



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.

7. 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

Abstrakt	v
1 Úvod	3
1.1 State of the Art	3
1.2 Thesis Objectives	3
1.3 Thesis Outline	3
 2 Současný stav problematiky	5
2.1 Neuronové sítě	5
2.1.1 Aktivační funkce	5
2.1.2 Dopředná neuronová síť	5
2.1.3 Rekurentní neuronová síť, LSTM	5
3 Hardwarové komponenty	7
3.1 Zařízení pro měření teploty vzduchu a intenzity osvětlení . . .	7
3.1.1 Zapojení	8
3.1.2 Deska plošných spojů	8
3.1.3 ESP-12E	9
3.1.4 DS18B20	9
3.1.5 TSL2591	10
3.2 Raspberry Pi	10
3.3 Pracovní stanice KKY-PC	10
4 Sběr dat a komunikace komponent	11
4.1 MQTT	11
4.2 REST API výrobce pohonu žaluzie	12
4.3 OpenWeather API	14
4.4 Databáze	14
4.5 Webserver	15
5 Příznaky pro strojové učení	17
6 Regresory	21
6.1 If-else	21
6.2 FFNN	23
6.3 LSTM	23
6.4 Příprava dat pro strojové učení	23
6.4.1 Testovací datová sada	23
7 Grafické uživatelské rozhraní a retraining	25
7.1 Data	25
7.2 Simulátor	25

7.3	Control	25
7.4	Live	25
7.5	Retraining	25
8	Vyhodnocení	27
8.1	Referenční data	27
8.2	Vliv příznaků na predikci	27
8.3	Porovnání regresorů	27
8.4	Vyhodnocení predikce jednotlivých regresorů na vybraných datech	27
8.5	Vyhodnocení vlivu retrainingu	27
9	Možnosti nasazení systému	29
10	Diskuze	31
10.1	Recapitulation of Methods	31
10.2	Shrnutí výsledků	31
11	Závěr	33
11.1	Future Work	33
	Bibliografie	35
A1	Structure of the Workspace	35

Seznam obrázků

3.1	Schéma zapojení měřicích zařízení	8
3.2	Návrh DPS měřicích zařízení	10
4.1	Schéma komunikace komponent	12
4.2	Schéma endpointů a přenášených struktur	16

Seznam tabulek

4.1	Příznaky a jejich MQTT témata	12
5.1	Příznakový vektor	19
6.1	Prohledávané parametry dopředné neuronové sítě	23
6.2	Prohledávané parametry rekurentní neuronové sítě	24

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 vlastní konstrukce osazených modulem ESP-12E od společnosti Ai-Thinker (sekce 3.1). Na jednodeskovém počítači Raspberry Pi 2B, označme ho RPi1, běží 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 a MQTT broker¹ Mosquitto, ke kterému jsou připojeny všechny zdroje dat. Posledním celkem je workstation označovaná jako KKY-PC, která hostuje DB server MongoDB, GUI webserver a REST a WebSocket backend, dále sloužila také k trénování neuronových sítí regresorů.

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.

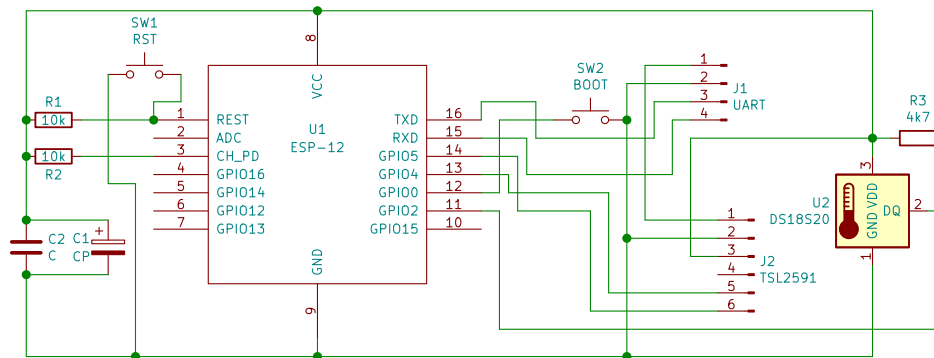


Jedna z nich je umístěna v žaluziemi zatemňované místnosti na polici tak, aby na připojené senzory nedopadalo přímé sluneční světlo, druhá pak na střeše domu, ve kterém je tato místnost, uvnitř krabičky s okénkem tak, aby byla elektronika krytá před povětrnostními vlivy, ale na senzor intenzity osvětlení i tak dopadalo světlo. Voděodolnost krabičky zajišťuje přišroubované víko se silikonovým těsněním a silikonem utěsněné kabelové průchodky. 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.



