



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.



11. května 2022

# Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 23. května 2022

---

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>ii</b>
<b>1 Úvod</b>	<b>1</b>
<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 Příznaky pro strojové učení</b>	<b>4</b>
<b>4 Hardwarové komponenty</b>	<b>7</b>
4.1 Zařízení pro měření teploty vzduchu a intenzity osvětlení . . . . .	7
4.1.1 Zapojení . . . . .	8
4.1.2 Deska plošných spojů (DPS) . . . . .	9
4.1.3 Modul ESP-12E s MCU a WiFi . . . . .	10
4.1.4 Teplotní senzor DS18B20 . . . . .	10
4.1.5 Senzor intenzity osvětlení TSL2591 . . . . .	11
<b>5 Sběr dat a komunikace komponent</b>	<b>12</b>
5.1 Použití komunikačního protokolu MQTT . . . . .	13
5.2 Komunikace s pohonem žaluzie . . . . .	13
5.3 Využití služby OpenWeather . . . . .	15
5.4 Databáze . . . . .	15
5.5 Webový server pro komunikaci s GUI . . . . .	16
<b>6 Automatické ovládání žaluzií</b>	<b>18</b>
6.1 Příprava dat pro strojové učení . . . . .	18
6.2 Regresor založený na pravidlech (If-else) . . . . .	19
6.3 Dopředná neuronová síť jako regresor (FFNN) . . . . .	21
6.4 Rekurentní neuronová síť jako regresor (LSTM) . . . . .	21
6.5 Přetrénování neuronových sítí (retraining) . . . . .	22
<b>7 Grafické uživatelské rozhraní</b>	<b>23</b>
7.1 Stránka Data . . . . .	23
7.2 Stránka Simulator . . . . .	24
7.3 Stránka Control . . . . .	25

7.4	Stránka Live . . . . .	26
<b>8</b>	<b>Vyhodnocení</b>	<b>28</b>
8.1	Vyhodnocení sběru dat a spolehlivosti . . . . .	28
8.2	Volba struktury NS a parametrů učení . . . . .	29
8.3	Vliv jednotlivých příznaků na predikci . . . . .	30
8.4	Porovnání regresorů . . . . .	34
8.5	Vyhodnocení vlivu retrainingu . . . . .	35
<b>9</b>	<b>Diskuze</b>	<b>37</b>
<b>10</b>	<b>Závěr</b>	<b>39</b>
	<b>Bibliografie</b>	<b>40</b>
	<b>A1 Structure of the Workspace</b>	<b>41</b>

# Seznam obrázků

4.1	Skutečné umístění měřicího zařízení . . . . .	8
4.2	Schéma zapojení měřicích zařízení . . . . .	9
4.3	Návrh DPS měřicích zařízení . . . . .	10
5.1	Schéma komunikace komponent . . . . .	12
5.2	Schéma endpointů a přenášených struktur . . . . .	17
7.1	Záhlaví GUI . . . . .	23
7.2	Stránka Data v GUI . . . . .	24
7.3	Stránka Simulator v GUI . . . . .	24
7.4	Vizualizace výstupu regresoru . . . . .	25
7.5	Stránka Control v GUI . . . . .	25
7.6	Stránka Live v GUI . . . . .	26
8.1	Graf sesbíraných dat (příklad) . . . . .	28
8.2	Graf druhu sesbíraných vzorků a výpadků . . . . .	29
8.3	Důležitost příznaků FFNN . . . . .	33
8.4	Důležitost příznaků LSTM . . . . .	33
8.5	Porovnání predikce regresorů . . . . .	34
8.6	Porovnání predikce FFNN (retraining) . . . . .	36
8.7	Porovnání predikce LSTM (retraining) . . . . .	36

# Seznam tabulek

3.1	Příznaky pro automatické ovládání žaluzie . . . . .	6
5.1	Příznaky a jejich MQTT téma . . . . .	13
6.1	Prohledávání parametry dopředné neuronové sítě . . . . .	21
6.2	Prohledávání parametry rekurentní neuronové sítě . . . . .	21
8.1	Výpadky . . . . .	29
8.2	Výsledky prohledávání parametrů FFNN . . . . .	31
8.3	Výsledky prohledávání parametrů LSTM . . . . .	32
8.4	Porovnání regresorů . . . . .	34
8.5	Retraining FFNN . . . . .	35
8.6	Retraining LSTM . . . . .	35



# 1 Úvod

Stínění oken v pozemních stavbách určených k bydlení se využívá k regulaci teploty a zajištění soukromí obyvatel. (Lubinová, 2013) Jednou z možností stínění jsou venkovní žaluzie osazené pohonem na dálkové ovládání. Jejich uživatel může měnit výšku vytažení žaluzie a sklon lamel a ovlivňovat tím množství záření, které skrz okno prochází. Pomocí vzdáleného ovládání pak lze žaluzie řídit automaticky.

Komplexní systém může generovat akční zásahy pro řízení žaluzií na základě různých vstupních informací (datum a čas, měření veličin pomocí senzorů) bez účasti uživatele za účelem zajištění jeho komfortu a úspory prostředků vynaložených na regulaci teploty v interiéru (vytápění a chlazení). Sběr informací na základě kterých se má rozhodovat o nastavení žaluzie i samotné rozhodování musí probíhat v reálném čase.

Tato práce se zabývá návrhem takového systému, návrhem měřicích zařízení použitelných pro měření veličin užitečných při rozhodování systému a možnostmi využití strojového učení k ovládání žaluzií v reálném čase. Zkoumá časový vývoj použitých algoritmů při opakovaném učení na postupně sbírány datech s ohledem na měnící se požadavky uživatele a měnící se podmínky v průběhu roku. Její cíle jsou následující:

1. Určit vhodné vstupní veličiny (příznaky) a výstupní veličiny pro automatické rozhodování o stavu žaluzie.
2. Zkonstruovat měřicí zařízení pro měřitelné příznaky odesílat jejich hodnoty pomocí MQTT<sup>1</sup>.
3. Zajistit komunikaci s vybranými žaluziemi pomocí poskytovaného API<sup>2</sup> a MQTT<sup>1</sup>.
4. Zajistit komunikaci se zdroji všech příznaků pomocí MQTT a data ukládat pro pozdější použití.
5. Navrhnout základní rozhodovací systém založený na pravidlech (baseline).
6. Navrhnout model neuronové sítě a porovnat jeho použitelnost s baseline systémem.

V práci se využívá skutečná venkovní žaluzie s hliníkovými lamelami a pohonem od firmy Somfy. Vzdálenou komunikaci s ním zajišťuje řídicí jednotka Tahoma připojená do internetu, která komunikuje se servery výrobce, jejichž prostřednictvím je možné žaluzii ovládat a zjistit její aktuální stav. Žaluzie se nachází před oknem pokoje autora v rodinném domě v obci vzdálené asi

---

<sup>1</sup>MQTT je komunikační protokol blíže popsaný v sekci 5.1

<sup>2</sup>Application Programming Interface (rozhraní pro programování aplikací)

12 km severovýchodně od Plzně. Uvnitř tohoto pokoje a na střeše daného domu také probíhají všechna měření.

## 3 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 detektovat 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 venkovním

vzduchem. Jako příznaky se tedy zvolily předpovězené teploty na hodinu, 2 a 3 dopředu, předpověď nejvyšší denní teploty a odhadovaný stav počasí ve formě číselného kódu. Okamžité teploty byly měřeny pomocí digitálního teploměru (sekce 4.1), předpovědi a odhad se získávaly z internetu pomocí OpenWeather API (sekce 5.3).

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 obtékaní 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 5.3). 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 3.1

Po natrénování použitých algoritmů (kapitola 6) byla vyhodnocena důležitost jednotlivých příznaků pomocí algoritmu *permutation feature importance* (Breiman, 2001), při kterém se sleduje vliv permutace hodnot jednotlivých příznaků mezi vzorky v rámci datové sady na přesnost predikce. Důležitost každého příznaku pak odpovídá rozdílu MSE s promíchanými hodnotami tohoto příznaku a s původní datovou sadou (nezamíchanou). Pro každý z příznaků se MSE vyhodnotila 20x pro různé permutace a vypočetl se z nich aritmetický průměr. Důležitost byla vyjádřena jako rozdíl tohoto průměru a původní MSE.

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	$\langle 0, 2000 \rangle$
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 3.1: 15 příznaků použitých pro automatické ovládání žaluzie, jejich označení v systému a množiny očekávaných hodnot.

# 4 Hardwarové komponenty

K provozování určitých součástí systému a měření 4 příznaků je nutné použít fyzické technické vybavení, označované také jako hardware. Tato kapitola popisuje jeho funkční celky složené z jednotlivých použitých součástí a zmíňuje využití hardwaru, které je blíže specifikováno v příslušných kapitolách (kapitoly 5 a 7).

