součástek a jejich pouzder a do desky se nezakreslují, protože nemají tvořit vodivé spojení). Pájecí body pro modul U1 (tedy ESP-12E) byly zakresleny pomocí samotného modulu použitého jako šablony zarovnané s důlčíkem vyznačenými krajními body.

Přebytečná měď, která nebyla překryta barvou, se vyleptala při pokojové teplotě v roztoku chloridu železitého pro leptání plošných spojů za 25 minut



OBRÁZEK 3.3: Návrh desky plošných spojů zařízení, která měří intenzitu osvětlení pomocí senzoru TSL2591 a teplotu vzduchu pomocí senzoru DS18B20. Hodnoty se přenáší přes WiFi. Šrafováná oblast je ochranná zóna antény.

a na desce tak zůstaly jen potřebné vodivé cesty. Leptání proběhlo tak, že se deska položila na hladinu roztoku nalitého do plastové leptací vaničky. Díky tomu mohly produkty chemické reakce klesat ke dnu a uvolnit prostor ještě nepoužitému roztoku. Deska byla několikrát z lázně vyjmuta a vizuálně zkонтrolována, jestli barva nebyla smyta a jestli je ještě potřeba leptat dále. Když už na desce zbývaly jen části překryté barvou, tedy cesty a pájecí body, se barva očistila technickým lihem pomocí vatového polštářku a do desky se vyvrtaly otvory pro součástky. Nakonec byla deska znova přebroušena a opatřena pájitelným ochranným lakem (v tomto případě kalafunou rozpuštěnou v acetolu). Součástky byly osazeny na příslušná místa dle návrhu desky a schématu.

### 3.1.3 Modul ESP-12E s MCU a WiFi

Modul ESP-12E vyvíjený společností **Ai-thinker Team** obsahuje čip ESP8266, který integruje kompletní řešení WiFi, 32 bitový procesor *Tensilica L106 Diamond Series*, taktovaný na 80 MHz, s vestavěnou SRAM a 16 univerzálních vstupně-výstupních pinů. Další součástí modulu je flash paměť připojená přes rozhraní SPI, na kterou se kladá program. **ESP-12E WiFi Module 2015** V popisovaném měřicím zařízení slouží ke zpracování dat naměřených pomocí senzorů teploty a intenzity osvětlení a jejich přenosu do dalších součástí systému.

### 3.1.4 Teplotní senzor DS18B20

DS18B20 je digitální teploměr s volitelnou přesností 9 – 12 bitů, který svá měření poskytuje po sběrnici 1-Wire, k jeho provozu tedy postačuje pouze datový a zemnící vodič. Volitelně lze také připojit napájecí napětí o velikosti 3 – 5,5 V třetím vodičem. Na stejně sběrnici může být připojeno více teploměrů DS18B20, protože každý z nich má z výroby přidělené unikátní sériové číslo. Je dostupný v pouzdrech TO92 a SOP8 a naleze využití ve stavebnictví při řízení vytápění, ventilace nebo klimatizace, teplot v budovách, v průmyslu při měření a monitorování teploty zařízení nebo procesů a jejich řízení. (**DS18B20 – Programmable Resolution 1-Wire Digital Thermometer 2019**) V měřicím zařízení poskytuje data o teplotě vzduchu.



### 3.1.5 Senzor intenzity osvětlení TSL2591

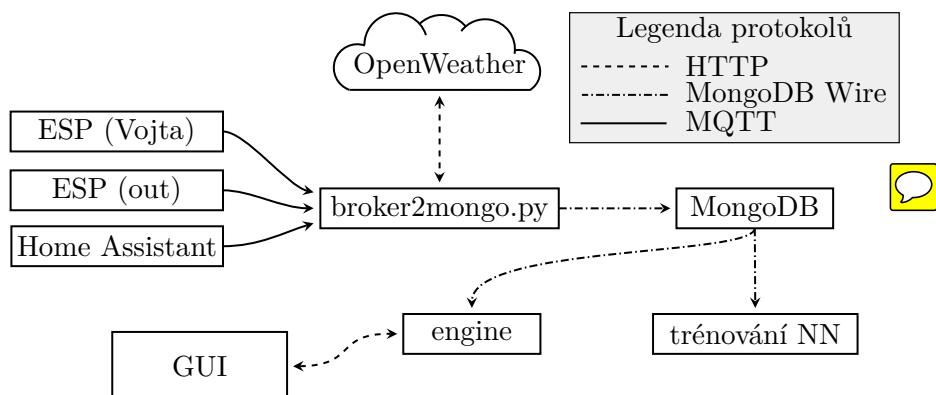
TSL2591 je digitální senzor intenzity osvětlení, který pracuje ve viditelném a infračerveném spektru. Parametry měření, kterými jsou zesílení a čas integrace, jsou volitelné a je tak možné měřit intenzitu osvětlení od 188  $\mu$ lux do 88000 lux. Měření lze číst po standardní sběrnici I<sup>2</sup>C, která využívá dvou vodičů - datový vodič a vodič s hodinovým signálem. Použitý modul od firmy Adafruit s tímto integrovaným obvodem má také vlastní lineární regulátor napětí MIC5225 proudově zatížitelný až do 150 mA. ([Adafruit TSL2591 High Dynamic Range Digital Light Sensor 2014](#)) Proto byl kromě měření intenzity osvětlení interiéru i exteriéru využit také pro napájení celého zařízení z rozšířených 5 V USB síťových zdrojů (např. adaptéry pro nabíjení spotřební elektroniky).





## 4 Sběr dat a komunikace komponent

Použitým algoritmem strojového učení je **MQTT** dodat data sestávající z počtu hodnot příznaků (více v kapitole 5) a odpovídajících stavů žaluzí (výška vytažení a sklon lamer) v daných časových okamžicích. Tato kapitola se zabývá metodami sběru dat v jednotlivých součástech systému, přenosem z nich a jejich vzájemnou komunikací.