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

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



OBRÁZEK 3.1: Schéma zapojení zařízení, která měří intenzitu osvětlení pomocí senzoru TSLS2591 a teplotu vzduchu pomocí senzoru DS18B20 a hodnoty přenáší po WiFi

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

Na pinech **GPIO4** a **GPIO15** jsou zapojeny po řadě datový a hodinový vodič sběrnice I²C, 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 elektricky propouje **ostatí** komponenty těchto zařízení a byla vyrobena **po domácku**. V opensource programu pro návrh **elektronických** zařízení KiCad byl sestaven obvod a návrh DPS (obrázek 3.2), jenž byl pomocí kancelářské tiskárny překlopeně vytištěn na běžný papír v měřítku 1:1.

Z jednostranně poměděného laminátu (cuprexit) byla pákovými nůžkami vystřižena deska o příslušných rozměrech dle návrhu, přebroušena brusným papírem a byla zabalena do papíru s návrhem tak, aby strana s mědí překrývala 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 vodítko pro vrták.

Po vyznačení všech děr byla deska vyňata z papíru a lihovým fixem se zakreslily pájecí body součástek (černá mezikruží a obdélníky) a cesty mezi nimi podle papírové předlohy (nejtenčí linie jsou ohraničení 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.

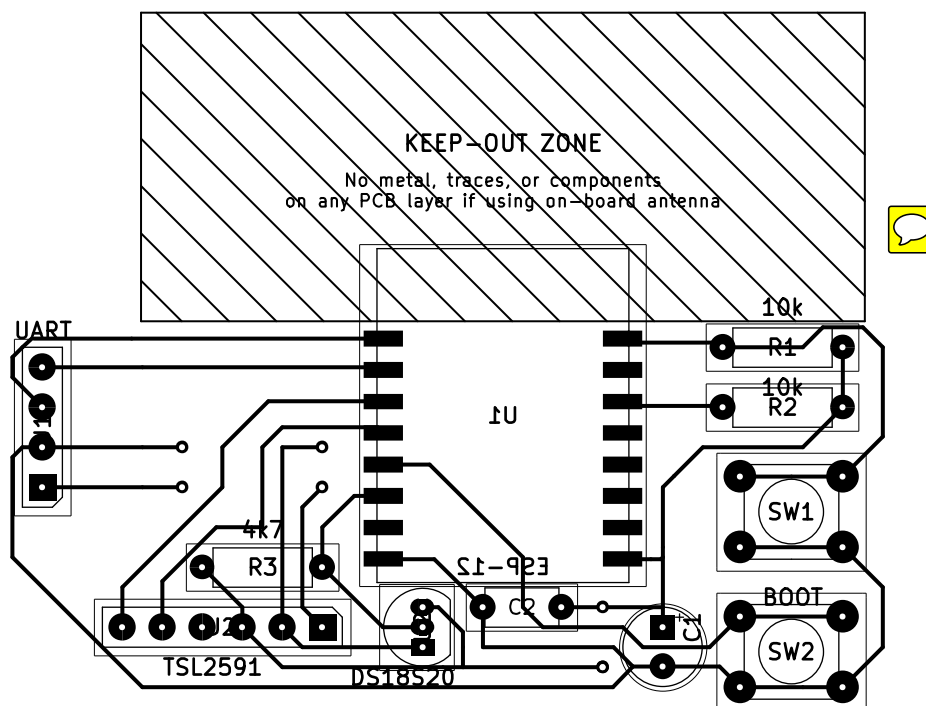
Přebytečná měď, která nebyla překryta barvou, se vyleptala při pokojové teplotě v roztoku chloridu železitého pro leptání plošných spojů za 25 minut a na desce tak zůstaly jen potřebné vodivé cesty. Leptání proběhlo tak, že se deska položila na hladinu roztoku nalitého do plastové leptací vaničky. Díky tomu mohly produkty chemické reakce klesat ke dnu a uvolnit prostor ještě nepoužitému roztoku. Deska byla několikrát z lázně vyjmuta a vizuálně zkontrolována, 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 znovu přebroušena a opatřena pájitelným ochranným lakem (v tomto případě kalafunou rozpuštěnou v acetonu). Součástky byly osazeny na příslušná místa dle návrhu desky a schématu.