K měření příznakových veličin (kapitola 3) 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 4.1). Na jednodeskovém počítači Raspberry Pi 2B, 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. Poháně ji operační systém Ubuntu 20.04.3 a po dobu vývoje běžela na Katedře kybernetiky Západočeské univerzity v Plzni.

## 4.1 Zařízení pro měření teploty vzduchu a intenzity osvětlení



Čtyři z příznaků pro regresory tvoří teplota vzduchu a intenzita osvětlení, obojí v interiéru i exteriéru. Obě veličiny byly měřeny periodicky každých 5 minut a v případě změny stavu žaluzie uživatelem pomocí již v úvodu této

---

<sup>1</sup>MQTT broker je centrální uzel, který zprostředkovává přenos zpráv od klienta, který zprávu odeslal, ke klientům, kteří se přihlásili k jejich odběru. Více v sekci 5.1.



OBRÁZEK 4.1: Skutečné umístění měřicího zařízení na střeše rodinného domu.

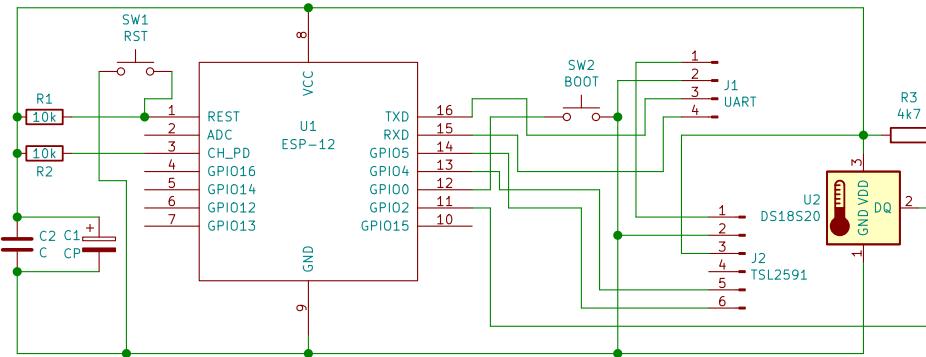
kapitoly zmíněných, osazených DPS. Jednotlivá měření se předávají po WiFi dalším součástem systému pomocí protokolu MQTT.

Jedna z DPS je umístěna na polici v žaluzií zatemňované místnosti v domě autora tak, aby na připojené senzory nedopadalo přímé sluneční světlo. Druhá je položena na střeše tohoto domu uvnitř krabičky s průhledným okénkem tak, aby byla elektronika krytá před povětrnostními vlivy, ale na senzor intenzity osvětlení i tak dopadalo světlo. Voděodolnost krabičky zajišťuje příšroubované víko se silikonovým těsněním a silikonem utěsněné kablové 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 4.1. Napájecí kabel je zatížen betonovou kostkou, aby se předešlo samovolnému pohybu zařízení po střeše vlivem silného větru.

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.

#### 4.1.1 Zapojení

Zapojení vychází z dokumentace součástek a jeho schéma je na obrázku 4.2. 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



OBRÁZEK 4.2: Schéma zapojení zařízení, která měří intenzitu osvětlení pomocí senzoru TSL2591 a teplotu vzduchu pomocí senzoru DS18B20. Hodnoty se přenáší přes WiFi.

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, jednak přes rezistor R3 na napájecí napětí jako pull-up. Napájecí vodiče sběrnice jsou přímo spojené s napájecími vodiči MCU.

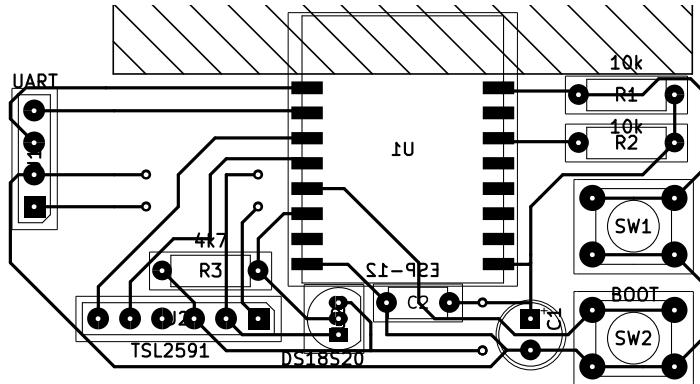
Na GPIO4 a GPIO5 jsou zapojeny po řadě datový a hodinový vodič sběrnice I<sup>2</sup>C, která slouží ke komunikaci se senzorem intenzity osvětlení. Ten dále slouží jako stabilizovaný zdroj napětí pro celé zařízení, součástí modulu je totiž lineární snižující zdroj napětí o velikosti 3,3 V.

#### 4.1.2 Deska plošných spojů (DPS)

DPS elektricky propjuje ostatní komponenty této zařízení a byla vyrobena po domácku. V opensource programu pro návrh elektronických zařízení KiCad byl sestaven obvod a návrh DPS (obrázek 4.3), jenž byl pomocí kancelářské tiskárny překlopeně vytisknán 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 měďí 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 vrtáku.

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



OBRÁZEK 4.3: 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.

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 zkонтrolována, jestli barva nebyla smyta a jestli je ještě potřeba leptat dále. Když už na desce zůstaly 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 lihu). Součástky byly osazeny na příslušná místa dle návrhu desky a schématu.

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

Modul ESP-12E vyvíjený společností Ai-Thinker 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. (AI-Thinker team, 2015) V popisovaném měřicím zařízení slouží tento modul ke zpracování dat naměřených pomocí senzorů teploty a intenzity osvětlení a jejich přenosu do dalších součástí systému.

#### 4.1.4 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 stejné sběrnici může být připojeno více teploměrů DS18B20, protože každý z nich má z výroby přidělené unikátní

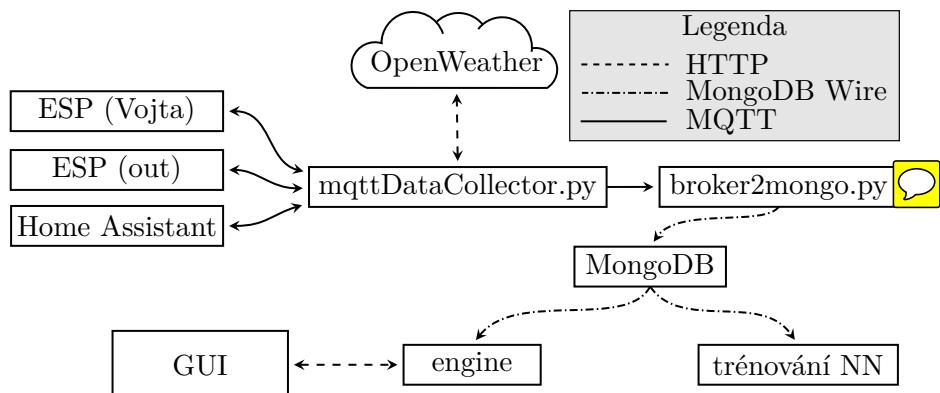
sériové číslo. Je dostupný v pouzdrech TO92 a SOP8 a nalezne využití ve stavebnictví při řízení vytápění, ventilace nebo klimatizace, měření teplot v budovách, v průmyslu při měření a monitorování teploty zařízení nebo procesů a jejich řízení. (Maxim Integrated Products, Inc., 2019) V měřicím zařízení poskytuje data o teplotě vzduchu.

#### 4.1.5 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 Industries, 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).

# 5 Sběr dat a komunikace komponent

Použitým algoritmům strojového učení je nutné dodat data sestávající z počtu hodnot příznaků (více v kapitole 3) 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 5.1: Schéma komunikace komponent systému automatického řízení žaluzí, tok dat mezi nimi a komunikační protokoly.

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 sbírájí příslušná zařízení<sup>2</sup>. Ostatní informace o počasí<sup>3</sup> 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 `broker2mongo.py`, který je uloží do databáze MongoDB (sekce 5.4). Na obrázku 5.1 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>rychlosť (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)

## 5.1 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 Raspberry Pi (kapitola 4) 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 4.1) i systém domácí automatizace a po jejím přijetí odešlou aktuální hodnoty jimi sledovaných příznaků do odpovídajících témat dle tabulky 5.1. 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 5.4. 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 5.1: Názvy příznaků a MQTT téma, kam se hodnoty příznaků odesírají, a která odebírá skript `mqttDataCollector.py`.

## 5.2 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 Application Programming Interface (rozhraní pro programování aplikací) (API), které nabízí výrobce. Komunikace s API 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 v jazyce Python, která mapuje akce pro manipulaci

se žaluzí a její stav na metody a proměnné objektu, který je tak modelem žaluzie. Dále zpřístupňuje některé obecnější metody pro práci s API jako je například zjištění všech montáží zákazníka či zjištění všech dostupných zařízení v rámci konkrétní montáže.

