



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**VYUŽITÍ SENZORICKÝCH DAT PRO ODVOZENÍ STAVU
PROSTŘEDÍ**

USING SENSOR DATA TO DERIVE ENVIRONMENT STATE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN SAKIN

VEDOUcí PRÁCE

SUPERVISOR

Ing. JAN VIKTORIN

BRNO 2018

Zadání diplomové práce

Řešitel: **Sakin Martin, Bc.**

Obor: **Inteligentní systémy**

Téma: **Využití senzorických dat pro odvození stavu prostředí
Using Sensor Data to Derive Environment State**

Kategorie: **Vestavěné systémy**

Pokyny:

1. Seznamte se se systémem inteligentní domácnosti BeeeOn.
2. Popište dostupné senzorické prvky a měřené veličiny, které je možné získat pomocí systému BeeeOn v různých lokalitách.
3. Analyzujte chování jednotlivých veličin z naměřených senzorických dat.
4. Natrénujte klasifikátor, který na základě senzorických dat automaticky detekuje výskyt různých událostí v prostředí, ve kterém probíhá měření.
5. Implementujte navržený systém.
6. Diskutujte dosažené výsledky a možnosti pokračování projektu.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Viktorin Jan, Ing., UPSY FIT VUT**

Datum zadání: **1. listopadu 2017**

Datum odevzdání: **23. května 2018**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
602 00 Brno, Božetěchova 2



prof. Ing. Lukáš Sekanina, Ph.D.
vedoucí ústavu

Abstrakt

Tato diplomová práce se zabývá analýzou, popisem a možným využitím sensorových dat ze systému inteligentní domácnosti. Dále je popsán inteligentní systém BeeeOn, který slouží jako zdroj sensorových dat, a možnosti rozšíření tohoto systému o automatizační úlohy. Následuje analýza všech všech měřených fyzikálních veličin, jejich vlastnosti a vliv na člověka. Zjištěné výsledky z naměřených dat byly využity k vytvoření klasifikátoru, založeném na hlubokých neuronových sítích, k detekování aktuálně probíhajících událostí v domácnosti. Detekované události mohou být využity pro navazující automatizační systém a tím přispět ke zlepšení životních podmínek. V závěru práce jsou diskutovány dosažené výsledky a možnosti pokračování na projektu.

Abstract

This diploma thesis deals with the analysis, description and usage of sensor data from an intelligent home system. This term also describes the intelligent system BeeeOn, which provides a sensor data and the possibility of extending this system to automation tasks. This is followed by the analysis of all the measured physical quantities, their properties and their influence on humans. The results from the measured data were used to create a classifier based on deep neural networks to detect current events at home. Detected events can be used for the following automation system to help improve living conditions. At the end of this thesis are discussed the results and options to continue with this project.

Klíčová slova

inteligentní domácnost, IoT, domácí prostředí, automatizace, využití dat, klasifikace, zdravé prostředí

Keywords

intelligent home, IoT, home environment, automation, use of data, classification, healthy environment

Citace

SAKIN, Martin. *Využití sensorických dat pro odvození stavu prostředí*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Viktorin

Využití senzorických dat pro odvození stavu prostředí

Prohlášení

Prohlašuji, že jsem tuto semestrální práci vypracoval samostatně pod vedením pana Ing. Jana Viktorina. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Martin Sakin
22. května 2018

Poděkování

Děkuji svému vedoucímu práce za odborné vedení a všem, kteří se podíleli na vývoji systému BeeeOn. V neposlední řadě také děkuji své rodině za podporu a umožnění dosáhnout vysokoškolského vzdělání.

Obsah

1	Úvod	3
2	Inteligentní domácnost a její automatizace	4
2.1	Obecná architektura IoT	4
2.2	Využívaný inteligentní systém	5
2.3	BeeOn brána	6
2.4	BeeOn server	8
3	Vlastnosti prostředí	10
3.1	Měřitelné fyzikální veličiny	10
3.2	Případy užití	13
3.2.1	Udržování teploty	13
3.2.2	Kritické hodnoty a změny	14
3.2.3	Kvalita vzduchu	14
3.2.4	Zabezpečení domácnosti	14
3.2.5	Naučení akcí	15
3.2.6	Detekce režimu dne	15
3.2.7	Vybíjení baterie	15
4	Analýza získaných dat	16
4.1	Manuální analýza	17
4.2	Statistické údaje	21
4.3	Předzpracování dat	22
4.3.1	Vyhazení signálu	22
4.3.2	Velikost změny	22
5	Návrh automatizovaného vyhodnocování	24
5.1	Návrh architektury	25
5.2	Detekované události	25
5.3	Datové sady	27
5.4	Umělá neuronová síť	28
5.5	Úprava vstupních dat	30
5.6	Vstupní vrstva neuronové sítě	31
5.7	Detekovaná událost a kontrola kritických hranic	32
6	Implementace navrženého řešení	34
6.1	Jádro klasifikátoru	34
6.2	Hlavní činnost aplikace	35

6.3	Popis tříd	36
6.4	Možná rozšíření	39
7	Výsledky	40
7.1	Přesnost datových sad	40
7.2	Přesnost detekce událostí	41
7.3	Zhodnocení	42
8	Závěr	45
	Literatura	46
A	Obsah CD	49

Kapitola 1

Úvod

Pojem inteligentní domácnost je ve světě již všeobecně známý a na trhu stále přibývají další a nové „chytré věci“ (angl. smart things). Tento termín neznamena nic jiného než rozšiřování běžných předmětů o elektronická zařízení, která přidávají možnost získávat a zpracovávat informace o svém fyzickém okolí a posílat je do sítě *Internet*. Díky tomu je pak možné získat informace o vzdálených zařízeních kdykoliv a kdekoliv. Pomocí chytrého telefonu můžeme kontrolovat svou domácnost i z práce nebo být včas upozorněn na nečekané události (narušení bezpečnosti, zaplavení bytu, atd.).

Internet of Things (IoT) tedy může být libovolné elektronické zařízení připojené do Internetu, které pracuje bez nutnosti lidského zásahu a generuje velké množství dat. V těchto datech se často skrývají informace, které mohou zvýšit efektivitu, ušetřit peníze či zlepšit životní komfort. Faktem ale je, že tímto způsobem bývá využíváno jen minimální množství dat a větší množství dat slouží jen k zobrazení aktuálního stavu bez smysluplného zpracování, vyhodnocení a použití. V dnešní době je již k dispozici velké množství různých dat, ale stále nám chybí jejich porozumění. Navíc čím více senzorů je k dispozici, tím více informací z nich můžeme potencionálně odvodit.

Jedním z problémů při získávání dat je ten, že jednotlivá zařízení od různých výrobců nejsou interoperabilní, tedy nedokáží mezi sebou komunikovat. Nejsou zatím dostupné ani všeobecně uznávané standardy, které by toto propojení umožňovaly nebo alespoň usnadňovaly.

Jednotlivá zařízení bývají navíc stále poměrně drahá, jelikož poptávka po těchto zařízeních stále není natolik masivní, aby se vyráběla sériově ve velkém množství. O sjednocení různých zařízení a technologií se snaží systém BeeeOn, který byl využit v této práci.

Cílem této práce je zjistit, které veličiny má význam měřit a jakým způsobem takto získaná dat využít pro automatizaci, zvýšení životního komfortu a úspore energií. V následujících kapitolách je představen inteligentní systém, který slouží jako zdroj sensorických dat, dále soupis měřitelných veličin se zaměřením se na jejich využití a problémy, a v závěru analýza získaných dat.

Kapitola 2

Inteligentní domácnost a její automatizace

V této kapitole je popsáno obecné schéma IoT¹ systému pro domácnost a kancelář. Dále je přehledově popsán systém *BeeOn*, který slouží jako zdroj sensorických dat. Následuje detailní popis nejdůležitějších částí tohoto systému: brány, která přijímá data ze sensorické sítě a posílá do Internetu, a serveru, který zpracovává data a zpřístupňuje uživatelským aplikacím. Systém *BeeOn* bude později rozšiřován o automatizační úlohy vycházející z této práce.

2.1 Obecná architektura IoT

IoT pro domácnost se neustále vyvíjí a není jednoduché najít univerzální řešení pro všechny typy bytů či domů. Nejjednodušší systémy pouze sbírají data a zobrazují je uživateli nebo umožňují vzdálené ovládání akčních prvků. Složitější systémy mohou nabízet některé prvky automatizace, jakým je např. vyvolání příslušné akce při překročení některých definovaných hodnot. Inteligentní systémy se dokáží samy naučit zvyklostem uživatele, přizpůsobovat se uživatelským příkazům a predikovat potenciální požadavky. Systém tak může uživateli nabídnout komfort automatické kontroly, zlepšení životních podmínek a snížení provozních nákladů.

Zjednodušeně lze IoT rozdělit na tři základní části:

- Koncová zařízení (věci) – zařízení, která jsou připojena do sítě kabelem nebo bezdrátově za účelem sbírat data nebo aktivně ovlivňovat své okolí.
- Síť – komunikační prostředek propojující všechny komponenty systému dohromady (komunikační sítě, brány, směrovače).
- Cloud – vzdálené servery datových center, které bezpečně sbírají data, analyzují a zpřístupňují informace.

Fyzické veličiny jsou snímány senzory, což mohou být malá „hloupá“ zařízení. Senzor nasnímanou hodnotu pošle na centrální místo v domácnosti nebo přímo do Internetu. Místo, kam se data zasílají, záleží na konkrétním systému a požadavcích uživatele. Pro podniky

¹IoT – z anglického Internet of Things, což lze přeložit jako Internet věcí. Tento termín se ale nepřekládá a v této práci bude používána anglická zkratka.

může být výhodnější si centrální místo s úložištěm zřídit lokálně a tím mít celý systém kompletně pod kontrolou. Pro menší podniky a domácnosti je jednodušší si pořídit nejnútnejší zařízení a uložení dat a analýzu přenechat nějaké službě s cloudovým řešením. Dnešní IoT služby mnohdy nabízí bezplatné uložení dat i se zálohou na vzdálených serverech s dostupností odkudkoliv z Internetu.