OBRÁZEK 4.1: Schéma komunikace komponent. Obrázek znázorňuje vybrané komponenty systému automatického řízení žaluzí, tok dat mezi nimi a komunikační protokoly. Plnou šipkou jsou propojené komponenty komunikující pomocí MQTT, přerušovanou pomocí HTTP a čerchovanou ty, které ke komunikaci využívají protokol pro komunikaci s databázovým serverem MongoDB Wire



Data se získávají ze 3 hlavních zdrojů na základě požadavku zasláného skriptem `mqttDataCollector.py`. Stav žaluzie a informaci, zda je uživatel přítomen v domácnosti<sup>1</sup> odesílá systém domácí automatizace *Home Assistant*, měřené veličiny (teplota a intenzita osvětlení) uvnitř a venku pak příslušná zařízení<sup>2</sup> a ostatní informace o počasí<sup>3</sup> se získávají přes OpenWeather. Data získaná z komponent se pak v jedné společné zprávě odesírají přes MQTT skriptu `mqtt2mongo.py`, který je uloží do databáze. Na obrázku 4.1 je přehled komponent, které se podílejí na sběru dat, a jejich komunikace včetně používaných protokolů.

<sup>1</sup>příznaky position, tilt a home

<sup>2</sup>příznaky lum\_in, lum\_out, temp\_in a temp\_out

<sup>3</sup>rychllosť (owm\_wind\_speed) a směr (owm\_wind\_heading) větru, předpověď teploty na  $x = 1, 2, 3$  h dopředu (owm\_temp\_xh) a předpověď nejvyšší denní teploty (owm\_temp\_max)

## 4.1 Použití komunikačního protokolu MQTT

Protokol MQTT (sekce 2.2.3) se využívá pro veškerou komunikaci součástí při sběru dat. Centrální uzel se nazývá broker a jedná se o software spuštěný na RPi1 (kapitola 3) na výchozím TCP portu 1883. Připojují se k němu všechny komponenty, které využívají MQTT. Po připojení mohou publikovat zprávy do hierarchicky uspořádaných témat a přihlašovat se k jejich odběru. Broker pak zajistí doručení zpráv publikovaných v určitém tématu klientům, kteří jsou přihlášeni k jeho odběru.

Při sběru dat se periodicky nebo na základě změny stavu žaluzie odešle do tématu smartblinds/command zpráva ve formátu JSON, která obsahuje pod klíčem „command“ hodnotu „request\_values“. K odběru tohoto tématu jsou přihlášena obě zařízení s ESP8266 (sekce 3.1) i systém domácí automatizace a po jejím přijetí odešlou aktuální hodnoty jimi sledovaných příznaků do odpovídajících témat dle tabulky 4.1. Všechna tato téma odebírá skript mqttDataCollector.py, který hodnoty příznaků společně s časovou známkou odesílá v jedné zprávě do tématu smartblinds/data. Tato zpráva dále obsahuje informaci o tom, zda je aktivní testovací režim využívaný při vývoji. Téma smartblinds/data odebírá skript broker-2mongo.py popsaný v sekci 4.4. Ten všechna takto přenesená data ukládá do databáze k pozdějšímu použití.

Příznak nebo stav	Téma
temp_inside	smartblinds/temp/Vojta
temp_outside	smartblinds/temp/out
lum_inside	smartblinds/lux/Vojta
lum_outside	smartblinds/lux/out
home	smartblinds/presence/Vojta
position	smartblinds/position/Vojta
tilt	smartblinds/tilt/Vojta

TABULKA 4.1: Příznaky a jejich MQTT téma. Tabulka zahrnuje názvy příznaků a odpovídající téma pro odesílání jejich naměřených hodnot.

## 4.2 Komunikace s pohonem žaluzie

S pohonem žaluzie komunikuje centrální jednotka Tahoma, dodávaná výrobcem žaluzie, pomocí bezdrátového proprietárního protokolu *io*. Tato jednotka je dále připojená do internetu a je možné s ní komunikovat prostřednictvím aplikativní rozhraní (API), které nabízí výrobce. Komunikace s API je šifrovaná a probíhá pomocí protokolu HTTPS zasíláním požadavků na *endpoints* serverů výrobce dostupné na adrese <https://api.somfy.com/api/v1/>. Použitý systém domácí automatizace i backend vyvýjeného systému používají ke komunikaci s API knihovnu Pymfy, která mapuje akce pro *endpoint* ipulaci se žaluzí a její stav na metody a proměnné *objektu* v jazyce Python, který je

tak modelem žaluzie. Dále zpřístupňuje některé obecnější metody pro práci s API jako je například zjištění všech montáží zařízení, zjištění všech dostupných zařízení v rámci konkrétní montáže atd.

Jednotlivé požadavky se autorizují na základě krátkodobého tokenu s platností 1 h. Pokud vyprší jeho platnost, je nutné pomocí obnovovacího tokenu ze serveru získat nový a přikládat ho k budoucím požadavkům. Oba tokeny se získávají pomocí OAuth2 metodou „Authorization Code Grant“, z důvodu odlišnosti od dokumentace v implementaci Somfy se ale ani po kontaktování technické podpory nepodařilo získat nové přístupové údaje k API a tak muselo být využito nedokumentovaného postupu k jejich získání. Přestože jsou tokeny v obou systémech odvozené od stejných přístupových údajů, zdají se být nezávislé (včetně kvóty na četnost požadavků) a na funkčnost to tedy nemá vliv. Hledání tohoto postupu se zdálo být časově nákladné a proto se ke zjišťování aktuálního stavu žaluzí využívá právě systém domácí automatizace.

Běžně je nutné si na webových stránkách na adrese <https://developer.somfy.com> vytvořit tzv. aplikaci pod uživatelským účtem, ke kterému je technikem při instalaci žaluzí přiřazena konkrétní montáž. Na základě zadaného názvu, *callback URL*<sup>1</sup> a popisu aplikace se získájí dva řetězce: *Consumer key* a *Consumer secret*.(*Somfy Open API Developer Portal / APIs & Documentation*, 2018) Ty se pak společně s *callback URL*<sup>1</sup> předají konstruktoru objektu, který v rámci knihovny Pymfy reprezentuje API. (Etienne, 2021)

Takto vytvořené *Consumer key* a *Consumer secret* se však nedařilo použít, server Somfy je totiž zamítal. Byla tedy kontaktována technická podpora, ale ani po několika týdnech se nedostavila odpověď. Mezitím pokračovaly pokusy o získání tokenu jinak. Pro systém domácí automatizace, který se se žaluziemi již používal, existovaly tyto údaje a ukázalo se, že jsou funkční. Místo správné *callback URL* se tedy do konstruktoru zadala ta, která příslušela k aplikaci ve vývojářském portálu, a byl zahájen proces získání tokenů. Po přesměrování zpět se v adresním rádku prohlížeče ručně změnila adresa tak, aby odpovídala endpointu vytvořenému k získávání a ukládání tokenů v rámci backendu. Pokud by bylo možné postupovat standardním způsobem, po přihlášení na stránkách Somfy by byl prohlížeč přesměrován právě na tuto adresu. Tokeny se tak uložily do souboru `somfycache` pro pozdější použití.