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 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 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, 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 zamítl. 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 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. 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 5.1), 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 5.1. Naopak po přijetí

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

zprávy backendem se v závislosti na jejím obsahu odešle požadavek API na změnu stavu žaluzie. Zpráva má stejnou strukturu jako zpráva v úryvku kódu 5.1, ale obsahuje jen jeden z klíčů. Navíc může obsahovat speciální klíč `testing` s hodnotou datového typu `boolean`, 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 5.1: Struktura zprávy o stavu žaluzie, která je všem klientům zasílána každých 15 s. Klíč `position` označuje výšku vytažení žaluzie (0 – zavřeno, 100 – otevřeno), obdobně hodnota `tilt` vyjadřuje naklopení lamel.

### 5.3 Využití služby OpenWeather

Skript `mqttDataCollector.py` kromě měřených příznaků a stavu žaluzií zjíšťuje také odhad počasí a jeho předpověď. Tato data poskytuje služba 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 Českým hydrometeorologickým ústavem 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, hodinové předpovědi teploty na následující 3 hodiny a předpovědi nejvyšší denní teploty.

### 5.4 Databáze

Data, která se sbírají pomocí skriptu `mqttDataCollector.py` v pětiminutových intervalech nebo na základě změny stavu žaluzie uživatelem, ukládá skript `broker2mongo.py` do databáze MongoDB (MongoDB, Inc., 2021). 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 5.2.