2.2 Využívaný inteligentní systém

Na Fakultě informačních technologií VUT v Brně se inteligentní domácností zabývá Výzkumná skupina akcelerovaných síťových technologií, zkráceně ANT (Accelerated Network Technologies). Vyvíjený IoT systém byl pojmenován jako *BeeOn* a je vyvíjen studenty a doktorandy zabývající se touto tematikou. Systém je koncipován jako stavebnice tak, aby se jednotlivé části dali jednodušeji vyvíjet a testovat. Je vyvíjen jako open-source a open-hardware, což znamená, že veškeré zdrojové kódy jsou veřejně dostupné na GitHub². Jednotlivé logické vrstvy systému, zobrazeny na obrázku 2.1 od nejvyšší po nejnižší, jsou následující:

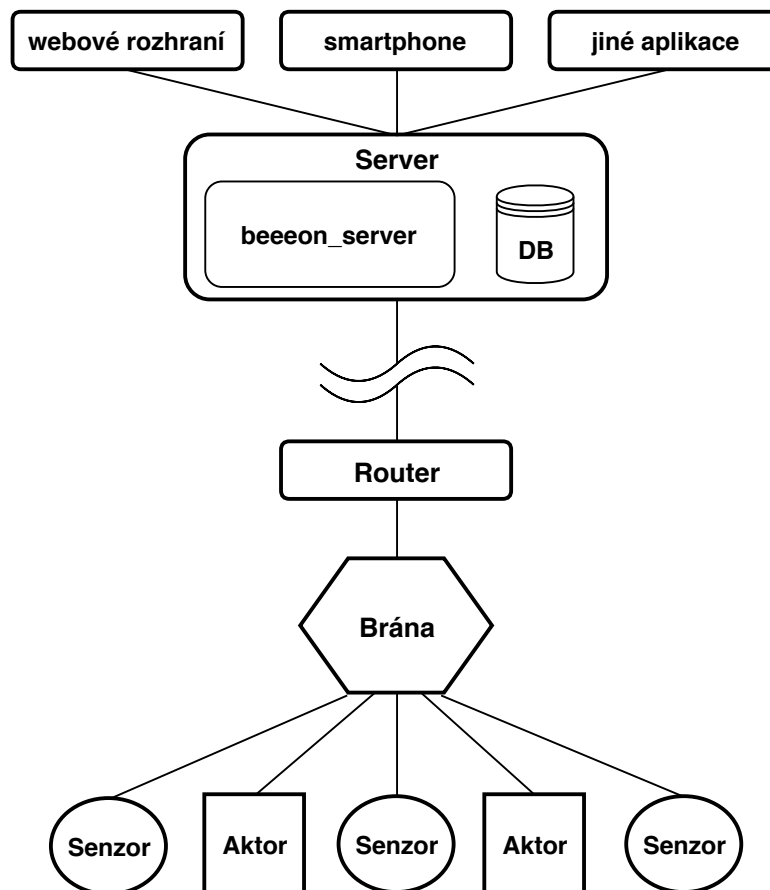
- kontrolní uživatelská **aplikace** – mobilní či webová aplikace umožňující zobrazení informací ze senzorů, nastavení aktorů nebo definování automatizovaného chování,
- **server** – prostředník mezi uživatelem a domácností, uchovává, zpracovává a zpřístupňuje data uživateli, *beeeon server* komunikuje s bránami a aplikacemi požadující data,
- **brána** – centrální místo v domácnosti pro sběr dat z různých bezdrátových prvků či zařízení a odesílání na server,
- **senzory** – snímají nějakou (nejen fyzikální) veličinu a bezdrátově odesílají hodnoty bráně nebo jsou na hodnoty dotazovány,
- **aktory** – zařízení, která požadovaným způsobem ovlivňují své prostředí (např. spínání zásuvky) a pokud i poskytuje informaci o svém aktuálním stavu, je také částečně senzorem.

Aktuálně podporované senzory a aktory jsou:

- BeeeOn senzor snímající teplotu a vlhkost komunikující bezdrátově pomocí vlastního experimentálního protokolu pojmenovaného *FIT Protocol*,
- Jablotron zařízení z projektu Turrís Gadgets³ detekující teplotu, pohyb, kouř, otevírání dveří a ovládání zásuvky,
- Belkin Wemo zásuvky a žárovky,
- Thermona VPT regulátor,
- Bluetooth pro detekci přítomnosti osob skrze mobilní telefon nebo bluetooth klíčenku,
- ZigBee, Z-Wave, IQRf a další bezdrátové technologie pro různá zařízení.

²BeeeOn na GitHub: <https://github.com/BeeeOn/>

³Jablotron Turrís Gadgets: <https://www.turrís.cz/gadgets/>



Obrázek 2.1: Schéma celého BeeeOn systému.

Nejsilnější a cílenou stránkou tohoto systému je modularita prvků, tedy možnost podporovat zařízení od mnoha výrobců co nejjednodušším a univerzálním způsobem. Další výhodou je zaměření na bezdrátové prvky, což přináší tu výhodu, že systém může být instalován bez nutnosti provádět stavební úpravy či instalovat novou kabeláž a jiné komplikované úpravy.

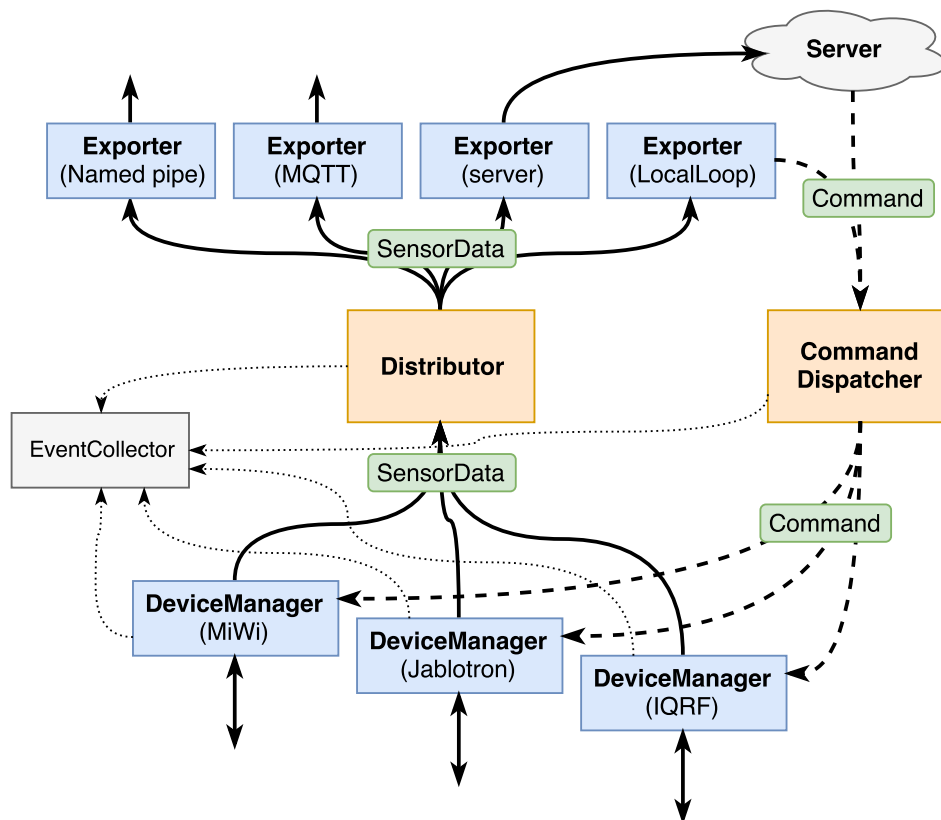
2.3 BeeeOn brána

Centrálním místem v domácnosti je brána, která přijímá data ze senzorové sítě a posílá je do Internetu. Brána se skládá z malého počítače A10-OLinuXino-LIME⁴ o velikosti platební karty, na kterém běží distribuce operačního systému Linux vytvořená prostřednictvím OpenEmbedded, a rozšiřující desku s anténou, RTC⁵ modulem, reset tlačítkem a informační diodou. Počítač obsahuje procesor Allwinner A10 ARM Cortex-A8 1 GHz, 512 MB DDR3 paměť RAM, 100 MBit Ethernet, 2x USB 2.0 a je napájen 5 V zdrojem. Operační systém i další aplikace jsou uloženy na paměťové kartě.

⁴A10-OLinuXino-LIME documentation webpage: <https://www.olimex.com/Products/OLinuXino/A10/A10-OLinuXino-LIME/open-source-hardware>

⁵RTC – Real-time clock module

Softwarovým centrálním prvkem je aplikace **Gateway**, která má primárně za úkol zpracovávat data ze senzorů, posílat je na server a zpracovávat příkazy ze serveru (z uživatelské aplikace). Interní rozhraní této aplikace je na obrázku 2.2.



Obrázek 2.2: Interní rozhraní aplikace Gateway.

V nejnižší vrstvě *Gateway* je pro každou technologii senzorů či aktorů vytvořen modul (*DeviceManager*) s vlastním vláknem. Každý *DeviceManager* naslouchá nebo se dotazuje na stav zařízení a získané informace zpracovává způsobem specifickým pro konkrétní zařízení. Výstupem každého modulu *DeviceManager* je unifikovaná zpráva *SensorData*. Tato zpráva je předána třídě *Distributor* pro rozeslání na všechny definované *Exportéry* (server, MQTT, mamed pipe, atd.). Zpracování příkazů (ze serveru, MQTT, atd.) zajišťuje *CommandDispatcher*, který přijímá a kontroluje příkazy, a přeposílá je do cílového modulu *DeviceManager*. Tento manager provede akci dle příkazu specifickým způsobem pro každé zařízení.