3.1.3 ESP-12E

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 ukládá 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 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 zemnicí vodič. Volitelně lze také připojit napájecí napětí o velikosti 3 – 5,5 V třetím vodičem. Na stejné 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 nalezne využití ve stavebnictví při řízení vytápění, ventilace nebo klimatizace, měření 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.



OBRÁZEK 3.2: Návrh desky plošných spojů zařízení, která měří intenzitu osvětlení pomocí senzoru TSL2591 a teplotu vzduchu pomocí senzoru DS18B20 a hodnoty přenáší po WiFi

3.1.5 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 μ lux do 88000 lux. Měření lze číst po standardní sběrnici I²C, 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).

3.2 Raspberry Pi

3.3 Pracovní stanice KKY-PC

CPU: Intel(R) Core(TM) i7-7800X CPU @ 3.50GHz, 64-bit, 6 jádrový

GPU: Nvidia GTX 1080

RAM: 64 GiB Ubuntu 20.04.3

4 Sběr dat a komunikace komponent

Použitým algoritmům strojového učení je nutné dodat data sestávající z poloupnosti hodnot příznaků (více v kapitole 5) a odpovídajících stavů žaluzií (výška vytažení a sklon lamel) v daných časových okamžicích. Tato kapitola se zabývá metodami sběru dat v jednotlivých součástech systému, ~~přenosem z nich~~ a jejich vzájemnou komunikací.

Data se získávají ze 3 hlavních zdrojů na základě požadavku zaslání skriptem `mqttDataCollector.py`. Stav žaluzie a informaci, zda je uživatel přítomen v domácnosti¹ odesílá systém domácí automatizace *Home Assistant*, měřené veličiny (teplota a intenzita osvětlení) uvnitř a venku pak příslušná zařízení² a ostatní informace o počasí³ se získávají přímo v rámci skriptu `mqttDataCollector.py` z REST API OpenWeather. Data získaná z komponent se pak v jedné společné zprávě odesílají 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.

4.1 MQTT

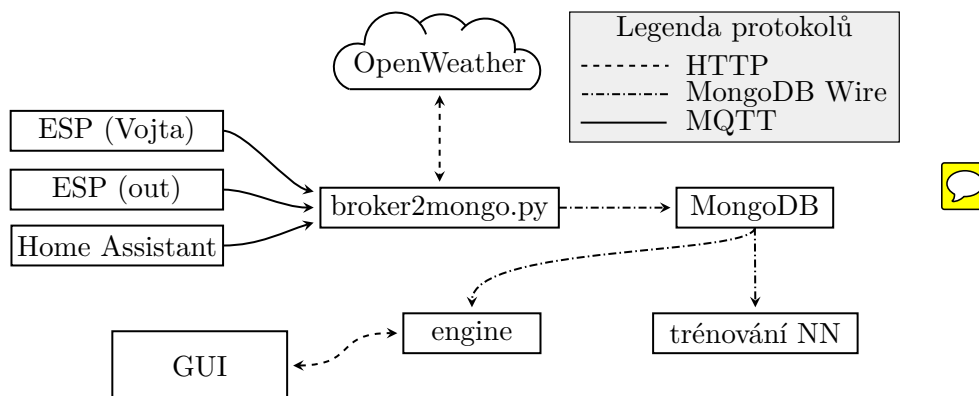
Protokol MQTT se využívá pro veškerou komunikaci součástí při sběru dat. Centrální uzel se nazývá broker a jedná se o software spuštěný na RPi1 (sekce 3.2) na výchozím TCP portu 1883. Připojují se k němu všechny komponenty, které využívají MQTT. Po připojení mohou publikovat zprávy do hierarchicky uspořádaných témat a přihlašovat se k jejich odběru, broker pak zajistí doručení zpráv publikovaných v určitém tématu klientům, kteří jsou přihlášení 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 „*req values*“. K odběru tohoto tématu jsou přihlášena obě zařízení s ESP8266 i systém domácí automatizace a po jejím přijetí odešlou aktuální hodnoty jimi sledovaných příznaků do odpovídajících témat dle tabulky 4.1. Všechna tato témata odebírá skript `mqttDataCollector.py`, který hodnoty příznaků společně s časovou známkou odesílá v jedné zprávě, která dále obsahuje informaci o tom,