Data a služby, které API poskytuje jsou využívány ke dvěma účelům. Jedenak systém domácí automatizace každou minutu kontroluje aktuální stav žaluzí a pokud se změní, odešle novou hodnotu přes MQTT a spustí tak posloupnost akcí (popsanou v sekci 4.1), které vedou k vytvoření nového záznamu v databázi, jednak stránka „Control“ v GUI umožňuje zobrazit aktuální stav žaluzí a zadávat příkazy k jeho změně. Samotnou komunikaci s API zajišťuje backend systému, se kterým se komunikuje přes protokol WebSockets (sekce 2.2.2). Po připojení alespoň jednoho klienta se každých 15 s zjistí stav žaluzie a připojeným prohlížečům se odešle zpráva ve formátu JSON. Její struktura je uvedená v úryvku kódu 4.1. Naopak po přijetí zprávy backendem se v závislosti na jejím obsahu odešle požadavek API

<sup>1</sup>Návratová URL v rámci využití aplikace, na kterou se přesměruje webový prohlížeč uživatele po úspěšné autorizaci, prostřednictvím parametrů se aplikaci předá kód, na základě kterého může získat tokeny

na změnu stavu žaluzie. Zpráva má stejnou strukturu jako zpráva v úryvku kódu 4.1, ale obsahuje jen jeden z klíčů. Navíc může obsahovat speciální klíč `testing` s hodnotou datového typu `boolean`<sup>4</sup>, pomocí kterého lze měnit příznak `testing` dat ukládaných do databáze, který slouží k rozlišení akcí vyvolaných při vývoji v rámci testování a skutečných akcí, které má systém napodobovat. Zpráva o změně režimu se v tomto případě odešle přes MQTT skriptu `mqttDataCollector.py`, který příznak `zaznamenává` k datům. Zprávy se odesílají na základě interakce uživatele s ovládacími prvky GUI.

```
{
    "position": int
    "tilt": int
}
```

ÚRYVEK KÓDU 4.1: Struktura zprávy o změně stavu žaluzie, která je všem klientům zasílána každých 15 s. `position` označuje výšku vytažení žaluzie (0 – zavřeno, 100 – otevřeno), obdobně hodnota `tilt` vyjadřuje naklopení lamel.

### 4.3 Využití služby OpenWeather

Skript `mqttDataCollector.py` kromě měřených příznaků a stavu žaluzie zajišťuje také odhad počasí a jeho předpověď. Tato data poskytuje společnost OpenWeather prostřednictvím svého API, které má několik možností využití. Z důvodu jednoduchosti byla pro přístup k API zvolena knihovna PyOWM, která používá variantu „One-Call“. Na základě jednoho požadavku se tak získá aktuální počasí, předpověď po minutách na následující hodinu, předpověď po hodinách na následující 2 dny, předpověď po dnech na následující týden, výstrahy vydané ČHMÚ a historická data z posledních 5 dnů, vše je pak přístupné pomocí atributů a metod objektu v Pythonu. K požadavku je vždy připojen klíč, který byl bezplatně získán po registraci na webových stránkách <https://openweathermap.org>. Limity četnosti požadavků stanovené poskytovatelem jsou vyšší než požadavek za 5 minut. Jako příznaky se využívá kódů počasí (vyjadřuje jeho shrnutí – např. slunečno, polojasno, dešt, jasná noc atp.), rychlosti a směru větru, předpovědi teploty na následující 3 hodiny a předpovědi nejvyšší denní teploty.

### 4.4 Databáze

Data, která se sbírají v pětiminutových intervalech nebo na základě změny stavu žaluzie pomocí skriptu `mqttDataCollector.py`, ukládá skript `broker2mongo.py` do databáze MongoDB [MongoDB Documentation 2021](#). Jednotlivé vzorky jsou získávány z MQTT zpráv zasílaných do tématu `smart-blinds/data` a ukládají se ve stejné podobě jako příchozí zpráva. Struktura je uvedena v úryvku kódu 4.2.

```
{
    "timestamp": float,
    "testing": boolean,
    "periodical": boolean,
    "features": {
        "year_day": int,
```

<sup>4</sup>`boolean` je datový typ, který vyjadřuje pravdivostní hodnotu.

```

        "week_day": int,
        "day_secs": int,
        "home": boolean,
        "temp_in": float,
        "temp_out": float,
        "lum_in": float,
        "lum_out": float,
        "owm_temp_max": float,
        "owm_temp_1h": float,
        "owm_temp_2h": float,
        "owm_temp_3h": float,
        "owm_code": int,
        "owm_wind_speed": float,
        "owm_wind_heading": float
    },
    "targets": {
        "position": int,
        "tilt": int
    }
}

```

ÚRYVEK KÓDU 4.2: Struktura vzorku dat uloženého jako záznam v databázi MongoDB. Obsahuje hodnoty příznaků i stav žaluzí v okamžiku jeho pořízení a časovou známku. Místo hodnot jsou zde uvedeny jejich datové typy (float - číslo s plovoucí řádovou čárkou, boolean - pravdivostní hodnota, int - celé číslo)

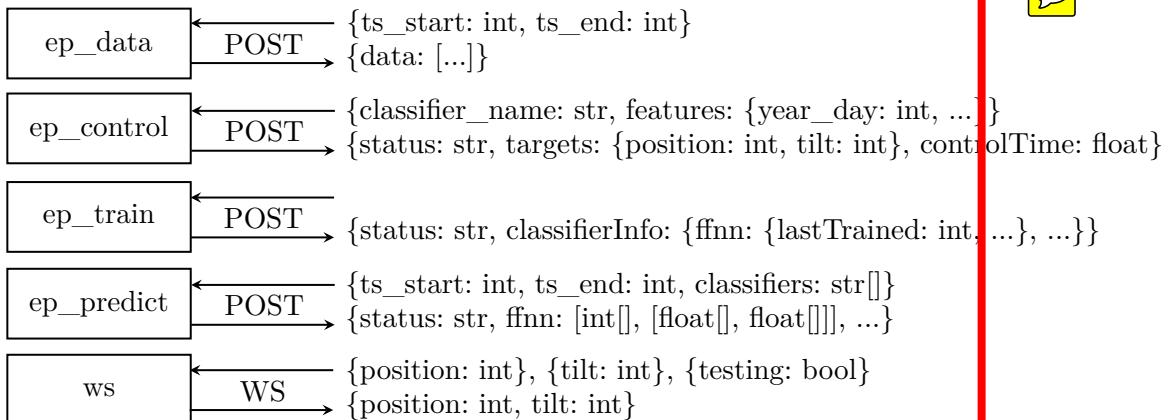


Uložená data se využívají při strojovém učení modelů, ve vizualizaci dat na stránce „Live“ v GUI a v porovnání skutečného řízení a řízení jednotlivých regresorů tamtéž. Také se zobrazují v tabulce na stránce „Data“.