Dále jsou v aplikaci *Gateway* doplňující komponenty, které nejsou pro tuto práci příliš podstatné. Například cache pro uložení sensorických dat, které se exportérům nedařilo odeslat, *EventCollector* pro shromažďování statistických údajů a chyb nebo *CredentialsStorage* pro bezpečné uložení potřebných citlivých informací pro zařízení (uživatelské jméno, heslo, pin, atd.).

Jedna z možností, jak jednoduše získat potřebná data ze senzorů pro analýzu, je MQTT exportér. Tento exportér zajišťuje distribuci dat pomocí protokolu MQTT. Klienti se připojí a posílají zprávy na centrální MQTT broker. Ostatní klienti, připojení na stejný broker, mohou tyto zprávy číst a nebo zasílat svoje zprávy. Klasifikátor, připojený na stejný MQTT broker jako brány, tak bude moct číst data z senzorů a zpracovávat je. Tím pádem je také

možné rozdělit brány na skupiny s odlišnými MQTT brokery, např. pro domácnosti a kanceláře, Data z každé skupiny bran tak mohou být zpracovávány specifickým klasifikátorem na odlišných serverech.

2.4 BeeeOn server

Serverová aplikace je rozdělena do tří základních vrstev pro oddělení zodpovědnosti:

- **aplikační rozhraní** – realizuje komunikaci s uživateli skrze mobilní či webovou aplikaci, s bránami pro příjem dat a posílání příkazů, atd.,
- **servisní vrstva** – hlavní logika, jaké zprávy kam poslat nebo jak je zpracovat,
- **datová vrstva** – přístup k datům v databázi.

Jednotlivé vrstvy jsou provázány pomocí tříd z datového modelu a jednotlivé vrstvy neznají implementační detaily vrstev ostatních. Tento přístup umožňuje úpravy některé z vrstev bez zásadního vlivu na ostatní vrstvy a zjednodušuje testování.

Aplikační vrstva poskytuje rozhraní XML-UI a REST-UI, volba záleží na konkrétní aplikaci. XML-UI je starší procedurálně orientovaný XML protokol pro komunikaci mezi mobilní aplikací a serverem. REST-UI je novější datově orientovaný protokol pro komunikaci mezi serverem a libovolnou aplikací posílající JSON zprávy.

Servisní vrstva představuje hlavní logiku serveru, tedy jak zpracovávat příkazy a data. Každá servisní operace má následující kroky:

1. Vytvoření transakce.
2. Ověření přístupových práv.
3. Načtení požadovaných dat z datové vrstvy.
4. Provedení příslušné operace nad načtenými daty.
5. Uložení dat zpět do datové vrstvy.
6. Potvrzení transakce.

Každá operace je obvykle jedna databázová transakce a měla by proběhnout atomicky. Tedy pokud selže dílčí krok, selže celá operace a nezanechá žádné nežádoucí následky. Výsledkem je pak provedení operace a odpověď s výsledkem nebo oznámení o neúspěchu.

Datová vrstva zajišťuje pouze práci s databází. Data jsou ukládána do PostgreSQL⁶ databáze. SQL dotazy jsou formulovány mimo C++ zdrojový kód a načítají se při inicializaci serveru. Data mohou být následně přesunuta do databáze Cassandra⁷, která nabízí jiný způsob uložení vhodný pro velké množství dat.

Libovolná část serveru může generovat notifikace pro jinou část systému BeeeOn, a to jak mezi vrstvami aplikace, tak i mimo server do uživatelské aplikace. Tím může být uživatel dobře informovaný o vzniku různých událostí či problémů.

Všechny vrstvy serveru mezi sebou sdílejí datový model systému s následujícími entitami:

⁶PostgreSQL – open-source, objektově orientovaná databáze s důrazem na dodržování a rozšiřování norm. <https://www.postgresql.org/>

⁷Apache Cassandra – open-source distribuovaná NoSQL databáze, <http://cassandra.apache.org/>

- Identity – identita uživatele, jedná se v zásadě o e-mail, přes který se uživatel přihlašuje.
- User – uživatelský profil, informace, které sám poskytl (jméno, příjmení, nastavení profilu, atd.).
- VerifiedIdentity – identita ověřená některým podporovaným poskytovatelem ověření identity (Google, Facebook).
- Gateway – brána, ke které se připojují zařízení obsahující sensorové a aktivní prvky.
- LegacyGateway – reprezentace brány pro XML-UI API.
- RoleInGateway – role přiřazující dvojici (Identity, Gateway) přístupová práva.
- Device – zařízení, které obsahuje jeden nebo více sensorů nebo jiných aktivních prvků.

Server všechny validní přijaté datové zprávy přeposílá na ZeroMQ⁸. ZeroMQ je asynchronní knihovna zpráv pro použití v distribuovaných nebo souběžných aplikacích. Systém ZeroMQ poskytuje frontu zpráv a může běžet i bez zprostředkovatele zpráv. Pokud tedy není možnost nastavit na každé bráně MQTT exportér, využití ZeroMQ se jeví jako vhodná alternativa.

⁸ZeroMQ - <http://zeromq.org/>

Kapitola 3

Vlastnosti prostředí

Tato kapitola se zabývá různými fyzikálními veličinami a jejich chováním v domácnosti. Každá veličina je detailně popsána a k ní i scénáře s problémy, které můžeme detekovat a akčním způsobem na ně reagovat či je eliminovat.

3.1 Měřitelné fyzikální veličiny

Následující seznam obsahuje informace k fyzikálním veličinám, které jsou pro automatizaci zajímavé. Zajímají nás především kritické hodnoty, které mohou člověku škodit a doporučené či normované hodnoty pro ideální, zdravotně nezávadné, životní prostředí. Zde jsou zmíněny veličiny, které můžeme měřit a analyzovat, ale ne všechny musí být ve výsledku využity. Mezi některými veličinami jsou vazby a zásahem do prostředí můžeme ovlivnit více než jednu další veličinu.

Relativní vlhkost

Vlhkost v domě vzniká vařením, sprchováním, praním a sušením prádla, vydechováním obyvatel, ale také třeba i z rostlin. To všechno, v případě čtyřčlené rodiny, může znamenat až 10 litrů vody denně. Navíc se také v nepatrném množství odpařují škodlivé látky z nábytku, koberců, barev a jiného vybavení domu, které se vážou na tuto vlhkost [15]. Podle studií z roku 1985, kde se zkoumalo vystavení člověka různým úrovním vlhkosti v obytných budovách, se ideální vlhkost pohybuje v rozmezí 40 – 60 % [20].

Vzduch dokáže vlhkost částečně absorbovat v závislosti na teplotě: teplý vzduch absorbuje více než studený. Když se ochladí vlhký teplý vzduch na studené ploše, dochází ke kondenzaci: plocha se orosí. Při vnitřní teplotě 20 °C a vlhkosti 60 % obsahuje 1 m² vzduchu v místnosti přibližně 10 g vody. Při poklesu teploty na 8 °C, klesne míra vlhkosti na polovinu. Přebytkových 50 % vlhkosti se vysráží jako kondenzát na chladnějších stěnách, nejčastěji v rozích a na oknech. Taková místa potom zvyšují možnost tvorby plísní [12].

Naopak nízká vlhkost v místnosti také není vhodná. Škodí zdraví tím, že vysušuje dýchací cesty (sliznici) a snižuje obranyschopnost organismu. Navíc v suchém prostředí má člověk pocit, že je chladněji než skutečně je, a to nás nutí topit více (pocitový rozdíl může být až 3 °C) [28], což vede ke zbytečně zvýšeným nákladům za vytápění.

Pro nové obytné domy dnes platí vyhláška č. 406/2000 Sb. o hospodaření energií, podle které musí být budovy téměř vzduchotěsné. To tedy uzavírá poslední cesty pro větrání, jakými byli například spáry v oknech a dveřích (viz [15, 22]).

V takovém domě se pak už při stavbě montují např. ventilační jednotky, šachty, klimatický systém, atd. Ve starých domech ale takové možnosti nejsou a ani kompletní rekonstrukce s ventilačním systémem nemusí být dokonalým řešením. Dalším důvodem může být i to, že majitel staršího domu nemá zájem investovat do vysokorozpočtových řešení.

Teplota

Teplota má velký vliv na psychiku a zdraví člověka. Navíc příjemná pokojová teplota je dosti individuální a záleží na mnoha faktorech, např.:

- ženy obecně požadují vyšší teploty než muži,
- fyzicky aktivním lidem stačí nižší teploty než méně aktivním,
- děti a důchodci vyžadují vyšší teploty,
- dočasně nemocní lidé by měli být v teplejší místnosti, atd.

Obecně lze ale říct, že ideální pokojová teplota v obytných místnostech se pohybuje mezi 20 a 21 °C. Tato hranice je dána i vyhláškou č. 194/2007 Sb. o vytápění obytných budov [21]. Hodnota 20 °C se týká také pracovny, relaxační místnosti, kanceláře, učebny, a dalších místností, kde člověk běžně tráví svůj den. Vyšší teplota 24 °C by měla být v místnostech, kde by se člověk mohl svlékat, jako například: koupelna, převlékárna, ošetřovna, atd. Naopak v kuchyni a ložnici může být chladněji. V ložnici se doporučuje teplota v rozmezí 17–19 °C, která je ideální pro kvalitnější a hlubší spánek. V kuchyni se předpokládá přísun tepla z elektronických spotřebičů, které jsou použity pro přípravu jídla a ze samotného jídla, které člověk zkonzumuje, čímž se zahřeje. Teplota v bytě má vliv i na náš výkon a myšlení. Nejlepší výsledky člověk může podávat při teplotě okolo 22 °C (viz [2, 21, 3]).