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

```

    "year_day": int,
    "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 5.2: 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 čárkou, boolean - pravdivostní hodnota, int - celé číslo)

Uložená data se vyplňují 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“. Vyhodnocení sběru dat je uvedené v sekci 8.1.

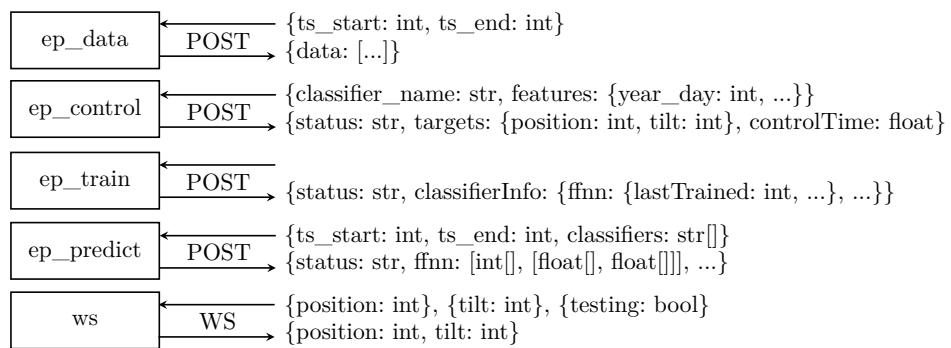
## 5.5 Webový server pro komunikaci s GUI

Systém má webové uživatelské rozhraní poskytované *Next.js* serverem (kapitola 7). Jeho propojení s jádrem systému (backend) je řešeno pomocí frameworku Tornado v jazyce Python. Tento webserver tak tvoří rozhraní mezi GUI a databází, vsemi 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 5.2) 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` (jeden vzorek), vrací `controlTime` - dobu trvání predikce, `status: ok` a navržené hodnoty řízení (jako v úryvku kódu 5.1)).
- `ep_train` spustí přetrenování NS (sekce 6.5) na všech dostupných datech.

- `ep_predict` slouží k predikci řízení regresorů předaných v parametru `classifiers` pro reálná data (přijímá začátek a konec intervalu stejným způsobem jako `ep_data`, načte všechny vzorky z tohoto intervalu). Pro každý z regresorů vrací pod klíčem s jejich názvem dobu predikce a seznam dvojic doporučeného řízení a odpovídajích časových známek.
- `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 5.2.



OBRÁZEK 5.2: Schéma endpointů backendu systému pro automatické ovládání žaluzí a datové struktury přenášené v těle jednotlivých požadavků a odpovědí včetně metody.

V posledním případě se jedná o protokol WebSockets.

# 6 Automatické ovládání žaluzií

Pro automatické ovládání žaluzií byly navrženy tři regresory, které odhadují vhodné nastavení žaluzie (výška vytažení (position) a míra naklopení (tilt)) na základě 15 místního příznakového vektoru (případně časové posloupnosti těchto vektorů). Použité regresory byly nazvány „If-else“, „FFNN“ a „LSTM“ podle svého principu. Tato kapitola se zabývá jejich návrhem, specifickými vlastnostmi, které z nich vyplývají, a jeho metodami.

Poslední dva zmíněné regresory (FFNN a LSTM) jsou neuronové sítě (NS). K práci s jejich modely bylo využíváno rozhnávání Keras v jazyce Python, které umožňuje jejich učení, vyhodnocování i následné použití pro regresi (Keras, 2022). Při volbě jejich vnitřní struktury a parametrů učení se slepě zkoušely natrénovat sítě se všemi možnými kombinacemi určitých parametrů a následně se vybrala nejlepší z nich ve smyslu minimální střední kvadratické odchylky (MSE) na testovací datové sadě (sekce 6.1). Parametry a prohledávané hodnoty jsou uvedeny v sekcích 6.3 a 6.4.

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 množinu. 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 NS se celá vstupní data zpracovávají najednou, což má pozitivní vliv na rychlosť.

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

## 6.1 Příprava dat pro strojové učení

Naměřená data jsou uložená v databázi (kapitola 5) 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 očekávaného intervalu jejich hodnot uvedeného v tabulce 3.1 v kapitole 3. Pro regresor FFNN každý vzorek v datových sadách odpovídá 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.

Datová sada se disjunktně dělí na 3 menší datové sady. Každá z nich slouží k různým účelům při vývoji systému.

**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. Neslouží přímo pro učení NS - ověřuje se na ní, zda síť dobře zobecňuje a nedochází k přetrénovalení na trénovací datovou sadu (overfitting). Vybíral se vždy doposud nejlepší model podle hodnoty ztrátové funkce vyhodnocené na validační datové sadě.

**Testovací datová sada** sestává z 10% náhodně vybraných vzorků. Byla použita pro vyhodnocení úspěšnosti učení sítí s různou strukturou při hledání té nejvhodnější (sekce 6.3 a 6.4) a také pro celkové vyhodnocení regresorů a vlivu příznaků na odhad regresorů (sekce 8.3 až 8.5).

## 6.2 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 navržených metod strojového učení s rozhodováním dle neměnných pravidel, které se v současnosti v praxi v domácí automatizaci obvykle používá (Apple Inc., 2022; OpenHAB Foundation, 2022; Home Assistant, 2022).

Pravidla byla konstruována podle slovy formulovaných požadavků uživatele na stav žaluzí v závislosti na aktuálních podmínkách (vyjádřených příznamkovým 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

<sup>1</sup>Den znamená, že je vnější intenzita osvětlení vyšší než nastavená mez a zároveň je čas vyšší než nastavený čas vstávání v 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 vytápěním 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  $18^{\circ}\text{C}$ .

je teplota vnějšího vzduchu vyšší než obě meze, žaluzie má být ve stejné 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.

2. V noci: Žaluzie úplně naklopená, vyjímkou je případ, kdy je vnitřní teplota nižší než nastavená mez - to může znamenat, že se uživatel snaží větrat 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.

---

<sup>5</sup>Nižší hranice pásma 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  $\langle 321, 366 \rangle \cup \langle 1, 79 \rangle$

### 6.3 Dopředná neuronová síť jako regresor (FFNN)

První regresor založený na neuronových sítích byl nazvaný „FFNN“ podle anglického sousloví *Feedforward Neural Network*, které označuje dopřednou neuronovou síť. Při volbě její vnitřní struktury byly prohledávány všechny kombinace hodnot parametrů uvedených v tabulce 6.1. Všechny vrstvy sítě

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