## 4.5 Webový server pro komunikaci s GUI

Systém má webové uživatelské rozhraní poskytované *Next.js* serverem (kapitola 7). Jeho propojení s jádrem systému (backend) je řešeno pomocí frameworku Tornado pro jazyk Python, který tak tvoří rozhraní mezi GUI a databází, všemi regresory (predikce, simulace i učení) a Somfy API. Klient v podobě webového prohlížeče odesílá HTTP požadavky na tyto endpointy:

- **ep\_data** poskytuje data z databáze na základě POST požadavku s volitelnými JSON parametry (`ts_start` a `ts_end`) v těle označujícími časovou známku začátku a konce intervalu, vrací seznam vzorků (úryvek kódu 4.2) pod klíčem `data` ve formátu JSON jako tělo odpovědi.
- **ep\_control** slouží k predikci řízení jednoho z regresorů (parametr `classifier_name` na základě hodnot příznaků předaných jako JSON parametr `features`, vrací dobu trvání predikce (`controlTime`), status: `ok` a navržené hodnoty řízení (jako v úryvku kódu 4.1)).
- **ep\_train** TODO: Doplnit po zprovoznění.
- **ep\_predict** slouží k predikci řízení regresorů předaných v parametru `classifiers` pro reálná data (přijímá začátek a konec intervalu stejným způsobem jako `ep_data`). Pro každý z regresorů vrací pod klíčem s jejich názvem dvojici seznamů časových známek a uspořádaných dvojic hodnot řízení doporučených daným regresorem.



OBRÁZEK 4.2: Schéma endpointů a přenášených struktur.  
 Obrázek znázorňuje endpointy backendu systému pro automatické ovládání žaluzí a datové struktury přenášené v těle jednotlivých požadavků a odpovědí. Zobrazuje také použitou metodu požadavků, v posledním případě se nejedná o metodu HTTP ale o protokol WebSockets.

- ws zajišťuje spojení pomocí protokolu WebSockets (sekce 2.2.2) pro pravidelnou aktualizaci vizualizace stavu žaluzí a pro přenos požadavků na jeho změnu.

Schéma těchto endpointů a některých přenášených dat je na obrázku 4.2.

## 5 Příznaky pro strojové učení



Při návrhu systému se vycházelo z předpokladu, že uživatel rozhoduje o ručním řízení žaluzí na základě subjektivního odhadu, případně měření, některých fyzikálních veličin. Jedním z cílů této práce je rozhodování uživatele nahradit automatickým rozhodováním, je tedy nutné stanovit množinu veličin, které se budou v pravidelných intervalech měřit a na základě jejich hodnot automaticky generovat vhodné řízení pomocí modelů získaných strojovým učením. Uspořádá-li se tato množina do vektoru, nazve se příznakovým vektorem, jeho prvky pak příznaky. Tato kapitola se zabývá volbou příznaků.

Uživatel může nastavením žaluzie sledovat různé cíle, hlavní 4 byly identifikovány takto:

1. Za šera a tmy mají být žaluzie zatažené pro zajištění soukromí uvnitř místnosti při použití umělého osvětlení a zabránění vniknutí světla od projíždějících vozidel v době spánku uživatele v místnosti.
2. Při slunečných dnech, kdy venkovní vzduch dosahuje teplot blízkých pokojové teplotě nebo je převyšuje, mají žaluzie bránit průniku přímého slunecního záření do místnosti.
3. Při teplotách nižších, kdy je nutné interiér vytápět, by žaluzie naopak měly umožnit maximální průchod záření oknem, aby se tak místnost vytápěla a nebylo nutné využívat zbytečně vysoké množství energie na běžné vytápění.
4. V případě silného větru by měly být žaluzie vytažené, aby se zabránilo jejich poškození.

Z bodu 1 vyplývá, že uživatel sleduje intenzitu osvětlení exteriéru, orientačně je možné ji určit podle času a data, ale silně závisí také na počasí (zejména oblačnosti) a protože jsou na trhu dostupné senzory, které komunikují pomocí standardních sběrnic, je vhodnější ji přímo měřit. Použití umělého osvětlení uvnitř místnosti jimi lze detekovat také. Vnitřní a vnější intenzita osvětlení tedy byly zvoleny jako příznaky.

Body 2 a 3 zmiňují vliv teploty venkovního vzduchu na manuální ovládání uživatelem. Dá se ale předpokládat, že pokud by teplota vzduchu uvnitř místnosti byla podle uživatele příliš nízká, nezastiňoval by okno a nechal místnost vytápět i slunečním zářením i přes vysokou vnější teplotu. Proto byly dalšími příznaky zvoleny teplota vzduchu uvnitř a teplota vzduchu venku. Kromě toho uživatel může sledovat předpověď počasí a na jejím základě vyhodnotit, že je vhodné stínit dříve než teplota v průběhu dne

vzroste, protože by už nemusel mít možnost vyhřátý interiér ochladit větrním vzduchem. Jako příznaky se tedy zvolily předpovězené teploty na hodinu, 2 a 3, předpověď nejvyšší denní teploty a odhadovaný stav počasí ve formě číselného kódu. Okamžité teploty byly měřeny pomocí digitálního teploměru (sekce 3.1), předpovědi a odhad se získávaly z internetu pomocí OpenWeather API (sekce 4.3).