Pokud jsou v domácnosti malé děti, starší osoby nebo je některý ze členů domácnosti nemocen, optimální teplota v místnosti by se měla pohybovat kolem 23–24 °C [23]. Důchodci, kteří nemají mnoho fyzické aktivity, vyžadují teplotu dokonce vyšší.

Atmosférický tlak

Atmosferický tlak je veličina, kterou v běžném případě nemůžeme ovlivnit. Naopak ale může ovlivnit fyzický stav člověka, jelikož rychlé změny tlaku vzduchu může u oslabených jedinců vyvolávat zdravotní potíže. Například problémy s dýcháním, bolesti hlavy, nevolnost, a jiné [29]. Ale i u zdravého jedince může způsobovat nečekanou únavu a nevolnost. Je tedy vhodné tuto veličinu sledovat alespoň za účelem informování uživatele při velkých změnách hodnot.

Dále je možné z tlaku předvídat počasí. Když je tlak vzduchu nižší než 1000 hPa, očekává se deštivé počasí. Platí, že čím nižší tlak, tím je deštivé počasí stálejší. Pokud je tlak vzduchu vyšší než 1020 hPa, očekává se suché a většinou slunečné počasí. Ale i při vysokém tlaku může přijít průtrž mračen [19].

Oxid uhličitý

Se zateplováním rekonstruovaných budov se snižují tepelné ztráty, ale současně s tím vzrůstá i koncentrace CO₂. Oxid uhličitý vzniká při spalování a dýchání. V domácnosti jsou hlavním zdrojem především lidé, domácí mazlíčci a v malé míře i rostliny. Doporučená úroveň CO₂ ve vnitřních prostorech je 400–1000 ppm. Vyšší koncentrace (nad 1500 ppm) může způsobovat únavu, pálení očí a nepozornost. Při nebezpečné koncentraci (nad 5000 ppm) přichází

nevolnost, dýchací potíže a při extrémní úrovni (nad 40000 ppm) je možná i ztráta vědomí [27, 16]. Je tedy žádoucí na překročení kritických hodnot uživatele co nejdříve upozornit.

Hluk

Pod označením hluk chápeme nevyžádaný (hlasitý) zvuk. Nejčastějším zdrojem hluku v okolí obydlí bývá doprava (silniční, železniční, letecká). Jediným způsobem, jak tento hluk omezit, je zavřít okna. Ale nejen kvůli nutnosti větrat, není možné mít okna konstantně zavřená. Navíc některé zdroje hluku se nacházejí i v domácnosti, např. lednička, pračka, počítač, atd. Problém je, že člověk si na hluk časem zvykne, jenomže dlouhodobé vystavování většímu hluku ohrožuje naše duševní zdraví. Mozek se totiž snaží zpracovávat tyto vjemy a čím více jich je, tím více je mozek zahlcován. Tím se člověk rychleji unaví, bolí ho hlava a je méně efektivní [6].

Základní limity hluku uvnitř obytných místností jsou stanoveny na 40 dB přes den (6–22 h) a 30 dB v noci (22–6 h) [13]. Jednoduše lze ale říct, že čím méně hluku, tím lépe.

Avšak malé dočasné hluky některých zařízení, mohou být užitečné pro automatizaci domácnosti. Pokud bychom měli k dispozici dostatečně kvalitní mikrofon, bylo by teoreticky možné detekovat, co se v místnosti děje (zapnutí kávovaru, umývání nádobí, atd.) a na to navázat různé akce (regulace topení, alarm, atd.).

Vibrace

Vibrace lze definovat jako mechanické kmitání a chvění prostředí. I krátkodobá expozice může u člověka vyvolat nepříznivé následky. Závisí na intenzitě a délce působení vibrací. Vystavení vibracím může způsobovat nepříjemné subjektivní pocity, jako například únavu, snížení pozornosti, zhoršení vnímání, atd. Příkladem je například únava při cestování autobusem, což je kombinace vibrací a hluku. Vibrace v budovách obecně působí rušivě na duševní stav člověka [1]. Zdrojem mohou být basové reproduktory, které přenáší nízké frekvence přímo konstrukcí domu, pračka, myčka nebo železniční a městská hromadná doprava či frekventované silnice blízko domu [18].

Intenzita osvětlení

Potřebná intenzita osvětlení závisí na typu místnosti a činnosti, jakou zde člověk bude provádět. Minimální hodnota pro čtení doma na pohovce je 50 lx. Za tmy (do 0,1 lx) má lidské oko možnost pouze černobílého vidění. Smíšené vidění se pohybuje v rozmezí 0,1–100 lx. Přibližně 90 % svých možností lidské oko dosáhne až při hodnotách nad 1000 lx. Pro společenské a obytné prostory je doporučeno rozmezí 20–200 lx, pro pracovní a dlouhodobou zrakovou činnost je doporučeno rozmezí 200–2000 lx [14].

Jako nejlepší zdroj světla je přirozené denní světlo. V ostatních případech se doporučuje umělé osvětlení rovnoměrně rozprostřené po celé místnosti [11]. To z toho důvodu, aby oko nemuselo příliš rozšiřovat a zužovat zornici.

UV záření

Tato forma elektromagnetického záření má kratší vlnové délky (cca 10–400 nm) než viditelné světlo (cca 400–800 nm). Běžné křemenné sklo propouští jen část UV záření a příměsová skla mají propustnost na hranici viditelného světla (okolo 400 nm) [1]. Narozdíl od senzoru intenzity osvětlení nám senzor ultrafialového záření poskytuje informaci,

zda do místnosti svítí sluneční paprsky. Této informace lze využít např. ke snížení výkonu topného systému.

Mezinárodním standardním měřením intenzity ultrafialového záření je *UV index*, který popisuje, kolik je produkováno slunečních paprsků na určitém místě a čase. Primárně se používá v předpovědi počasí. Účelem UV indexu je pomáhat lidem se chránit před silným UV zářením ze zdravotních důvodů. Vysoké úrovně UV záření totiž mohou způsobit popáleniny pokožky, poškození DNA, rakovinu kůže, poškození zraku, atd. Měřítka UV indexu se pohybuje v rozmezí 0–11+, kde pro nízké hodnoty stačí použít sluneční brýle a pokrývku hlavy, a v případě vysoké hodnoty se vůbec nedoporučuje vystavovat slunečnímu záření [9, 4].

3.2 Případy užití

Abychom mohli docílit vhodného automatizovaného chování, musíme určit, co chceme z naměřených hodnot fyzikálních veličin získat nebo jak je využít. K tomu slouží následující seznam případů užití zajímavých pro domácnost a uživatele. Podle kritérií úlohy můžeme nechat natrénovat vhodnou formu umělé inteligence (např. neuronovou síť), která může poskytnout dříve neznámé informace a spojitosti, a ve výsledku zvýšit uživatelský komfort. Je ale také potřeba počítat s možností, že ne všechny hodnoty jsou správné a vždy k dispozici. Například pokud dojde k vybití baterie v senzoru nebo se přeruší spojení.

Jako možné vylepšení lze úlohy částečně kombinovat. Informace získané z jedné úlohy mohou být užitečné (doplňující) pro úlohu jinou a tím odvodit další informace.

V následujících scénářích je obvykle zmíněná jako výstupní akce notifikace uživatele. Notifikací je primárně myšleno upozornění přijaté mobilním zařízením, stejně jako upozornění na příchozí zprávu, e-mail, atd. Další možností je vyskakovací okno ve webovém prohlížeči, pokud bude uživatel v dané chvíli přihlášen. Alternativou může být poslání pouze informačního e-mailu. Druhů notifikací je mnoho a mohou být řešeny různými kreativními způsoby, např. upozornění na vydýchaný vzduch v místnosti může být signalizován změnou barvy vícebarevné žárovky, atd.

3.2.1 Udržování teploty

Asi nejzákladnější automatizační úlohou pro domácnost je regulace teploty. Teplota by se měla pohybovat v mezích příjemných pro uživatele. Udržování ideální teploty je ale nutné především tehdy, když je uživatel přítomen, nebo když je očekáván jeho příchod. Pokud máme k dispozici informaci o přítomnosti uživatelů v domácnosti, a v domácnosti se nikdo nenachází, můžeme snížit teplotu a tím ušetřit na energiích a financích.

V nejjednodušším případě je zapotřebí alespoň jeden teplotní senzor. V ideálním případě může být teplotních senzorů více. Dále je velmi užitečný pohybový senzor, senzor detekce přítomnosti uživatele a informace o venkovní teplotě. Na základě dat z těchto senzorů může úloha poskytovat uživateli doporučení pro úpravu nastavení topného systému formou notifikací nebo týdenních přehledových e-mailů. V případě, že je k dispozici i řízení topného systému, klimatizace, nebo jiný systém přímo ovlivňující teplotu pro danou místnost, může se provádět i automatické řízení nebo změna konfigurace.

Na začátku, kdy úloha ještě nemá žádné informace o prostředí, se výchozí udržovací teplota stanoví dle vyhlášky nebo dle požadavků uživatele. Naměřené agregované hodnoty, po půl nebo celé hodině, se budou předkládat umělé inteligenci, která se naučí různé re-

žimy dne typické pro uživatele. Následně bude umělá inteligence detekovat tyto režimy a předkládat případné doporučení na změnu nastavení topného systému.

Podobné schéma lze aplikovat i pro vlhkost s tím rozdílem, že musíme rozlišovat krátkodobou a dlouhodobou vlhkost. Na krátkodobě zvýšenou vlhkost je možné reagovat spuštěním odvětrávání, pokud je k dispozici, nebo nereagovat vůbec. V případě dlouhodobé vlhkosti, která může vést ke vzniku plísní, je potřeba uživatele na tuto skutečnost upozornit alespoň notifikací.