TABULKA 6.1: 4 parametry dopředné neuronové sítě a jejího učení, jejichž kombinace se prohledávaly za účelem získání nejlepší sítě pro úlohu regrese stavu žaluzie podle 15 příznaků. Parametr struktura sítě reprezentuje posloupnosti (v hranatých závorkách) počtu neuronů v jednotlivých vrstvách sítě. MSE značí střední kvadratickou odchylku.

jsou plně propojené. Neurony v nich využívají 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.

### 6.4 Rekurentní neuronová síť jako regresor (LSTM)

Tento regresor je rekurentní neuronová síť, která používá *Long Short-Term Memory* (LSTM) buňky (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

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

TABULKA 6.2: 4 parametry rekurentní neuronové sítě a jejího učení, jejichž kombinace se prohledávaly za účelem získání nejlepší sítě pro úlohu regrese stavu žaluzie podle 15 příznaků. Parametr struktura sítě reprezentuje posloupnosti (v hranatých závorkách) počtu neuronů v jednotlivých vrstvách sítě. MSE značí střední kvadratickou odchylku.

funkcí je v tomto případě ve všech vrstvách hyperbolický tangens a funkce sigmoid pro rekurentní krok. Výstupní vrstva je stejná jako v případě NS regresoru FFNN.

## 6.5 Přetrénování neuronových sítí (retraining)

Přetrénováním se rozumí opakované trénování NS regresorů na všech v daný okamžik dostupných datech. Přetrénování probíhá na již existujícím modelu, který se jen dále upravuje a zpřesňuje. Automaticky se v backendu plánuje na půlnoc každého dne, ale je ho možné také ručně spustit v záhlaví každé ze stránek GUI. Díky tomu se systém může přizpůsobovat novým okolnostem v datech, kterými mohou být zejména nové návyky uživatele.

Aby bylo přetrénování rychlé, používá se méně epoch a větší batch size než při běžném trénování. Obě NS se přetrénovávají s batch size 128, FFNN 50 epoch, LSTM pak 2 epochy.

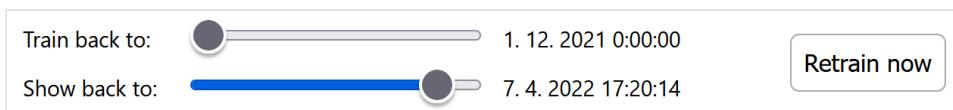


# 7 Grafické uživatelské rozhraní

K ovládání navrhovaného systému slouží grafické uživatelské rozhraní (GUI) implementované jako webová aplikace s využitím frameworku Next.js (Vercel, 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 - vyhodnocování výstupu regresorů pro uživatelem libovolně nastavené hodnoty příznaků (simulace výstupu regresorů pro zadанé podmínky);
- Control - ruční řízení žaluzie;
- Live - grafické zobrazení hodnot skutečného řízení a řízení získaných z jednotlivých regresorů společně s naměřenými daty.

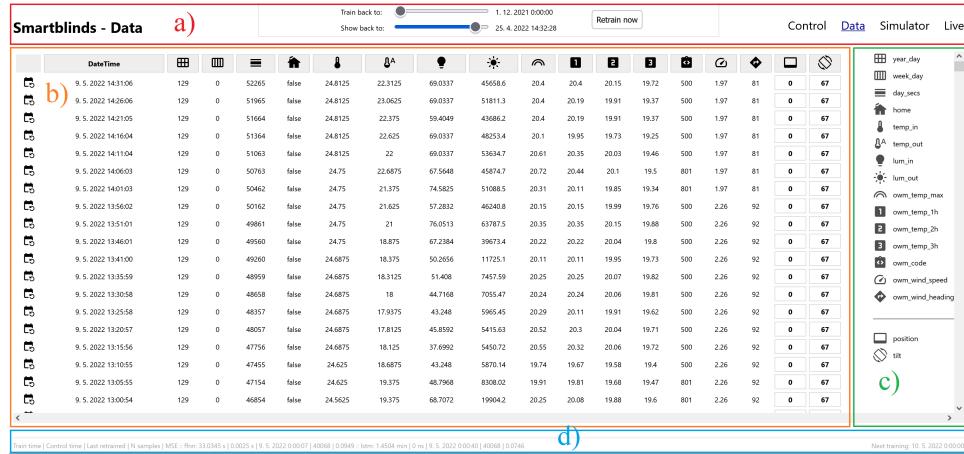
Celá aplikace se skládá z komponent psaných v jazyce TypeScript, Next.js se správcem balíků Yarn se pak stará o jejich komplikaci do JavaScriptu a následný přenos do prohlížeče uživatele. Všechny stránky mají společné záhlaví (obrázek 7.1, v obrázcích jednotlivých stránek (obrázky 7.2, 7.3, 7.5 a 7.6) ~~a~~), které obsahuje ovládací panel pro nastavení intervalu pro zobrazení dat a pro trénování neuronových sítí regresorů (kapitola 6), zároveň je možné pomocí tlačítka spustit přetrénování na zvolených datech (sekce 6.5). V pravém rohu záhlaví je pak menu, které slouží k přepínání stránek v rámci GUI.



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 přetrénování na zvolených datech.

## 7.1 Stránka Data

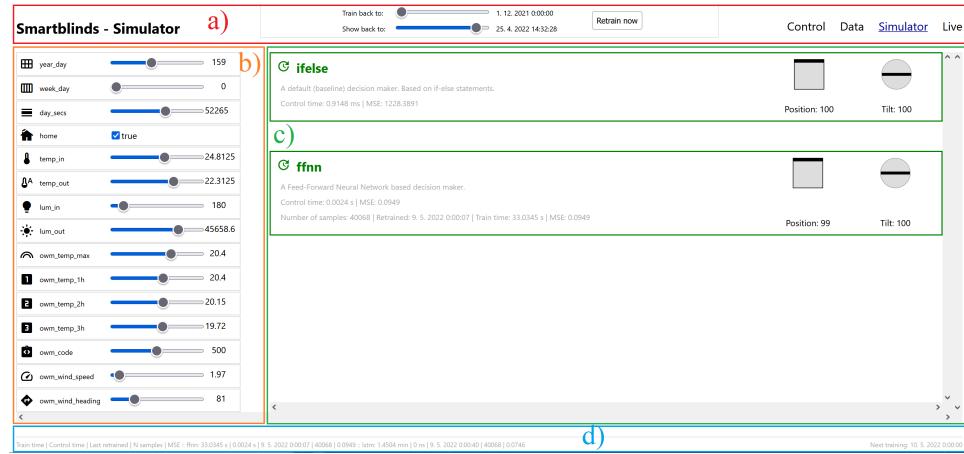
Většinu stránky Data (obrázek 7.2) zaujímá tabulka (b), ve které jsou v řádkách uvedeny jednotlivé vzorky naměřených dat s hodnotami všech příznaků a řízení žaluzií uživatelem společně s časem pořízení vzorku. Vlevo



OBRÁZEK 7.2: Stránka Data v grafickém uživatelském rozhraní systému pro automatické ovládání žaluzí s vyznačenými částmi pomocí barevných obdélníků a) – d)

od každého řádku je navíc ikona, která reprezentuje příčinu pořízení vzorku: pro periodické vzorky, pro vzorky pořízené na základě změny stavu žaluzie (toto chování je popsáno v kapitole 5). Vzorky jsou seřazeny chronologicky od nejmladších k nejstarším a zobrazují se pouze ty, které jsou z časového intervalu daného nastavením v záhlaví aplikace (a), 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 (c)). 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



OBRÁZEK 7.3: Stránka Simulator v grafickém uživatelském rozhraní systému pro automatické ovládání žaluzí s vyznačenými částmi pomocí barevných obdélníků a) – d)

Tato stránka slouží pro ruční vyhodnocování výstupu regresorů v závislosti na nastavených hodnotách příznaků. Je rozdělena do dvou částí, vlevo je sloupec vstupních prvků (b)), vpravo pak rámce jednotlivých regresorů s



Position: 33      Tilt: 91

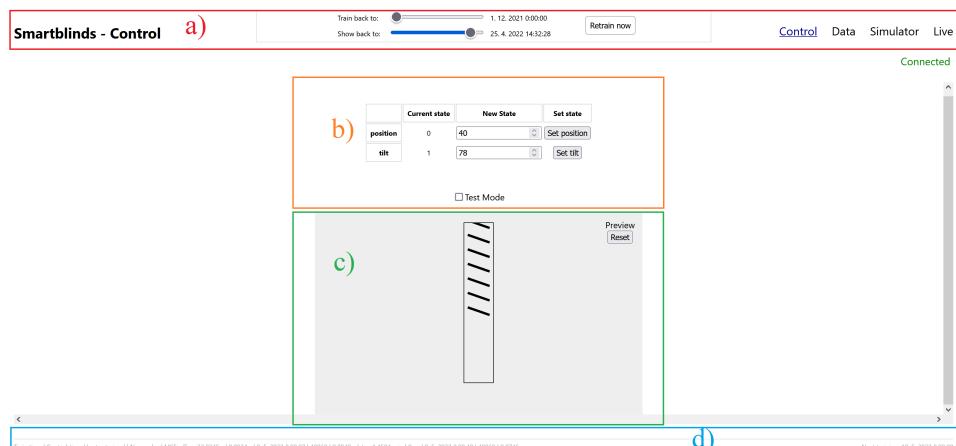
OBRÁZEK 7.4: Příklad vizualizace výstupu regresoru v úloze automatického ovládání žaluzie na základě 15 příznaků.

jejich výstupem a jeho jednoduchou vizualizací (c)). Výška vytažení žaluzie (Position) je reprezentována částečným vybarvením obdélníku shora, úhel naklopení lamel (Tilt) pak rotací čáry uvnitř kruhu (příklad je na obrázku 7.4).

Vzhledem k povaze vstupních prvků (b)), 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 odesle požadavek na endpoint `ep_control` backendu systému (sekce 5.5), v odpovědi se vrátí výstup regresoru, který je poté zobrazen.

### 7.3 Stránka Control



OBRÁZEK 7.5: Stránka Control v grafickém uživatelském rozhraní systému pro automatické ovládání žaluzí s vyznačenými částmi pomocí barevných obdélníků a) – d)

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, b)) a vizualizace (dole, c)).

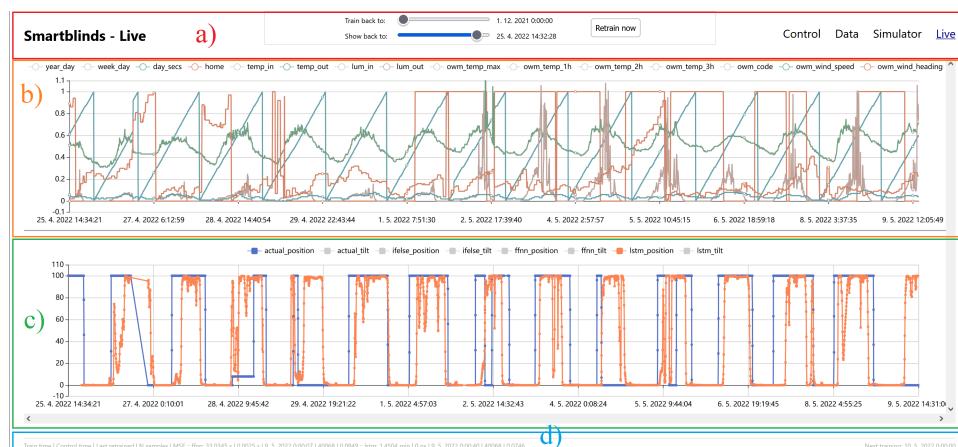
Tabulka (b)) má tři sloupce, v prvním z nich se zobrazuje aktuální stav, ve druhém jsou pole pro nastavení nového stavu a ve třetím tlačítka pro odeslání požadavku na změnu stavu. Každý řádek je pro jednu stavovou proměnnou žaluzie.