Z bodu 4 vyplývá vliv rychlosti větru na žaluzie, proto byla zvolena jako další příznak. Manuální vyhodnocení uživatelem probíhá obvykle subjektivně podle hluku způsobeného kmitáním lamel žaluzií v boční vodicí drážce. Na něj může kromě rychlosti mít vliv i směr větru, kvůli různícímu se obtékání domu proudícím vzduchem. Obě veličiny byly zvoleny jako příznaky, jejich měření bylo vyhodnoceno jako příliš složité a nákladné a odhad jejich hodnot se tedy získávají stejně jako předpovědi teploty (sekce 4.3). Systém žaluzií od firmy Somfy také obsahuje vlastní anemometr, který žaluzie v případě silného větru zadá žaluziím příkaz k vytažení, uživatel ale může být opatrný a žaluzie vytahovat již při nižších rychlostech větru.

Lidé dále mohou mít pravidelné zvyky, které ovlivňují nastavení žaluzií. Z toho důvodu byly jako příznaky zvoleny některé časové údaje: počet uplynulých sekund v rámci dne, počet uplynulých dnů v rámci týdne a pořadí dne v roce. Poslední z nich souvisí také s obvyklým počasím, které může ovlivňovat nastavení žaluzií. V případě nepřítomnosti uživatele v domácnosti nemusí nastavení žaluzií přesně odpovídat obvyklému záměru uživatele, proto je jedním z příznaků také jeho přítomnost v domácnosti. Ta se vyhodnocuje v systému domácí automatizace na základě periodicky přenášené polohy uživatelského telefonu.

Přehled všech 15 příznaků, jejich označení, jednotky a očekávané nejnižší a nejvyšší hodnoty je uveden v tabulce 5.1

Příznak	Označení	Množina
Den v roce	year_day	$\langle 1, 365 \rangle$
Den v týdnu	week_day	$\langle 0, 6 \rangle$
Denní čas [s]	day_secs	$\langle 1, 86400 \rangle$
Uživatel je doma	home	$\{0, 1\}$
Vnitřní teplota [°C]	temp_in	$\langle 10, 35 \rangle$
Venkovní teplota [°C]	temp_out	$\langle -20, 40 \rangle$
Osvětlení uvnitř [lux]	lum_in	$\langle 0, 1000 \rangle$
Osvětlení venku [lux]	lum_out	$\langle 0, 60000 \rangle$
OWM - maximální denní teplota [°C]	owm_temp_max	$\langle -20, 40 \rangle$
OWM - předpověď teploty za 1h [°C]	owm_temp_1h	$\langle -20, 40 \rangle$
OWM - předpověď teploty za 2h [°C]	owm_temp_2h	$\langle -20, 40 \rangle$
OWM - předpověď teploty za 3h [°C]	owm_temp_3h	$\langle -20, 40 \rangle$
OWM - kód stavu počasí	owm_code	$\langle 200, 804 \rangle$
OWM - rychlosť větru [ $\frac{\text{m}}{\text{s}}$ ]	owm_wind_speed	$\langle 0, 50 \rangle$
OWM - směr větru [°]	owm_wind_heading	$\langle 0, 359 \rangle$

TABULKA 5.1: 15 příznaků použitých pro automatické ovládání žaluzie, jejich označení v systému a množiny očekávaných hodnot.



# 6 Regresory

Při řízení žaluzí je nutné automaticky generovat akční zásahy. Regresory slouží právě k tomuto účelu, pro vzorek sestávající z určitého počtu (které se odvíjí od architektury konkrétního regresoru) příznakových vektorů vraci hodnoty doporučeného nastavení žaluzí, které by mělo co nejlépe odpovídat požadavkům uživatele. V práci jsou použity 3 regresory nazvané „If-else“, „FFNN“ a „LSTM“ s různými principy. Tato kapitola se zabývá jejich návrhem, specifickými vlastnostmi, které z nich vyplývají, a jeho metodami.

Poslední dva zmíněné regresory využívají ke generování výstupu neuronových sítí (NS). K práci s jejich modely bylo využíváno rozhraní Keras v jazyce Python, které umožňuje jejich učení, vyhodnocování i následné použití pro regresi (Keras API Reference 2022). Při volbě jejich vnitřní struktury a parametrů učení se sletěly natrénovat sítě se všemi možnými kombinacemi určitých parametrů a následně se vybrala nejlepší z nich ve smyslu minimální střední kvadratické odchylky na testovací datové sadě (sekce 6.4). Parametry a prohledávané hodnoty jsou uvedeny v sekcích 6.2 a 6.3.

Všechny regresory mají jednotné rozhraní a jimi doporučené řízení poskytují prostřednictvím metod `control()` a `predict()`. První z nich je určena pro jednotlivé vzorky, druhá pak pro jejich seznam. Obě se volají při obsluhování příslušných požadavků z uživatelského rozhraní při použití Simulátoru (sekce 7.2) nebo grafu řízení na záložce Live (sekce 7.4). Metoda `predict()` se od metody `control()` liší tím, že u regresorů, které využívají NS, se celá vstupní data zpracovávají najednou, což má pozitivní vliv na rychlosť. Kvůli architektuře použité knihovny Keras je ale nutné mít správně nastavený počet přijímaných vzorků ve vstupní vrstvě modelu, proto se při každém požadavku odvozuje nový model. Ten má stejné vnitřní parametry jako původní a poskytuje tak pro daná data stejné výsledky.

Přetrénování regresorů založených na NS je možné spustit pomocí metody `train()` na předaných datech.

## 6.1 Regresor založený na pravidlech (If-else)

Základní regresor „If-else“ tvoří výstup deterministicky na základě pravidel vyjádřených ve formě zřetězených konstrukcí if-elif-else (odtud tedy název) v programovacím jazyku Python. Slouží k porovnání dále navržených metod s přístupem v domácí automatizaci, který se v současnosti v praxi používá častěji (Apple Home 2022, OpenHAB 2022, Home Assistant 2022), tedy rozdělování dle neměnných pravidel.