3.2.2 Kritické hodnoty a změny

Pro tuto úlohu samotnou není úplně nutný inteligentní systém, jelikož postačí pouze nastavit tzv. trigger (spouštěče). Pro každou veličinu známe krajní hodnoty, které mohou být pro člověka nebo domácnost nevhodné. Při překročení kritické hodnoty je uživatel notifikován nebo se přímo vykoná uživatelem přednastavená akce. Inteligentní chování lze v tomto případě využít pro detekci výrazných změn hodnot. Kritické situace mohou být:

- překročení kritické úrovně CO₂ (spuštění alarmu),
- prudký pokles teploty v místnosti (může značit rozbití okna nebo právě otevřené okno),
- dlouhodobá neklesající vysoká vlhkost (prasklé potrubí, vytopení bytu),
- výrazně zvýšená teplota na místě, kde je obvykle chladněji (predikce požáru),
- prudký pokles atmosférického tlaku, atd.

Ovšem musíme brát v potaz, že ne všechny rychlé změny a překročení hraničních hodnot musí značit nějaký problém. Např. otevření okna v zimě způsobí velký pokles teploty, vlhkosti a CO₂. To je ale pravděpodobně důvod, proč uživatel okno otevřel a není nutné ho na tento fakt upozorňovat. K tomu může pomoci umělá inteligence, která rozpozná známou událost a varovat uživatele nebude.

Budou se rozlišovat dvě úrovně notifikací. Nízká úroveň, upozorňující pouze na neideální stav prostředí, kterou uživatel může ignorovat nebo vypnout. Vysoká úroveň, která upozorňuje na závažné problémy a může spouštět definované záchranné akce domácnosti, jako např. spuštění odvětrávání, klimatizace, otevření okna, atd.

3.2.3 Kvalita vzduchu

Kvalita ovzduší má na člověka velký dopad. Prostředí má přímý vliv na pohodu, výkon a zdraví člověka. Např. vydýchaný vzduch nebo vysoká teplota zvyšují únavu, nízká teplota může vést k nachlazení, nízká vlhkost k dýchacím problémům, atd.

Pokud bychom měli mnoho senzorů měřících různé vlastnosti vzduchu, bylo by možné vypočítat *skóre* kvality vzduchu a uživatel by měl informaci, v jak zdravém prostředí žije, popřípadně doporučit úkony ke zlepšení. Potřebné senzory by měly měřit: teplotu, vlhkost, CO (oxid uhelnatý), CO₂ (oxid uhličitý), kouř, prach, atd.

3.2.4 Zabezpečení domácnosti

I v případě, že inteligentní systém nemá k dispozici zabezpečovací senzory (detektor pohybu, magnetický senzor na dveřích, atd.), je do jisté míry možné odhadnout neočekávané chování.

Mějme k dispozici pouze teplotní a vlhkostní senzor a situaci, kdy se do domu v noci vloupá zloděj. Během noci, kdy uživatelé domácnosti již spí, do místností nesvítí slunce a okna jsou zavřená, se teplota ani vlhkost téměř nemění (záleží na přesnosti senzorů). Když do domácnosti vstoupí zloděj, tak otevřením vchodových dveří či okna a také pohybem po místnosti změní teplotu i vlhkost. Velikost změny těchto veličin závisí na ročním období, době otevření dveří, atd. Aby systém věděl, že se jedná o narušení objektu, musí uživatel v systému nastavit časový interval, ve kterém se mají detekované události požadovat za neočekávané a určit následnou akci (alarm, notifikace, atd).

Výhodou tohoto řešení je získání další funkce bez nákupu nových senzorů. Nevýhodou ale může být dlouhotrvající odezva. Sensory, které mají delší časový interval mezi posílanými daty (10 minut a více) mohou o změně veličin informovat pozdě, nebo změnu vůbec nezaznamenají.

3.2.5 Naučení akcí

Tento scénář je vhodný pro uživatele s pravidelným životním stylem. Umělá inteligence sleduje uživatelské ovládání aktorů a hledá v nich pravidelnosti. Tyto pravidelnosti se dají také využít pro výše zmíněné úlohy k zpřesnění predikce chování. Např. když si uživatel každé ráno před odchodem do práce pouští rádio, může se tato akce provádět automatizovaně a sloužit tak jako budík či upozornění. Samozřejmostí je rozlišování pracovních a nepracovních dnů.

Problém ale nastává při změně těchto úkonů, jelikož nelze hned jednoznačně poznat, zda se jedná o výjimku nebo naprostou změnu. Taková vlastnost systému musí ale mít silnou podporu v jednoduchém uživatelském nastavení, kde si uživatel může určit které akce povolí a které ne. Pokud by se totiž umělá inteligence rozhodla vykonávat opakovanou činnost sama místo uživatele, mohlo by to uživatele odradit od používání systému.

3.2.6 Detekce režimu dne

Nejzákladnější dělení dnů je na pracovní dny a víkendy. Tedy situace, kdy jsou uživatelé přes den mimo domov (v práci, ve škole, atd.) a nejvíce topí pouze odpoledne a večer, nebo dny, kdy je v domácnosti většinu dne. Další charakteristiky dnů mohou vzniknout, když je uživatel nemocný a tráví v domácnosti celý den, nebo když je slunečný den a sluneční svit vyhřívá okny místnost i bez zapnutého topení, atd. Pokud by se tedy umělá inteligence naučila několik režimů dnů, lze eliminovat nechtěné až „divoké“ chování celého inteligentního systému.

Tato znalost průběhu dne lze využít pro generování dat virtuálních senzorů pro účely testování nebo dočasné nahrazení nefunkčního senzoru. Pokud by byl systém nasazen např. na chatě, kde se uživatelé vyskytují jen některé dny, bylo by možné tuto informaci o přítomnosti uživatelů využít k zabezpečení objektu.

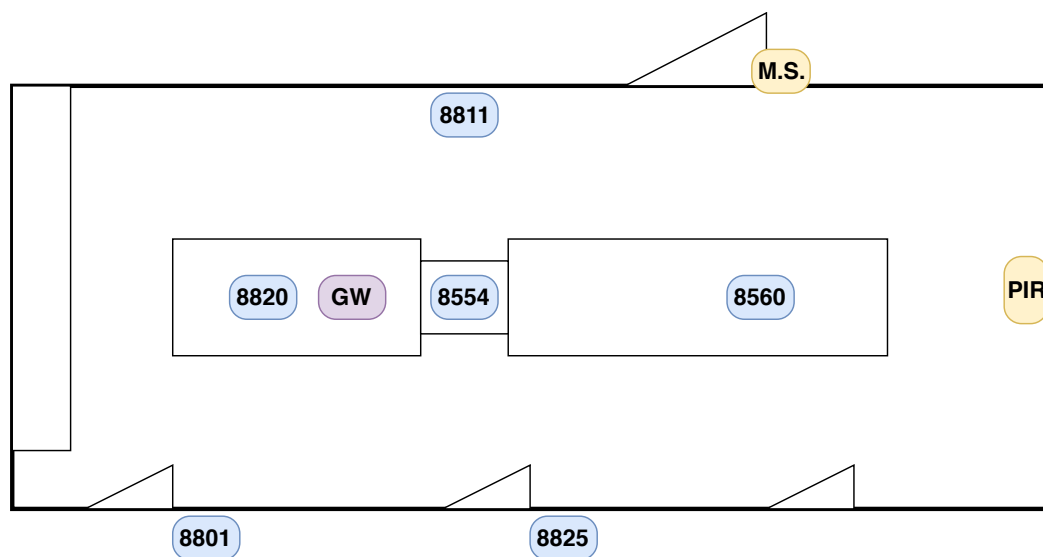
3.2.7 Vybíjení baterie

Tento bod nesouvisí přímo s měřenými fyzikálními veličinami, ale je důležitý pro funkčnost ostatních scénářů výše. Pokud bezdrátový senzor poskytuje stav baterie (nejčastěji v procentech), pak by mělo být možné zjistit z vybíjecí křivky o jakou baterii se jedná (NiMH, Li-ion, atd.) a tím odhadnout dobu života a včas uživatele upozornit na nízký stav baterie. S nízkým stavem by se senzor mohl přepnout do jiného probouzecího režimu nebo prodloužit interval mezi posíláním dat pro prodloužení života senzoru.

Kapitola 4

Analýza získaných dat

Získané sady dat byly reálně nasbírány ze tří různých lokalit. Všechny BeeeOn senzory měly nastaven *refresh time*¹ na pět minut. Při nastavení delšího časového intervalu bychom mohli zmeškat některé rychlé a krátkodobé změny prostředí, naopak při nastavení kratšího intervalu se rychleji vybíjí baterie. Senzory od firmy Jablotron nemají možnost nastavení časového intervalu a v případě, že nenastaly v prostředí žádné změny, posílají informaci o živosti senzoru každých devět minut. Snímané veličiny byly: teplota, vlhkost, pohyb, přítomnost, otevírání dveří.