¹příznaky `position`, `tilt` a `home`

²příznaky `lum_in`, `lum_out`, `temp_in` a `temp_out`

³rychlost (`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`)



OBRÁZEK 4.1: Schéma komunikace komponent. Obrázek znázorňuje vybrané komponenty systému automatického řízení žaluzií, 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

zda je aktivní testovací režim využívaný při vývoji, do tématu `smartblinds/data`. To odebírá skript `broker2mongo.py` popsáný v sekci 4.4.

Příznak	Téma
<code>temp_inside</code>	<code>smartblinds/temp/Vojta</code>
<code>temp_outside</code>	<code>smartblinds/temp/out</code>
<code>lum_inside</code>	<code>smartblinds/lux/Vojta</code>
<code>lum_outside</code>	<code>smartblinds/lux/out</code>
<code>home</code>	<code>smartblinds/presence/Vojta</code>
<code>position</code>	<code>smartblinds/position/Vojta</code>
<code>tilt</code>	<code>smartblinds/tilt/Vojta</code>

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

4.2 REST API výrobce pohyblivých žaluzií

Komunikace s aplikačním rozhraním je šifrovaná a probíhá pomocí protokolu HTTPS zasíláním požadavků na *endpointy* serverů výrobce dostupné na adrese `https://api.somfy.com/api/v1/`. Použitý systém domácí automatizace i backend vyvíjeného systému používají ke komunikaci s API knihovnu Pymfy, která mapuje akce pro manipulaci se žaluzií 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 zákazníka, 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 důvodů odlišností 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 žaluzií 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 žaluzií přiřazena konkrétní montáž. Na základě zadaného názvu, *callback URL*¹ a popisu aplikace se získají dva řetězce: *Consumer key* a *Consumer secret*. Somfy Open API Developer Portal | APIs & Docs 2018 Ty se pak společně s *callback URL*¹ předají konstruktoru objektu, který v rámci knihovny Pymfy reprezentuje API.Etienne, 202

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 pokračoval 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 řádku 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. Jednak systém domácí automatizace každou minutu kontroluje aktuální stav žaluzií 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, druhak stránka „Control“ v GUI umožňuje zobrazit aktuální stav žaluzií a zadávat příkazy k jeho změně. Samotnou komunikaci s API zajišťuje backend systému, se kterým se komunikuje přes protokol WebSockets. Po připojení alespoň jednoho klienta se každých 15 s zjistí stav žaluzie a připojeným prohlížečům se odešle zpráva ve formátu JSON. Její struktura je uvedena v úryvku kódu 4.1. Naopak po přijetí zprávy backendem se v závislosti na jejím obsahu odešle požadavek API 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*

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

s hodnotou datového typu `boolean`⁴, 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 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 OpenWeather API

Skript `mqttDataCollector.py` kromě měřených příznaků a stavu žaluzií zjišťuje 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 1 požadavek za 5 minut. Jako příznaků se využívá kód počasí (vyjadřuje jeho shrnutí – např. slunečno, polojasno, déšť, 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, na základě změny stavu žaluzie pomocí skriptu `mqttDataCollector.py`, ukládá skript `broker-2mongo.py` do databáze MongoDB. Jednotlivé vzorky jsou získávány z MQTT zpráv zasílaných do tématu `smartblinds/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,
    "week_day": int,
    "day_secs": int,
    "home": boolean,
    "temp_in": float,
```

⁴`boolean` je datový typ, který vyjadřuje pravdivostní hodnotu.