Vizualizace (c) 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. Uživatele 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 dvouozměrnou 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 5.5) 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



OBRÁZEK 7.6: Stránka Live v grafickém uživatelském rozhraní systému pro automatické ovládání žaluzií s vyznačenými částmi pomocí barevných obdélníků a) – d)

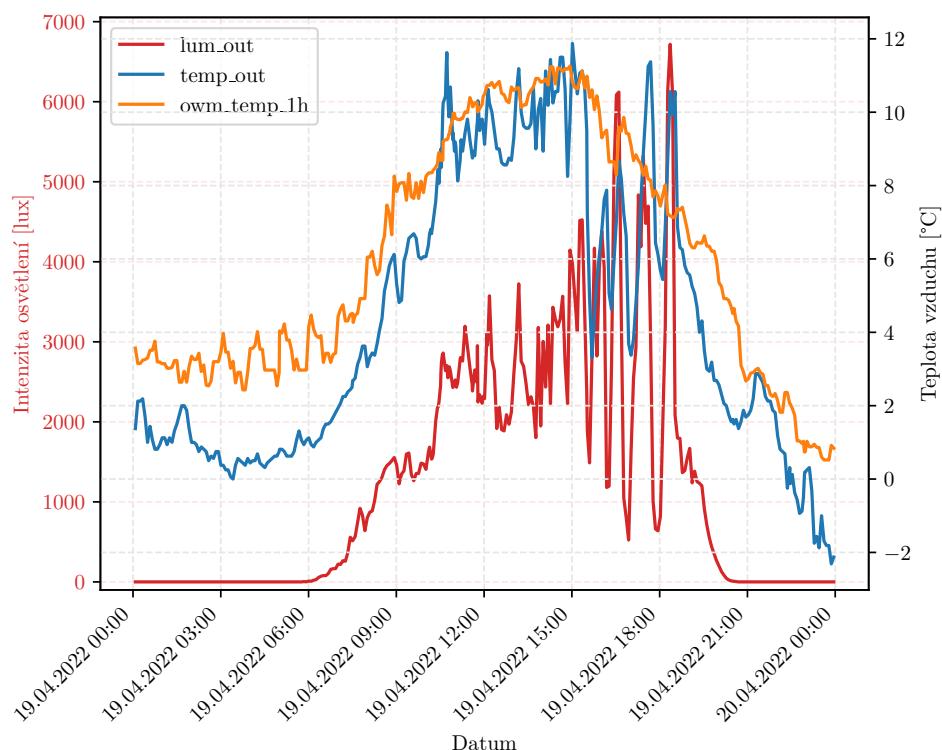
Poslední ze 4 stránek GUI je stránka Live, která slouží k živému vyhodnocování řízení doporučeného jednotlivými regresory na reálných datech. Skládá se ze dvou grafů uspořádaných pod sebou. V horním grafu ((b)) jsou zobrazena naměřená data, v dolním (c) pak řízení žaluzie (manuální řízení i doporučené 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 5.5). 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 (a), *Show back to:*.

# 8 Vyhodnocení

## 8.1 Vyhodnocení sběru dat a spolehlivosti

Pro účely této práce se sbírala data (kapitola 5) od 9. prosince 2021 17:38 (UTC) do 2. května 2022 11:59 (UTC). Za tu dobu bylo sesbíráno 38128 vzorků dat, sestávajících z 15 hodnot příznakových veličin, 2 hodnot stavu žaluzie, časové známky, důvodu pořízení vzorku a aktivity testovacího režimu. Z toho bylo 36306 pořízeno periodicky v pětiminutových intervalech a 1822 na základě události změny stavu žaluzie - ruční ovládání uživatelem. Příklad 3 sbíraných veličin na časovém úseku jednoho dne (19. dubna 2022) je v grafu na obrázku 8.1.



OBRÁZEK 8.1: Graf sesbíraných hodnot 3 vybraných příznakových veličin z celkových 15. Zobrazena je pouze část dat ze dne 19. dubna 2022

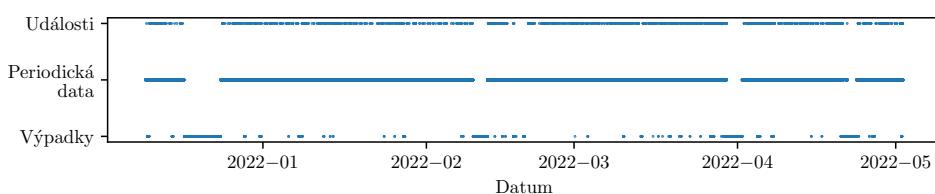
V průběhu sběru dat došlo ke 125 výpadkům při kterých bylo ztraceno 5240 vzorků. Nejčastějším důvodem výpadku byly problémy s připojením k API výrobce žaluzie (Somfy, celkem 105 výpadků). Mezi další příčiny patří výpadky senzorů (odpojení), plný disk databázového serveru (MongoDB) a chyba v systému domácí automatizace (Home Assistant). Poslední dvě

příčiny společně způsobily ztrátu asi 81 % měření z celkového počtu ztracených měření. Největší z výpadků, způsobený Home Assistantem, vznikl ještě před implementací monitoringu přicházejích MQTT zpráv a byl tak odhalen pozdě. Jednou došlo k výpadku připojení k DNS serverům poskytovatele připojení k internetu a nebylo kvůli tomu možné zjistit stav žaluzie. Počet výpadků, jejich důvod a počet při nich ztracených měření shrnuje tabulka 8.1.

Důvod výpadku	Počet výpadků	Počet ztracených vzorků
Somfy	105	383
Senzory	14	605
MongoDB	4	2244
DNS	1	27
Home Assistant	1	1981
Celkem	125	5240

TABULKA 8.1: Výpadky součástí systému automatického ovládání žaluzií při sběru dat a výpadky třetích stran, na kterých je systém závislý.

Vzorky podle důvodu sběru (událost, periodické) a výpadky v závislosti na čase jsou v grafu na obrázku 8.2. Každý vzorek je vyobrazen jako bod podle důvodu svého vzniku (Periodická data - vzorky, které byly pořízeny v pětiminutových intervalech; Události - vzorky získané na základě ruční změny stavu žaluzie uživatelem; Výpadky - vzorky, které měly být periodicky pořízeny, ale nebyly).



OBRÁZEK 8.2: Graf druhu sesbíraných vzorků dat a výpadků.

## 8.2 Volba struktury NS a parametrů učení

Na datech sesbíraných ve dnech od 1. prosince 2021 do 17. března 2022 bylo natrénováno 60 dopředných NS (regresor FFNN, sekce 6.3) a 32 rekurrentních NS (regresor LSTM, sekce 6.4). Dále budeme uvažovat pouze síť trénované se ztrátovou funkcí MSE, protože ostatní se trénovaly s využitím kosinové vzdálenosti, která je pro danou úlohu nevhodná (nízkou ztrátu totiž vykazují nejen stavy blízko požadovaným, ale také ty, které mají blízký poměr mezi dvěma stavovými veličinami žaluzie), což se potvrdilo i při tomto experimentu, kdy MSE predikcí na testovací datové sadě (sekce 6.1) u sítí trénovaných se ztrátovou funkcí MSE a s kosinovou podobností se lišila o

řád jak v případě dopředné, tak i v případě rekurentní NS. Struktura sítě, parametry použité při jejím trénování a MSE na testovací datové sadě jsou v tabulkách 8.2 a 8.3.

Na základě těchto výsledků byly pro další použití v navrhovaném systému automatického ovládání žaluzí zvoleny následující parametry pro NS a jejich trénování:

- Dopředná NS (FFNN):
  - 3 skryté vrstvy (po řadě 20, 15 a 10 neuronů);
  - 500 epoch trénování;
  - velikost dávky (batch\_size) 32.;
  - ztrátová funkce (loss) MSE.
- Dopředná NS (LSTM):
  - 2 skryté vrstvy (po řadě 64 a 64 neuronů);
  - 200 epoch trénování;
  - velikost dávky (batch\_size) 16;
  - ztrátová funkce (loss) MSE.

### 8.3 Vliv jednotlivých příznaků na predikci

Důležitost příznaků v regresorech byla vyjádřena jako průměrný rozdíl MSE jestliže jsou hodnoty daného  příznaku náhodně promíchány mezi vzorky (20 realizací) a MSE na datech v původním pořadí. Pro regresor FFNN jsou první 4 nejdůležitější příznaky day\_secs, lum\_in, lum\_out a year\_day, pro regresor LSTM pak day\_secs, lum\_in, year\_day a owm\_wind\_speed. Mezi málo důležité příznaky patří u obou regresorů všechny teploty (temp\_in, temp\_out, owm\_temp\_xh pro  $x = 1, 2, 3$ ). Grafy důležitostí všech příznaků pro jednotlivé regresory jsou na obrázcích 8.3 a 8.4.

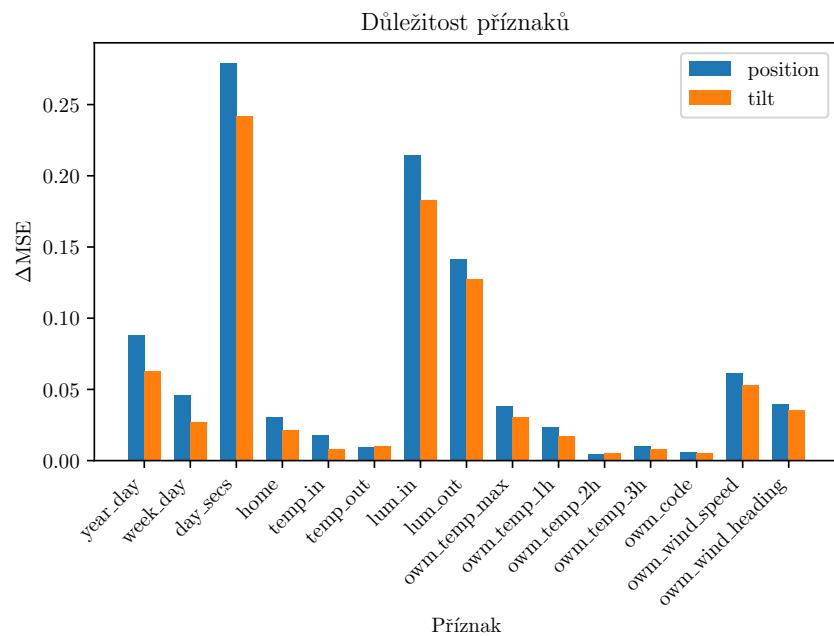
	Struktura	E	BS	MSE
1	[20, 15, 10]	500	32	0,01571
2	[20, 15, 10]	700	32	0,01615
3	[20, 15, 10]	500	16	0,01661
4	[20, 15, 10]	200	32	0,01867
5	[10, 15, 10]	700	16	0,01875
6	[10, 15, 20]	700	16	0,01947
7	[10, 15, 10]	500	16	0,01955
8	[10, 15, 20]	500	32	0,02004
9	[15, 10]	700	32	0,02071
10	[20, 15, 10]	700	16	0,02119
11	[15, 10]	700	16	0,02152
12	[10, 15, 20]	500	16	0,02155