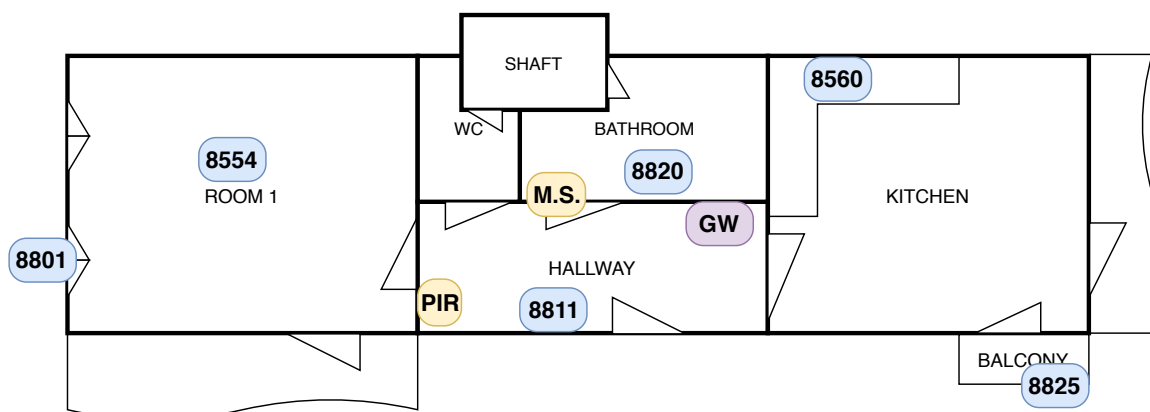


Obrázek 4.1: Rozmístění senzorů v laboratoři.

Prvním sběrným místem byla počítačová laboratoř, kde bylo hlavním cílem zjistit možnosti dostupných senzorů, otestovat jejich funkčnost a najít nejvhodnější umístění. Rozmístění senzorů je zobrazeno na obrázku 4.1. Všechny senzory byly umístěny pouze v této jedné místnosti o ploše cca 12 × 5 metrů. Ve středu místnosti se nachází rack se servery a stoly s pracovními počítači. Modré značky s čísly značí senzory, GW je brána sbírající data, MS je magnetický senzor umístěný na dveřích a PIR pohybový senzor.

Během dne se v laboratoři nepravidelně pohybovalo různé množství lidí (3 až 10 osob) a v noci nikdo. V ranních hodinách byl prováděn běžný úklid.

¹refresh time – časový interval mezi posílanými daty.



Obrázek 4.2: Plánek studentského bytu – druhé sběrné místo.

Druhým sběrným místem byl studentský byt ve starším domě se silnými zdmi, vysokými stropy a plastovými okny. Plánek bytu a rozmístění senzorů (modré ovály s čísly) jsou zobrazeny na obrázku 4.2. Byt nepravidelně obývalo šest lidí. Přes den v pracovní dny se v bytě vyskytovala maximálně jedna osoba a večer maximálně pět. O víkendech zde byli maximálně tři lidé. Sledovaly se veličiny: teplota, vlhkost, tlak, detekce pohybu, detekce otevřených dveří a spínání zásuvky. Senzory BeeeOn byly umístěny do kuchyně, koupelny, obývacího pokoje (ložnice), chodbě a na balkóně. Senzor pohybu sledoval pohyby na chodbě a magnetický senzor detekoval otevírání a zavírání dveří koupelny. V koupelně a WC se konstantně odvětrávalo ventilační šachtou.

Třetím sběrným místem byl rodinný dům o třech členech a domácí kočkou s relativně pravidelným režimem. Senzory BeeeOn byly rozmístěny do kuchyně, koupelny, tří obývacích pokojů a z venku na okně. V jednom z pokojů byl umístěn i senzor pohybu.

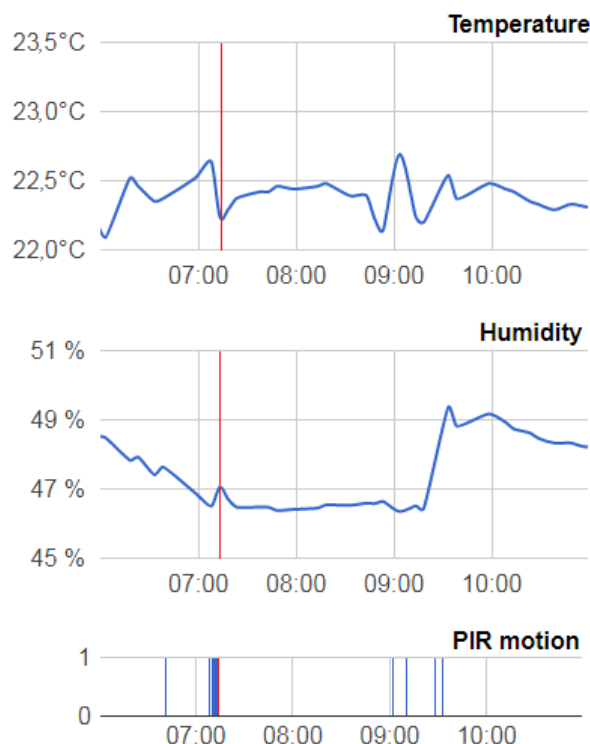
4.1 Manuální analýza

Prvním krokem byla vizualizace dat ve formě grafu pro manuální kontrolu, vyhledávání vzorů a detekci událostí. K tomuto účelu byl implementován skript v jazyce Python pro vyfiltrování dat vybraného senzoru a časového rozsahu, a konvertování do požadovaného formátu pro vývojářský online nástroj Google Chart². Vytvořené grafy pak byly vykreslovány ve webovém prohlížeči.

Odchod uživatele

Pozitivní byl nálezný ve spojitosti mezi teplotou, vlhkostí a pohybem na chodbě. V grafech na obrázku 4.3 jsou tyto změny veličin vyznačeny červenou čarou. Pohybový PIR senzor detekoval častější pohyb na chodbě po dobu tří minut. Následně pohyby ustaly, teplota poklesla o 0,6 °C a vlhkost klesla o 2,5 %. Tím lze vyvodit, že byly otevřeny vchodové dveře a uživatel odešel, jelikož pohyby na chodbě ustaly.

²Google Chart – <https://developers.google.com/chart/>



Obrázek 4.3: Detekce odchodu uživatele (červená čára) – teplota, vlhkost, senzor pohybu.

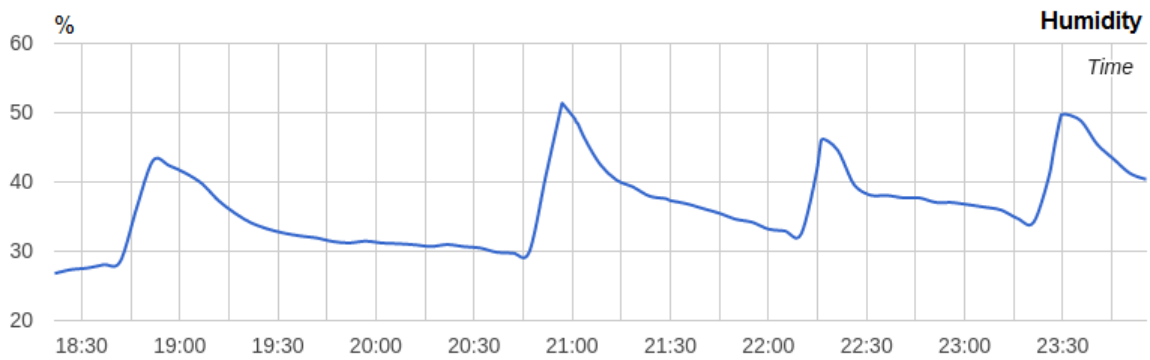
Sprchování

Další užitečná informace je detekce sprchování, která může indikovat například přípravu uživatele ke spánku, k odchodu z domácnosti, a nebo pouze eliminovat varování na vysokou vlhkost. I v případě částečně odvětrávané koupelny, vlhkost roste velmi rychle, cca 1.5 % za minutu a teplota cca 0.3 °C za minutu. Tím pádem by mělo být možné tuto událost detekovat již do 2–3 minut. V případě osmiminutového sprchování teplota vzroste cca o 1 až 3 °C v závislosti na teplotě vody. Charakteristické pro tuto detekci je rychlý růst vlhkosti, oproti dennímu průměru vzroste až o 12 %. Zajisté záleží také na velikosti koupelny a jak dlouho uživatel v koupelně stráví čas. Z dostupných měření se doba sprchování uživatelů se pohybuje mezi 6 až 20 minutami.

V případě měřeného bytu a domu je sprcha nejvýraznější událost, tedy snad i nejlépe detekovatelná. Na straně 19 v grafu vlhkosti na obrázku 4.4 jsou vidět *špičky*, v časech 18:46, 20:51, 22:15, 23:26, kdy se uživatelé sprchovali. Růst vlhkosti značí průběh sprchování a nejvyšší hodnota konec. Teoreticky by z těchto dat mohlo odvodit, kolik lidí se v domácnosti pohybuje.