```

    "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
  }
}

```

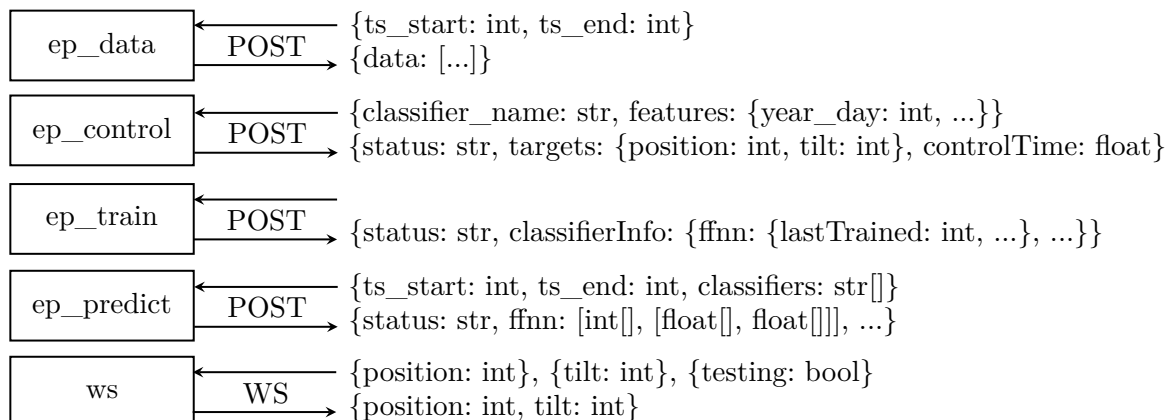
ÚRYVEK KÓDU 4.2: Struktura vzorku dat uloženého jako záznam v databázi MongoDB. Obsahuje hodnoty příznaků i stav žaluzií v okamžiku jeho pořízení a časovou známku. Místo hodnot jsou zde uvedeny jejich datové typy (*float* - číslo s plovoucí řádovou čárkou, *boolean* - pravdivostní hodnota, *int* - celé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

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

4.5 Webserver

Systém má webové uživatelské rozhraní získávané z *Next.js* serveru. Data a služby mu poskytuje HTTP backend implementovaný v Pythonu pomocí frameworku Tornado, který tak tvoří rozhraní mezi GUI a databází, **všemi regresory (predikce, simulace)** a Somfy API. **Přímo klientský prohlížeč vykonává požadavky na endpointy backendu, tedy:**

- *ep_data*, ~~který~~ 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*, ~~který~~ 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*, ~~který~~ přijímá začátek a konec intervalu stejným způsobem jako *ep_data* a seznam názvů regresorů v parametru *classifiers*. Pro každý z nich vrací pod klíčem s jejich názvem dvojici seznamů časových známek a uspořádaných dvojic hodnot řízení doporučených daným regresorem a



OBRÁZEK 4.2: Schéma endpointů a přenášených struktur. Obrázek znázorňuje endpointy backendu systému pro automatické ovládání žaluzií 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, který zjišťuje spojení pomocí protokolu WebSockets pro pravidelnou aktualizaci vizualizace stavu žaluzií a pro přenos požadavků na jeho změnu.

5 Příznaky pro strojové učení

Při návrhu systému se vycházelo z předpokladu, že uživatel rozhoduje o ručním řízení žaluzií 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 učním. 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 slunečního 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, nezastíňoval 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 venkovním 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 vodící drážce. Na něj může kromě rychlosti mít vliv i směr větru, kvůli různímu se obtékání domu proudícím vzduchem. Obě veličiny byly zvoleny jako příznaky, jejich měření bylo vyhodnoceno jako příliš složité a nákladné a odhady 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 - rychlost větru [$\frac{m}{s}$]	owm_wind_speed	$\langle 0, 50 \rangle$
OWM - směr větru [°]	owm_wind_heading	$\langle 0, 359 \rangle$

TABULKA 5.1: ~~Příznakový vektor. Tabulka zachycuje 15 příznaků použitých pro automatické ovládání žaluzie, jejich označení v systému a množiny očekávaných hodnot. Případné jednotky jsou uvedeny v hranatých závorkách za příznaky.~~

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] *Adafruit TSL2591 High Dynamic Range Digital Light Sensor*. Adafruit Industries. 2014.
- [3] *ESP-12E WiFi Module*. Version 1.0. AI-Thinker team. 2015.
- [4] *Somfy Open API Developer Portal / APIs & Docs*. Somfy. 2018. URL: <https://developer.somfy.com/apis-docs> (cit. 23.03.2022).
- [5] *DS18B20 – Programmable Resolution 1-Wire Digital Thermometer*. 19-7487; Rev 6; 7/19. Maxim Integrated Products, Inc. 2019.
- [6] Thibaut Etienne. *Somfy Open API*. 2021. URL: <https://github.com/tetienne/somfy-open-api#readme> (cit. 23.03.2022).