Pravidla byla konstruována podle slovy formulovaných požadavků uživatele na stav žaluzí v závislosti na aktuálních podmírkách (vyjádřených příznakovým vektorem) s ohledem na to, že konkrétní parametry může být nutné změnit, protože je uživatel není schopen přesně definovat, nastavují se proto

v konfiguračním souboru `cfg_ifelse.yml`. Formulována byla tato pravidla pro výšku vytažení žaluzie:

1. Ve dne<sup>1</sup>:

- (a) Během jara a podzimu<sup>2</sup>: Je-li chladno<sup>3</sup>, žaluzie má být vytažená, jinak v případě střední teploty<sup>4</sup> se má rozhodovat na základě rozdílu mezi teplotami předpovězenými za 2 a 1 hodinu (růst teploty). Pokud je růst teploty v nastaveném pásmu nebo vyšší nebo je teplota vnějšího vzduchu vyšší než obě meze, žaluzie má být ve stejně pozici jako by bylo léto (následující bod), v případě růstu nižšího se má vytáhnout.<sup>5</sup>
- (b) V létě<sup>6</sup>: Pokud to dovolují podmínky a uživatel je doma, žaluzie má být vytažená, jinak zatažená. Podmínky vhodné pro vytažení žaluzie jsou takové, kdy nehrozí přehřátí interiéru - rozdíl azimutu slunce a azimutu směru kolmého na žaluzii (směrem od domu) v absoultní hodnotě je vyšší než  $90^\circ$  nebo je předpovězená nejvyšší denní teplota nižší, než nastavená mez (například z důvodu oblačnosti, deště atp.).
- (c) V zimě<sup>7</sup>: Žaluzie má být vytažená.

2. V noci v kterémkoli období má být žaluzie zatažená.

Kromě pravidel pro výšku vytažení byla formulována také pravidla pro míru naklopení:

1. Přes den:

- (a) Během jara a podzimu: V případě potřeby (vysoká intenzita osvětlení venku a zároveň alespoň vysoká teplota vzduchu uvnitř nebo teplota venkovního vzduchu, vše vzhledem k nastavitelným mezmí) se má žaluzie naklopit jako by bylo léto, jinak mají být její lamely vodorovně.
- (b) Během léta se má žaluzie naklopit o nejmenší možný úhel vzhledem k vodorovné rovině tak, aby zabránila (je-li to vzhledem k vzájemné poloze Slunce a žaluzie nutné) průchodu přímého slunečního záření oknem. Jestliže je ale intenzita osvětlení nižší než nastavená mez (vlivem počasí, nebo času), mají být lamely žaluzie vodorovně.
- (c) V zimě je žaluzie vytažená a úhel naklopení nemá smysl uvažovat, z konstrukce bude žaluzie vodorovně.
- (d) V případě, že uživatel není doma, naklopení žaluzie (jiné než vodorovné) má odpovídat  $\frac{8}{10}$  jinak definovaného naklopení.

<sup>1</sup>Den znamená, že je vnější intenzita osvětlení vyšší než nastavená mez a zároveň je čas vyšší než nastavený čas vstávání v daný den týdne. Jinak je noc.

<sup>2</sup>Jaro a podzim jsou dny v roce z intervalů  $\langle 80, 134 \rangle \cup \langle 274, 320 \rangle$

<sup>3</sup>Chladno znamená, že je teplota venkovního vzduchu nižší než nastavená mez. Zvolena byla podle zkušenosti s vytápním použitého domu na  $10^\circ\text{C}$ .

<sup>4</sup>Střední teplota znamená že je teplota venkovního vzduchu v intervalu mezi zmíněnou mezí pro „chladno“ a další nastavitelnou mezí, která byla zvolena na  $18^\circ\text{C}$ .

<sup>5</sup>Nížší hranice pásmá byla zvolena na  $-1,5^\circ\text{C}$ , vyšší pak na  $1,5^\circ\text{C}$ .

<sup>6</sup>Léto jsou dny v roce z intervalu  $\langle 135, 273 \rangle$

<sup>7</sup>Zima jsou ostatní dny, tedy dny v roce z intervalů  $\langle 321, 366 \rangle \cup \langle 1, 79 \rangle$

2. V noci: Žaluzie úplně naklopená, vyjímkou je případ, kdy je vnitřní teplota nižší než nastavená mez - to může znamenat, že se uživatel snaží vétrat a je tak vhodné žaluzie otevřít jen tak, aby mezi lamelami mohl lépe proudit vzduch, ale i přes to bránily průchodu světla pod malými úhly (po ulici projíždějící auta atp.).

Z návrhu vyplývá, že se tento regresor nepřizpůsobuje novým návykům uživatele, protože využívá v čase neměnných pravidel. Některá pravidla mají volitelné parametry, které lze ladit a dosáhnout tak přesněji uživatelem požadovaných výsledků.

## 6.2 Regresor využívající dopřednou neuronovou síť (FFNN)

První regresor založený na neuronových sítích byl nazvaný „FFNN“ podle anglického sousloví *feed-forward neural network*, které označuje dopřednou neuronovou síť, ~~využívanou tímto regresorem~~. Při volbě její vnitřní struktury byly prohledávány všechny kombinace hodnot parametrů uvedených v tabulce 6.1. Všechny vrstvy sítě jsou plně propojené. Neurony v nich vy-

Parametr	Seznam hodnot
struktura sítě	[10, 15], [15, 10], [10, 15, 10], [20, 15, 10], [10, 15, 20]
počet epoch	500, 200, 700
velikost dávky	16, 32
ztrátová funkce	MSE, kosinová podobnost

TABULKA 6.1: Prohledávání parametry dopředné neuronové sítě. Tabulka zachycuje 4 parametry neuronové sítě a jejího učení jejichž vzájemné kombinace se prohledávaly za účelem získání nejlepší z nich pro úlohu regrese hodnot nastavení venkovní žaluzie na základě 15 příznaků. Hodnoty parametru struktura sítě jsou seznamy (v hranatých závorkách) počtu neuronů v jednotlivých vrstvách sítě. MSE značí střední kvadratickou odchylku.

užívají aktivační funkci ReLU, kromě výstupní vrstvy, která je aktivována funkcí sigmoid (obě funkce jsou popsány v sekci 2.1.1), která je omezená shora i zdola a výstup vynásobený koeficientem je tak možné přímo použít jako případný akční zásah.