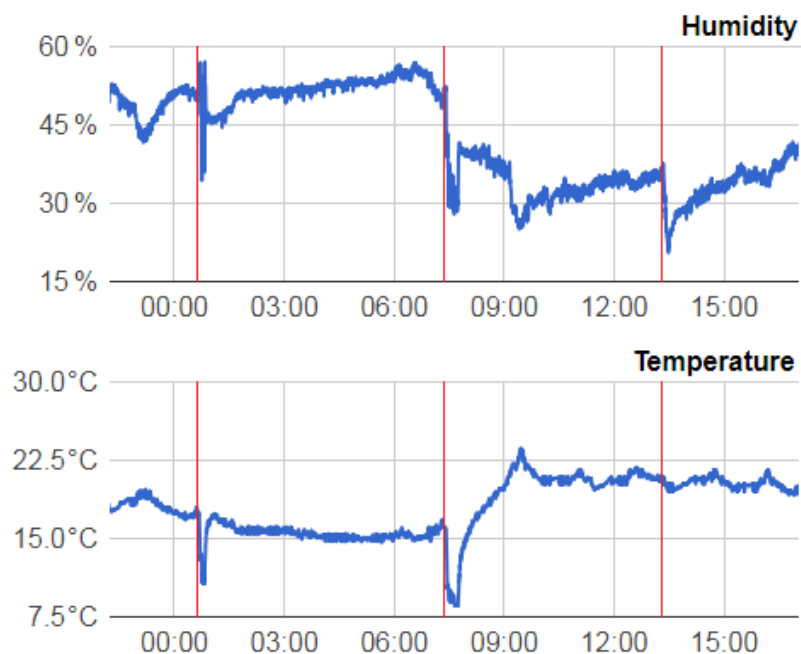
Otevření okna

Otevření okna je další událost, která vykazuje výrazné změny. V zimním období při otevření okna začne rychle klesat teplota i vlhkost. Při krátkodobém větrání se teplota rychle vrátí na původní hodnotu, vlhkost začne stoupat pozvolně. V grafech na obrázku 4.5 jsou vidět hned tři typy chování veličin při otevření okna v jeden březnový den. Tento senzor byl umístěn blízko okna. V čase 0:41 otevření způsobilo pokles teploty o 7 °C a vlhkost se pohybovala mezi 34% a 56%. V čase 7:25 se odehrál typický průběh. Vlhkost klesla z 52%



Obrázek 4.4: Graf vlhkosti se špičky v době sprchování uživatelů.

na 29% a teplota klesla o 8 °C. Pokles teploty byl příčinou nízké teploty venku, ale v čase 13:20 je vidět průběh, kdy vlhkost klesla o 12%, ale teplota se téměř nezměnila, jelikož venku byla velmi podobná teplota.



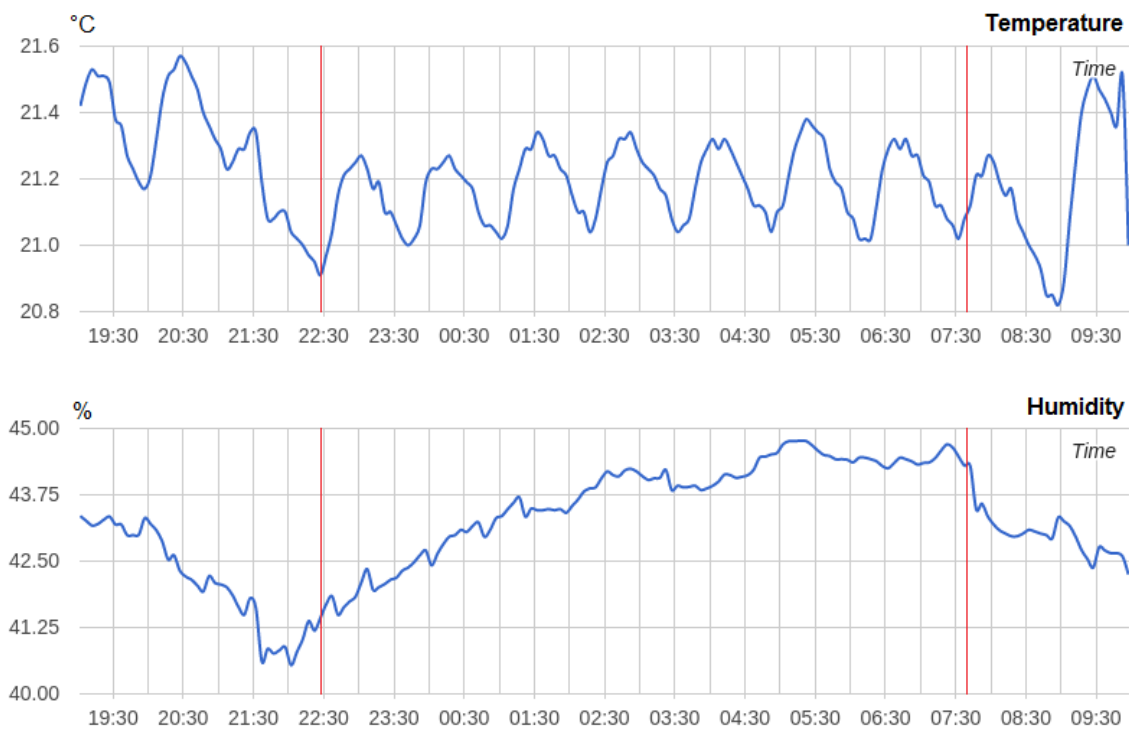
Obrázek 4.5: Grafy vlhkosti a teploty s vyznačeným časem otevření okna.

Vaření

V kuchyni se jako význačné události vykazují vaření a smažení či pečení. Vařením je myšlena příprava jídla, ze kterého se odpařuje voda a zvyšuje tím vlhkost. Charakteristika průběhu podobná sprchování v koupelně, ale u vaření je růst pomalejší a dlouhodobější. Prvně povyskočí teplota o pár desetín °C, což je způsobeno topnou spirálou, plynovým vařičem, atd. a o pár minut později následuje růst vlhkosti. V případě smažení a pečení se zvyšuje pouze teplota a vlhkost naopak mírně klesá. To je způsobeno absorpcí vlhkosti teplým vzduchem.

Spánek

Pokud máme senzor teploty a vlhkosti vhodně³ umístěný v ložnici, kde se pouze spí, lze z grafů vyčíst přibližné časy, kdy šel uživatel spát a kdy vstal. Příklad těchto událostí je vyznačen v grafech teploty a vlhkosti na obrázku 4.6 na straně 20. V případě více osob v jedné místnosti lze vyčíst časy, kdy šel spát poslední uživatel a kdy vstal první. V čase, kdy uživatel vstoupí do místnosti, hodnoty teploty a vlhkosti jemně zakolísají. Jakmile uživatel usne, jeho dýcháním začne v místnosti pozvolně stoupat vlhkost. V případě měření v bytě se jednalo o růst cca 1 % za hodinu. Ráno, když uživatel vstane, začne se po místnosti pohybovat a otevře dveře, vlhkost začne klesat. Teplota během spánku mírně klesá nebo je průměrně konstantní jelikož lidské tělo během spánku sníží svou teplotu a je zakryto pokrývkou (peřinou). K ránu teplota zase vzrůstá vlivem slunečního svitu, ale průběh této veličiny spíše závisí na tom, zda-li je topný systém zapnutý či vypnutý. Červené čáry značí, kdy šel uživatel spát a kdy vstával. V průběhu teploty jsou vidět *vlny*, které značí spínání ústředního topení. Průměrně je ale teplota stabilní. V grafu vlhkosti je zřetelný pozvolný růst, dokud uživatel nevstal.

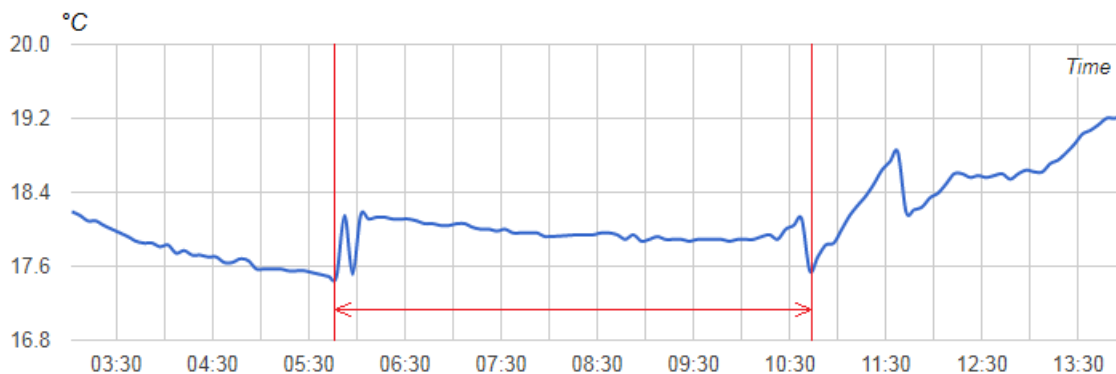


Obrázek 4.6: Průběh teploty a vlhkosti během spánku.

Nedostatky

Zobrazením dat v grafu také byla odhalena nepřesnost BeeOn senzorů. V případě teploty i vlhkosti se v náhodných časech objevují odchylky různě dlouhých intervalů. Příklad takové odchylky teploty je vidět na obrázku 4.7 v časovém intervalu mezi 6 a 11 hodinou.

³Vhodné umístění senzoru je co nejbližší středu místnosti a co nejdále od topného zařízení a míst, kde je průvan.



Obrázek 4.7: Viditelný posun teploty v čase mezi 6 a 11 hodinou.

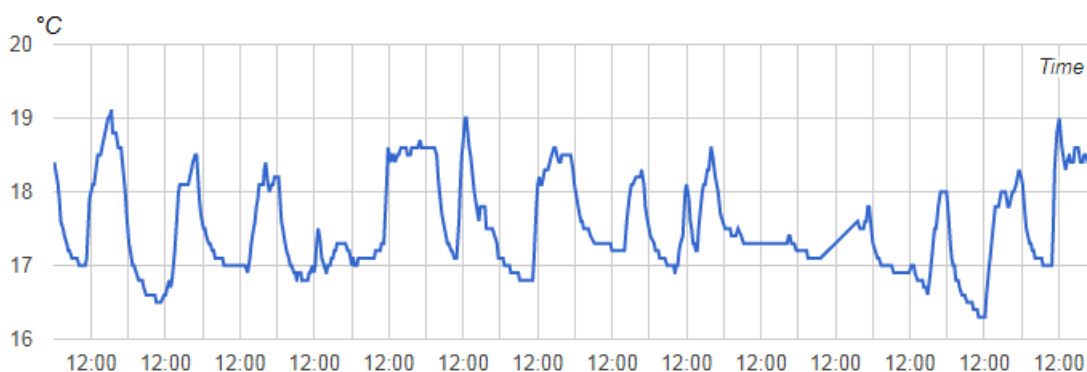
V případě teploty se jedná cca o $0,5^{\circ}\text{C}$ posuny, u vlhkosti cca 2%. Posuny se vyskytují nezávisle mezi teplotou, vlhkostí či stavem baterie. V případě Jablotron senzoru měřící teplotu se tyto anomálie nevyskytují. Data z BeeeOn senzorů bude tedy potřeba, pro odstranění tohoto šumu, vyhladit před dalším zpracováním.