13	[10, 15, 20]	200	16	0,02213
14	[15, 10]	500	16	0,02214
15	[10, 15, 10]	700	32	0,02297
16	[10, 15, 10]	500	32	0,02451
17	[10, 15, 20]	700	32	0,02478
18	[20, 15, 10]	200	16	0,02543
19	[10, 15]	700	32	0,02746
20	[15, 10]	200	16	0,02752
21	[10, 15, 10]	200	16	0,02914
22	[10, 15]	700	16	0,02925
23	[15, 10]	500	32	0,03059
24	[10, 15]	200	16	0,03144
25	[10, 15]	500	32	0,03357
26	[10, 15]	200	32	0,04301
27	[10, 15]	500	16	0,04732
28	[10, 15, 20]	200	32	0,05163
29	[15, 10]	200	32	0,05230
30	[10, 15, 10]	200	32	0,05512



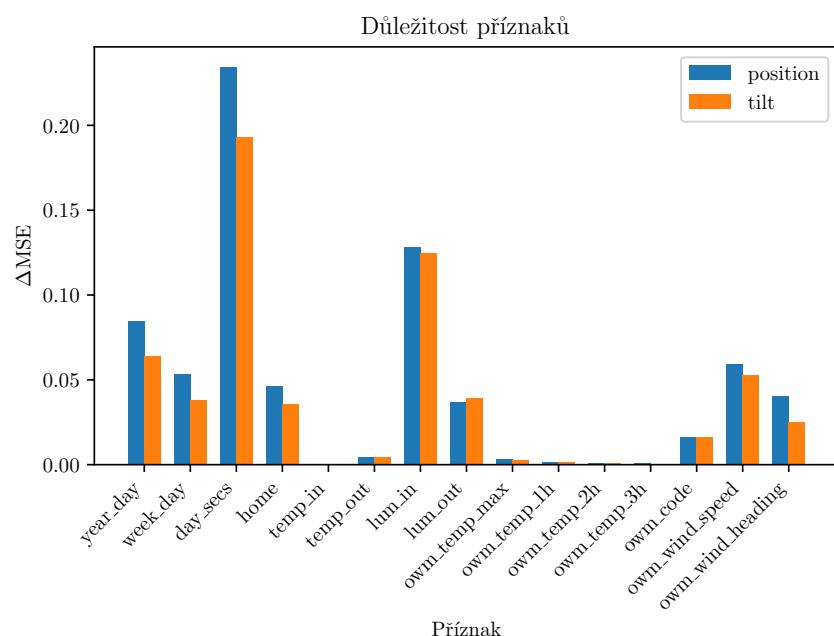
TABULKA 8.2: Výsledky prohledávání parametrů dopředné neuronové sítě a jejího učení pro použití v systému automatického ovládání žaluzií. Sloupce: Struktura sítě, E - počet epoch trénování, BS - velikost dávky (batch size), MSE - střední kvadratická odchylka na testovací datové sadě.

	Struktura	E	BS	MSE
1	[64, 64]	200	16	0,00348
2	[64, 32]	200	16	0,00357
3	[64, 16, 16]	200	16	0,00437
4	[64, 64]	200	64	0,00500
5	[64, 32]	100	16	0,00516
6	[64, 32]	200	64	0,00562
7	[64, 16]	200	16	0,00578
8	[64, 16]	200	64	0,00653
9	[64, 16, 16]	100	16	0,00657
10	[64, 16, 16]	200	64	0,00683
11	[64, 64]	100	16	0,00690
12	[64, 16]	100	64	0,00810
13	[64, 64]	100	64	0,00916
14	[64, 32]	100	64	0,00981
15	[64, 16]	100	16	0,00998
16	[64, 16, 16]	100	64	0,01058

TABULKA 8.3: Výsledky prohledávání parametrů rekurentní neuronové sítě a jejího učení pro použití v systému automatického ovládání žaluzí. Sloupce: Struktura sítě, E - počet epoch trénování, BS - velikost dávky (batch size), MSE - střední kvadratická odchylka na testovací datové sadě.



OBRÁZEK 8.3: Permutační důležitost příznaků v regresoru založeném na dopředné neuronové síti v úloze automatického ovládání žaluzie.



OBRÁZEK 8.4: Permutační důležitost příznaků v regresoru založeném na rekurentní neuronové síti v úloze automatického ovládání žaluzie.

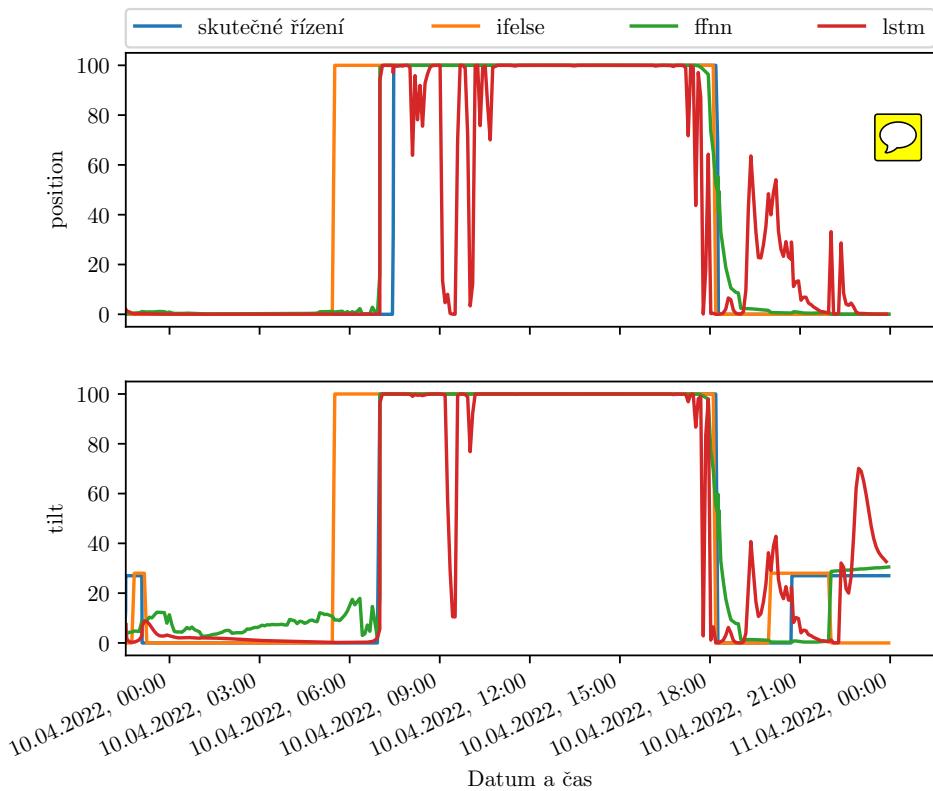
## 8.4 Porovnání regresorů

Porovnání regresorů použitých v systému automatického ovládání žaluzie (jestliže to vzhledem k jejich návrhu dává smysl) z hlediska počtu parametrů (u Ifelse ručně konfigurované, u FFNN a LSTM odhadnuté pomocí algoritmů strojového učení), doby učení, doby predikce výstupu a dosažené MSE na testovací datové sadě je uvedené v tabulce 8.4. Regresor Ifelse má nejmenší počet parametrů, je 2. v rychlosti predikce a má nejvyšší MSE. Největší počet parametrů, dobu trénování i predikce a nejnižší MSE má regresor LSTM.

	Ifelse	FFNN	LSTM
Počet parametrů	17	817	53634
Doba predikce [ms]	250	30	430
Doba trénování [s]	-	1839	12129
MSE	0,120	0,0246	0,00426

TABULKA 8.4: Porovnání vybraných vlastností regresorů.  
Doba predikce platí pro 400 vzorků.

Ukázka predikce obou stavových veličin žaluzie jednotlivými regresory na základě dat sesbíraných 10. dubna 2022 a porovnání se skutečným řízením je v grafu na obrázku 8.5.



OBRÁZEK 8.5: Porovnání predikce 3 regresorů a skutečného řízení na datech z 10. dubna 2022

## 8.5 Vyhodnocení vlivu retrainingu

Vyhodnocovat retraining má smysl jen u 2 z regresorů, které využívají strojové učení (FFNN a LSTM). Pro každý z nich se porovnaly 2 modely. První z modelů se trénovaly na datech do konce února, druhé pak vznikly z prvních opakovaným trénovaním na datech do konce dubna. Tabulky 8.5 a 8.6 porovnávají MSE od referenčních dat z testovací datové sady pro každý z regresorů celkově i pro každou stavovou proměnnou zvlášť.

	celkově	position	tilt
březen	0,0962	0,1156	0,0769
květen	0,0246	0,0329	0,0163

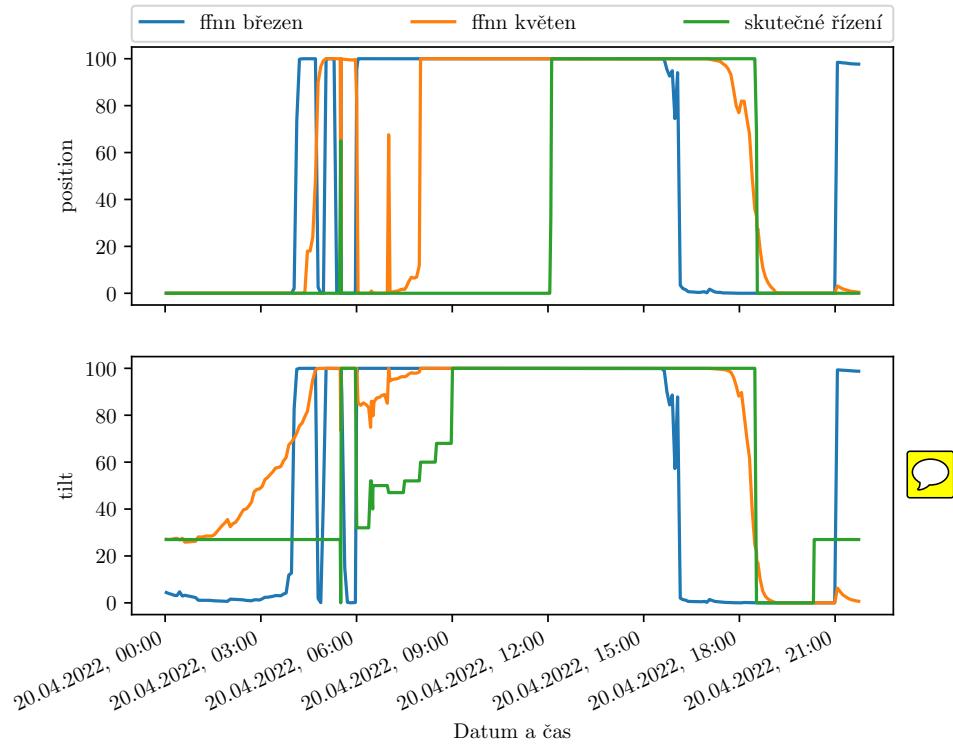
TABULKA 8.5: Porovnání středních kvadratických odchylek dvou modelů dopředné neuronové sítě natrénovaných na 2 různých datových sadách z období před daným měsícem.



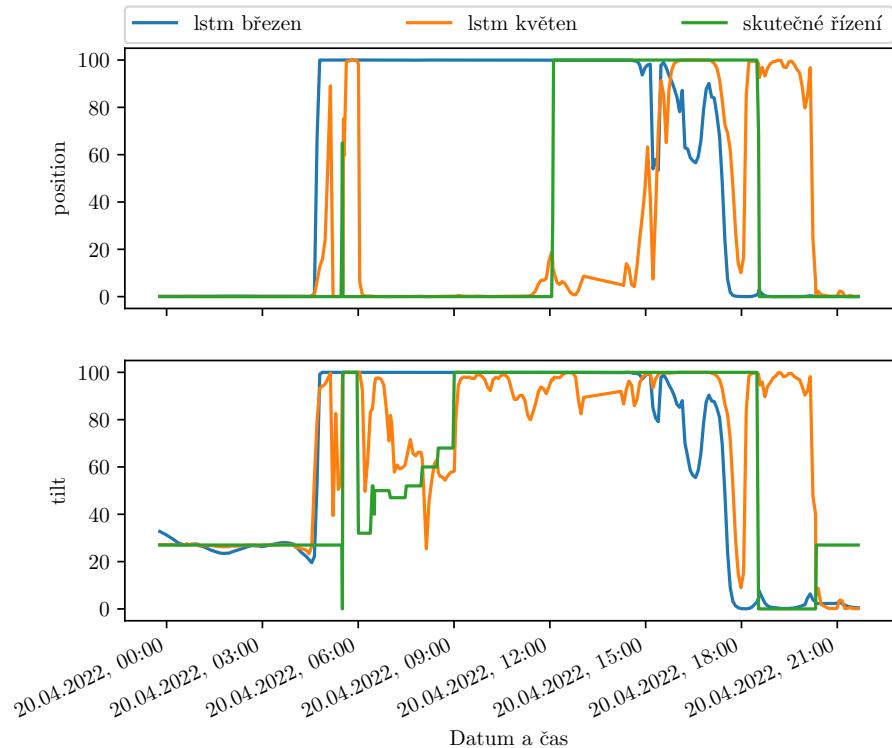
	celkově	position	tilt
březen	0,05911	0,07470	0,04351
květen	0,00426	0,00316	0,00536

TABULKA 8.6: Porovnání středních kvadratických odchylek dvou modelů rekurentní neuronové sítě natrénovaných na 2 různých datových sadách z období před daným měsícem.

Příklad predikce ovládání žaluzií (pro každou z proměnných ve vlastní soustavě souřadné) jednotlivými modely dne 20. dubna 2022 je na obrázcích 8.6 a 8.7.



OBRÁZEK 8.6: Porovnání predikce 2 modelů dopředné neuronové sítě a skutečného řízení na datech z 20. dubna 2022



OBRÁZEK 8.7: Porovnání predikce 2 modelů rekurentní neuronové sítě a skutečného řízení na datech z 20. dubna 2022

## 9 Diskuze

Cílem této práce bylo navrhnout systém pro automatické ovládání žaluzie na základě preferencí uživatele. Pro základní rozhodovací systém (Ifelse) se zvolil přístup pro vyjádření preferencí uživatele zřetězením jednoduchých pravidel. Další dva regresory navržené pro použití v systému jsou neuronové sítě (FFNN a LSTM), pro jejichž učení je nutné dodat data o skutečném nastavení žaluzí uživatelem a příznacích, na jejichž základě se o nastavení mohl uživatel rozhodovat.

Data se sbírala pomocí měřicích zařízení vlastní kostrukce a z některých zdrojů dostupných na internetu a ukládala se do databáze pro pozdější použití. V průběhu sběru dat docházelo opakovaně k výpadkům. Většina jich byla způsobena nedostupností služby 3. strany, ale největší ztráty měření způsobily vlastní provozované součásti. Jako rizikové se jevilo umístění jednoho z měřicích zařízení na střeše domu, kde bylo vystavené nízkým i vysokým teplotám a zejména dešti. Před zahájením sběru kvůli nedostatečné těsnosti okénka v krabičce došlo k vniknutí vody do zařízení a zničení vstupně-výstupních portů MCU. Po přetěsnění už k podobnému problému nedošlo. Na eliminaci velkých ztrát by mělo pozitivní vliv použití komplexnějšího monitoringu součástí, protože by výpadek bylo možné rychleji odstranit a případně mu i předejít. V průběhu vývoje systému byl nasazen monitoring funkčnosti všech zdrojů dat. Pokud by se zanedbaly výpadky 3. stran a výpadky, které by bylo možné kvůli monitoringu výrazně zkrátit nebo odstranit, ztraceno by bylo jen 1,4 % měření.