## 6.3 Regresor využívající rekurentní neuronovou síť (LSTM)

Tento regresor využívá rekurentní neuronovou síť **zvanou long short-term memory** (LSTM, sekce 2.1.3). Vnitřní struktura byla volena na základě výsledků prohledávání parametrů (popsáno v úvodu této kapitoly), prohledávané hodnoty každého z nich jsou uvedené v tabulce 6.2. Aktivační funkci je v tomto případě ve všech vrstvách hyperbolický tangens a funkce sigmoid

Parametr	Seznam hodnot
struktura sítě	[64, 32], [64, 16], [64, 64], [64, 16, 16]
počet epoch	100, 200
velikost dávky	16, 64
ztrátová funkce	MSE, kosinová podobnost

TABULKA 6.2: Prohledávání parametry rekurentní neuronové sítě. Tabulka zachycuje 4 parametry neuronové sítě a jejího učení jejich vzájemné kombinace se prohledávaly za účelem získání nejlepší z nich pro úlohu regrese hodnot nastavení venkovní žaluzie na základě 15 příznaků. Hodnoty parametru struktura sítě jsou seznamy (v hranatých závorek) počtu LSTM buněk v jednotlivých vrstvách sítě. MSE značí střední kvadratickou odchylku.

pro rekurentní krok. Výstupní vrstva je stejná jako v případě NS regresoru FFNN.



## 6.4 Příprava dat pro strojové učení

Naměřená data jsou uložená v databázi (kapitola 4) ve svých původních jednotkách a jednotlivě podle okamžiku pořízení. Pro jejich využití při strojovém učení byla data načtená z databáze přeškálována a vytvořily se z nich vzorky dle architektury jednotlivých NS. Vzorky byly náhodně a disjunktně rozděleny do 3 datových sad (trénovací, validační a testovací) v poměru 8:1:1. Výše popsané činnosti vykonával skript `mongo2h5.py`, který na závěr 3 datové sady pro každou síť uložil do souboru pro pozdější využití.

Hodnoty jednotlivých veličin byly převedeny do intervalu  $\langle 0, 1 \rangle$  posunutím a škálováním jejich očekávaného intervalu uvedeného v tabulce 5.1 v kapitole 5. Pro regresor FFNN každý vzorek v datových sadách odpovídal jednomu naměřenému vzorku, zatímco pro regresor LSTM každý obsahoval příznaky ze 64 časových okamžiků a požadované hodnoty nastavení žaluzie pro poslední z nich. Uložený soubor s datovými sadami byl ve formátu h5 a jeho název obsahoval informaci o časovém intervalu, ze kterého data v něm pochází, a o architektuře NS, pro kterou je určen.

**Trénovací datovou sadu** tvoří 80% náhodně vybraných vzorků a při trénování jsou její vzorky po dávkách předkládány NS. V každém kroku je pro ně vyhodnocována ztrátová funkce, jejíž hodnoty slouží k nastavení vnitřních parametrů učícím algoritmem.

**Validační datovou sadu** tvoří 10% náhodně vybraných vzorků. V každém kroku učení se na ní vyhodnocuje několik metrik. **Není na ní závislosti učení NS** a je tak možné ověřit, zda síť dobře zobecňuje.



**Testovací datová sada** sestává ze zbylých 10% vzorků, které neobsahuje ještě žádná z datových sad. Byla použita pro vyhodnocení úspěšnosti učení sítí s různou strukturou při hledání té nejhodnější (sekce 6.2 a 6.3).



# 7 Grafické uživatelské rozhraní a retraining

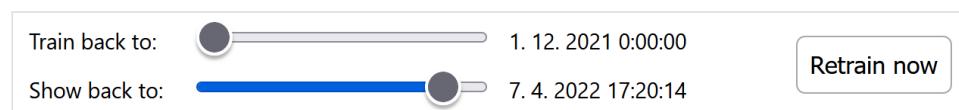
K ovládání navrhovaného systému slouží grafické uživatelské rozhraní implementované jako webová aplikace s využitím frameworku Next.js (*Next.js 2022*). Jednotlivé funkce systému jsou zpřístupněny na 4 oddělených stránkách, které jsou dále popsány v sekcích 7.1 až 7.4. Klíčové funkce jednotlivých stránek jsou následující:

- Data - zobrazení hodnot naměřených příznaků a ručního řízení v tabulce.
- Simulator - ruční vyhodnocování výsledku některých regresorů pro libovolně nastavené hodnoty příznaků.
- Control - ruční řízení žaluzie.
- Live - grafické zobrazení hodnot skutečného řízení a řízení získaných z jednotlivých regresorů společně s naměřenými daty.

Celá aplikace se skládá z komponent psaných v jazyce TypeScript, Next.js se pak stará o jejich komplikaci do JavaScriptu a následný přenos do prohlížeče uživatele. Všechny stránky mají společné záhlaví (obrázek 7.1), které obsahuje ovládací panel pro nastavení intervalu pro zobrazení dat a pro trénování neuronových sítí regresorů (kapitola 6), zároveň je možné pomocí tlačítka spustit trénování na zvolených datech (sekce 7.5). V pravém rohu záhlaví je pak menu, které slouží k přepínání stránek v rámci GUI.

## 7.1 Stránka Data

Většinu stránky Data zaujímá tabulka, ve které jsou v řádkách uvedeny jednotlivé vzorky naměřených dat s hodnotami všech příznaků a řízení žaluzií uživatelem společně s časem pořízení vzorku. Vlevo od každého řádku je navíc ikona, která reprezentuje důvod pořízení vzorku: 📅 pro periodické vzorky, 🎮 pro vzorky pořízené na základě změny stavu žaluzie (toto chování je popsáno v kapitole 4). Vzorky jsou seřazeny chronologicky od nejmladších



OBRÁZEK 7.1: Záhlaví grafického uživatelského rozhraní systému pro automatické ovládání žaluzie. Umožňuje nastavení intervalu pro zobrazení dat a pro trénování neuronových sítí využívaných pro ovládání žaluzie a spuštění trénování na zvolených datech.



OBRÁZEK 7.2: Část legendy tabulky s naměřenými daty v grafickém uživatelském rozhraní systému pro automatické ovládání žaluzí.



Position: 33                      Tilt: 91

OBRÁZEK 7.3: Příklad vizualizace výstupu regresoru v úloze automatického ovládání žaluzie na základě 15 příznaků.

k nejstarším a zobrazují se pouze ty, které jsou z časového intervalu daného nastavením v záhlaví aplikace (popsáno v úvodu této kapitoly). V záhlaví tabulky jsou místo označení příznaků použity ikony, které nezabírají tolík místa. Jejich význam vysvětluje legenda, která je umístěna vpravo od tabulky (obrázek 7.2). Záhlaví tabulky i legenda zůstávají při rolování tabulkou směrem ke starším datům zafixované v horní části pohledu tak, aby měl uživatel stále přehled o významu hodnot, které sleduje.