Jedním ze zdrojů dat byl systém domácí automatizace, který poskytoval informace o nastavení žaluzie z API výrobce, protože se v počátcích vývoje nedařilo získání přístupu k API běžným způsobem. Později se to však podle nedokumentovanou cestou a místo systému domácí automatizace by tak bylo možné používat knihovnu Pymfy pro přímý přístup k API.

**TODO: Doplnit volbu struktury po zopakování**

Každý z příznaků může mít různě významný vliv na hodnoty odhadované regresorem. Pro kvantifikaci tohoto vlivu byla využita metoda *Permutation Feature Importance* (kapitola 3). Výsledky (sekce 8.3) napovídají, že by mohlo být možné redukovat příznakový vektor, protože některé příznaky mají na odhad zanedbatelný vliv a úloha by se tak zjednodušila.

Jeden z nejdůležitějších příznaků je v obou případech NS příznak `lum_in`. Problémem tohoto příznaku je skutečnost, že je silně ovlivňován nastavením žaluzie. V případě reálného nasazení systému tak hrozí, že by sítě nemusely vygenerovat žádný akční zásah. Přestože je tedy příznak důležitý pro predikci v navržených modelech, mohlo by být lepší ho vynechat z příznakového vektoru.

Důležitou vlastností navrhovaného systému je retraining modelů na nových datech, která jsou průběžně sbírána (sekce 8.5). Za účelem zjištění, jaký vliv na odhad poskytovaný regresory má právě přetrénování, byly pro každou z použitých NS porovnány 2 modely, z nichž jeden přetrénováním druhého na čerstvějších data (mladší model). Na testovací datové sadě byl patrný výrazný rozdíl v MSE ve prospěch modelů s čerstvějšími datech (tabulky 8.5 a 8.6), v ukázce predikce starsích a novějších modelů na reálných datech (obrázky 8.6 a 8.7) je u těchto modelů pak zřejmá reakce na zvýšené riziko přehrátí interiéru v dopoledních hodinách, kdy by květnové modely narozdíl od březnových ponechaly žaluzii zataženou a sklápěly by lamely.

NS doporučované řízení není přímo vhodné pro přímé použití. Žaluzie by příliš často nevhodně měnily své nastavení a pro uživatele by používání systému nemuselo být pohodlné. Pro vylepšení poskytovaných výsledků by mohlo být vhodné nasbírat větší množství dat a dále zkoumat možné způsoby trénování NS a jejich strukturu, případně řízení vhodně filtrovat.



## 10 Závěr

V rámci této práce byl vyvinut systém, který v reálném čase sbírá hodnoty příznaků a aktuální uživatelské nastavení žaluzie, pomocí 3 různých regresorů odhaduje vhodné nastavení žaluzie a umožňuje řízení generované jednotlivými regresory porovnat. NS se v systému periodicky přetrénovávají (retraining), což má pozitivní vliv na přesnost jejich odhadu, vyskytnou-li se v datech nové skutečnosti.

Hardware součásti použité v měřicích zařízeních se osvědčily, stejně jako datové spojení s ostatními součástmi systému. Bylo by vhodné monitorovat všechny součásti i jejich spojení a případné problémy nejlépe automaticky odstraňovat, nebo hlásit uživateli. V systému se uživateli hlásí jen to, když nepřijdou data z některého zdroje dat v okamžiku pořizování vzorku.

V úloze automatického ovládání žaluzií může jejich stav záviset nejen na aktuálních podmínkách ale také na podmínkách, které panovaly v historii. Kvůli tomu je vhodné využít regresor, který historické podmínky uvažuje při svém odhadu. V této práci tuto vlastnost splňuje regresor založný na rekurentní neuronové síti s LSTM buňkami.

Přesto, že některé veličiny mohou být silně svázány s odhadovanými, nemusí být dobrými příznaky pro reálné využití, pokud je odhad řízení při jeho použití zpětně ovlivňuje.

Díky dalšímu sběru dat společně s přetrénováním NS by regresory mohly do budoucna poskytovat postupně lepší výsledky. V případě skutečného nasazení by bylo nutné ošetřit konfliktní stavy uživatelem vyžádaného nastavení žaluzie a systémem odhadnutým nastavením tak, aby interakce se systémem byla pro uživatele přívětivá.

# 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] Leo Breiman. „Random Forests“. In: *Machine Learning* 45 (říj. 2001), s. 5–32. DOI: 10.1023/A:1010950718922.
- [3] Erik Dahlström et al. *Somfy Open API*. 2011. URL: <https://www.w3.org/TR/SVG11/> (cit. 21. 04. 2022).
- [4] Štěpánka Lubinová. *Stínění oken: žaluzie, rolety, markýzy a slunolamy*. Grada Publishing, 2013. ISBN: 978-80-247-4579-4.
- [5] Adafruit Industries. *Adafruit TSL2591 High Dynamic Range Digital Light Sensor*. 2014.
- [6] AI-Thinker team. *ESP-12E WiFi Module*. Version 1.0. 2015.
- [7] Somfy. *Somfy Open API Developer Portal / APIs & Docs*. 2018. URL: <https://developer.somfy.com/apis-docs> (cit. 23. 03. 2022).
- [8] Maxim Integrated Products, Inc. *DS18B20 – Programmable Resolution 1-Wire Digital Thermometer*. 19-7487; Rev 6; 7/19. 2019.
- [9] Thibaut Etienne. *Somfy Open API*. 2021. URL: <https://github.com/tetienne/somfy-open-api#readme> (cit. 23. 03. 2022).
- [10] MongoDB, Inc. *MongoDB Documentation*. 2021. URL: <https://www.mongodb.com/docs/> (cit. 20. 04. 2022).
- [11] Apple Inc. *Apple Home*. 2022. URL: <https://www.apple.com/ios/home/> (cit. 21. 04. 2022).
- [12] Home Assistant. *Home Assistant*. 2022. URL: <https://www.home-assistant.io/> (cit. 21. 04. 2022).
- [13] Keras. *Keras API reference*. 2022. URL: <https://keras.io/api/> (cit. 21. 04. 2022).
- [14] OpenHAB Foundation. *OpenHAB*. 2022. URL: <https://www.openhab.org/> (cit. 21. 04. 2022).
- [15] Vercel. *Next.js*. Vercel. 2022. URL: <https://nextjs.org/> (cit. 21. 04. 2022).