## 7.2 Stránka Simulator

Tato stránka slouží pro ruční vyhodnocování výstupu regresorů v závislosti na nastavených hodnotách příznaků. Je rozdělena do dvou částí, vlevo je sloupec vstupních prvků, vpravo pak rámce jednotlivých regresorů s jejich výstupem a jeho jednoduchou vizualizací. Výška vytažení žaluzie (Position) je reprezentována částečným vybarvením obdélníku shora, úhel naklopení lamel (Tilt) pak rotací čáry uvnitř kruhu (příklad je na obrázku 7.3).

Vzhledem k povaze vstupních prvků, pomocí kterých je možné zadat hodnoty pouze v jednom časovém okamžiku, se vyhodnocuje výstup pouze regresoru If-else a FFNN, regresor LSTM totiž vyžaduje hodnoty příznaků ne z jednoho, ale z 64 okamžiků, což by vyžadovalo složitější způsob zadávání.

Při každé změně vstupu se pro každý sledovaný regresor odešle požadavek na endpoint `ep_control` backendu systému (sekce 4.5), v odpovědi se vrátí výstup regresoru, který je poté zobrazen.

### 7.3 Stránka Control

Další stránkou v GUI je Control, která slouží k zobrazení aktuálního stavu žaluzie a k jeho ruční změně. Skládá se ze dvou hlavních částí uspořádaných pod sebou - tabulky se stavů žaluzie (nahoře) a vizualizace (dole).

Tabulka má tři sloupce, v prvním z nich se zobrazuje aktuální stav, ve druhém jsou pole pro nastavení nového stavu a ve třetím tlačítka pro odeslání požadavku na změnu stavu. Každý řádek je pro jednu stavovou proměnnou žaluzie.

Vizualizace zobrazuje žaluzii v řezu. Počet lamel a vzdálenost spodní lamely od okraje odráží výšku vytažení žaluzie, jejich odchylka od vodorovné osy pak odpovídá náklonu lamel skutečné žaluzie. Při změně hodnot v tabulce se vizualizace přepne do režimu náhledu, kdy je možné sledovat, jak by skutečná žaluzie vypadala, pokud by se odeslaly hodnoty nastavené v tabulce. Uživateli o tom informuje text *Preview* v pravém horním rohu vizualizace. Tlačítkem *Reset*, nacházejícím se tamtéž, je možné tento režim opustit a vizualizace pak bude opět zobrazovat aktuální stav žaluzie. Vizualizace je psána ve značkovacím jazyce pro dvourozměrnou vektorovou grafiku SVG (Dahlström et al., 2011).

Mezi tabulkou a vizualizací je navíc ještě zaškrťávací pole *Test Mode*, které umožňuje aktivovat nebo deaktivovat testovací režim při sběru dat, který se využívá při vývoji, aby databáze neobsahovala data vzniklá na základě falešných událostí.

Pro získávání aktuálního stavu a nastavení nového se používá spojení s backendem (sekce 4.5) pomocí protokolu WebSockets na jeho endpointu ws. Po připojení je každých 15 s do webového prohlížeče uživatele GUI doručován aktuální stav žaluzie ve zprávách ve formátu JSON a stejným způsobem se odesílá požadavek na jeho změnu. V pravém horním rohu stránky je indikátor spojení pomocí protokolu WebSockets.

### 7.4 Stránka Live

Poslední ze 4 stránek GUI je stránka Live, která slouží k živému vyhodnocování řízení doporučeného jednotlivými regresory na reálných datech. Skládá se ze dvou grafů uspořádaných pod sebou. V horním grafu jsou zobrazena naměřená data, v dolním pak řízení žaluzie (manuální řízení i doporučené řízení regresorů). Nad každým z grafů se nachází interaktivní legenda, která umožňuje skrývat některé linie v grafu pro větší přehlednost podle požadavků uživatele GUI. Kurzorem myši lze sledovat konkrétní hodnoty všech veličin v grafu v okamžiku daném jeho polohou.

Data zobrazovaná v grafech se získávají pomocí 2 požadavků zasílaných na endpointy backendu (sekce 4.5). Naměřená data se přenáší vždy všechna na základě požadavku na ep\_data při prvním načítání aplikace, predikce regresorů kromě okamžiku prvního načtení aplikace také vždy při změně časového intervalu pro zobrazení dat v záhlaví aplikace (popsáno v úvodu této kapitoly). Oba grafy ale vždy zobrazují pouze data z časového intervalu daného nastavením v záhlaví stránky (*Show back to:*).

## 7.5 Retraining neuronových sítí regresorů





# Bibliografie

- [1] Warren S McCulloch a Walter Pitts. „A logical calculus of the ideas immanent in nervous activity“. In: *The bulletin of mathematical biophysics* 5.4 (1943), s. 115–133.
- [2] Erik Dahlström et al. *Somfy Open API*. 2011. URL: <https://www.w3.org/TR/SVG11/> (cit. 21. 04. 2022).
- [3] *Adafruit TSL2591 High Dynamic Range Digital Light Sensor*. Adafruit Industries. 2014.
- [4] *ESP-12E WiFi Module*. Version 1.0. AI-Thinker team. 2015.
- [5] *Somfy Open API Developer Portal / APIs & Docs*. Somfy. 2018. URL: <https://developer.somfy.com/apis-docs> (cit. 23. 03. 2022).
- [6] *DS18B20 – Programmable Resolution 1-Wire Digital Thermometer*. 19-7487; Rev 6; 7/19. Maxim Integrated Products, Inc. 2019.
- [7] Thibaut Etienne. *Somfy Open API*. 2021. URL: <https://github.com/tetienne/somfy-open-api#readme> (cit. 23. 03. 2022).
- [8] *MongoDB Documentation*. MongoDB, Inc. 2021. URL: <https://www.mongodb.com/docs/> (cit. 20. 04. 2022).
- [9] *Apple Home*. Apple Inc. 2022. URL: <https://www.apple.com/ios/home/> (cit. 21. 04. 2022).
- [10] *Home Assistant*. Home Assistant. 2022. URL: <https://www.home-assistant.io/> (cit. 21. 04. 2022).
- [11] *Keras API reference*. Keras. 2022. URL: <https://keras.io/api/> (cit. 21. 04. 2022).
- [12] *Next.js*. Vercel. 2022. URL: <https://nextjs.org/> (cit. 21. 04. 2022).
- [13] *OpenHAB*. OpenHAB Foundation. 2022. URL: <https://www.openhab.org/> (cit. 21. 04. 2022).