Další nedostatek byl objeven u magnetického senzoru na dveřích koupelny. Pokud byly dveře pouze přizavřené, průvan způsoboval jejich pohyb a tím se detekovalo falešné otevírání a zavírání i v době, kdy v domácnosti nikdo nebyl.

V datech taky byly vyhledávány vzory, které by určili, zda se v místnosti někdo nachází, ale detekce přítomnosti osob v místnosti je velmi obtížná. Když se člověk pohybuje po místnosti, senzory měří větší šum v datech (větší výkyvy hodnot), ale zpětně nelze jednoznačně určit, zda-li jsou tyto výkyvy způsobené člověkem. Stejného naměřeného vzorku dat lze totiž docílit i malým občasným průvanem v jedné části místnosti, který si člověk ani nemusí uvědomovat.

4.2 Statistické údaje

Vypočtení statistických hodnot pro každou veličinu nám pomůže určit *běžné* chování domácnosti. Zjištěním těchto hodnot pak můžeme detekovat neobvyklé chování a včas na něj upozornit uživatele.



Obrázek 4.8: Teplota v obývacím pokoji rodinného domu – rozsah 14 dní.

Následující analýza se zaměřuje na teplotu obývacího pokoje v rodinném domě. Na obrázku 4.8 je zobrazen graf naměřených hodnot z teplotního senzoru Jabloton za 14 dní. V grafu je vidět, že se v místnost topí každý den přibližně od poledne do půlnoci. Jsou zde ale také vidět dny, kdy se v místnosti topilo méně kvůli nepřítomnosti, nebo více při zapomenutém vypnutí topení. V tabulce 4.1 jsou vypočteny běžné statistické funkce. Delší časový interval (14 dní) nám poskytne komplexnější informace, jelikož zahrnuje dny kratší i delší doby přítomnosti uživatele (pracovní i nepracovní dny). Krátký časový interval (1 den) je přínosný, pokud bychom udržovali statistiku pro každý den v týdnu zvlášť a tím pak reagovali na stejné hodnoty příslušně dle konkrétního dne.

	Posledních 14 dní	Posledních 24 hodin
Aritmetický průměr	17,52	17,88
Maximální hodnota	19,1	19
Minimální hodnota	16,3	17
Modus	17,3	18,4
Medián	17,3	18,2

Tabulka 4.1: Porovnání statistických hodnot pro 14 dní a 1 den.

Z vypočtených hodnot lze vyčíst, že se uživatel ani maximální naměřenou hodnotou nepřiblížil doporučené hodnotě 20 °C. Tento fakt poukazuje na nutnost systém nejdříve naučit chování uživatele a až pak povolit notifikace či rozhodování. Porovnáním průměru a modusu v posledním dni (neděle) lze odhadnout, že se uživatel snažil v místnosti zatopit, avšak porovnání průměru a modusu ve 14-denní statistice ukazuje na udržování spíše nižší teploty. Maximální a minimální hodnoty s průměrem je možné využít pro detekci výrazné odchylky oproti běžnému intervalu teploty.

4.3 Předzpracování dat

Jelikož se data v čase jeví jako signál, byly aplikovány nejběžnější metody pro zpracování signálů, které by mohly zlepšit kvalitu dat. Pro matematické metody byl použit jazyk a nástroj Matlab⁴.

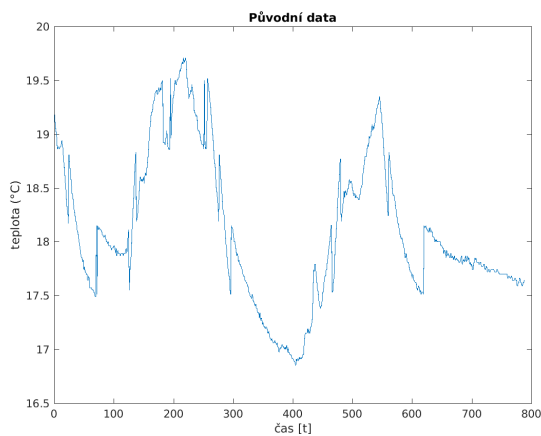
4.3.1 Vyhlazení signálu

Pro eliminování některých odchylek senzorů je vhodné data očistit. Data můžeme podobně jako signál vyhladit diskretní Fourierovou transformací. V jazyce Matlab je k dispozici pro tento případ funkce *filtfilt()*, která provádí digitální filtrování v nulové fázi. Výstupní signál dat z BeeeOn senzoru a vyhlazený signál těchto dat jsou zobrazeny na obrázku 4.9 a 4.10. V případě senzorů BeeeOn tedy bude potřeba data očistit před další zpracování.

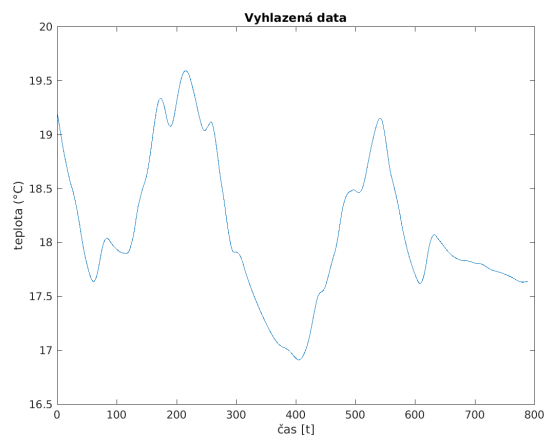
4.3.2 Velikost změny

Růst či pokles zvolené veličiny lze určit pomocí derivace. Tedy sumou rozdílů mezi sousedními hodnotami pro vybraný interval. Pokud máme spolehlivý senzor, který často vzorkuje (častěji než jednou za 5 minut), můžeme zvolit interval kratší (poslední půl hodinu

⁴Matlab – programovací jazyk, který je určený pro vědeckotechnické účely, simulace a různé výpočty. Zdroj: www.mathworks.com/



Obrázek 4.9: Původní získaná dat.



Obrázek 4.10: Vyhlazená data.

až hodinu). V běžných případech je ale vhodnější zvolit interval v řádu hodin (poslední 2–4 hodiny). Výsledná kladná hodnota značí růst, záporná pokles a velikost hodnoty značí rychlost změny.

Tuto metodu lze aplikovat pro úlohu sledování kritických změn, pro upozornění o prudké změny tlaku, teploty, atd. Výhodou derivace v tomto případě je, že je jednoduchá a nevdí jí malý šum v datech. Nevýhodou je, že potřebujeme pro každou veličinu a situaci mít předem nadefinovanou hodnotu určující, kdy lze míru změny označit za prudkou.

Kapitola 5

Návrh automatizovaného vyhodnocování

Dle výše uvedených poznatků byl vytvořen návrh pro klasifikaci událostí v domácnosti. Výsledná aplikace poběží jako samostatná služba na serveru. V první verzi se výsledná klasifikační aplikace nebude přímo integrovat do systému *BeeOn*. Teprve až na základě výsledků této práce bude aplikace přizpůsobena pro produkční nasazení.

Aby se inteligentní systém mohl korektně zachovat na změny v prostředí, je zapotřebí tyto změny správně detekovat a reagovat odpovídajícím způsobem. Tento zdánlivě jednoduchý úkol ale zesložitují problémy jako:

- různé typy měřeného prostředí (kancelář, domácnost, atd.),
- odlišné životní styly uživatelů (délka pobytu v domácnosti, průměrná požadovaná teplota, atd.),
- v neposlední řadě i počasí a roční období (otevření okna v zimě vs. v létě).

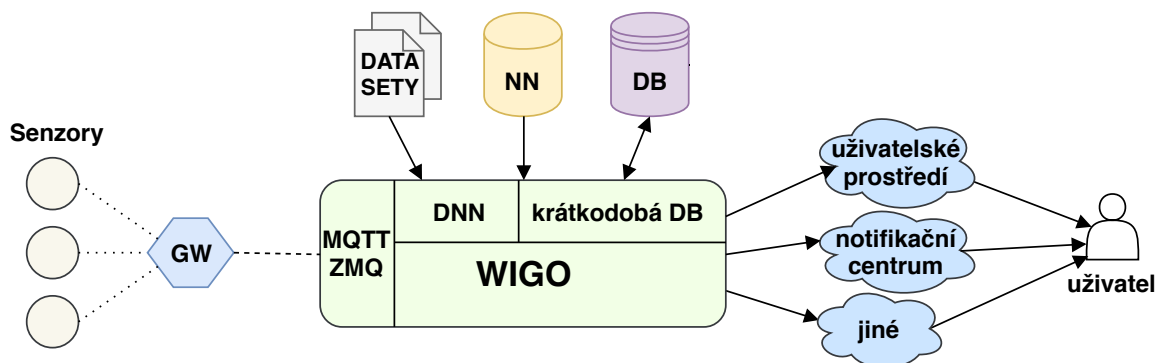
V poslední době se pro řízení a regulaci využívá novější vědecká disciplína – Soft Computing (SC). Tento obor zavádí takzvané „měkké“ požadavky na přesnost popisovaných jevů. Mezi hlavní zástupce SC patří: fuzzy logika, neuronové sítě a generické algoritmy. Tato práce je zaměřena na využití neuronových sítí [17].

Hlavní ideou je vytvořit tzv. umělou inteligenci na serveru, kde jsou přístupná historická data pro analýzu i současná data z bran pro aktuální vyhodnocování.

Pro některé složitější úlohy je vhodné použít neuronovou síť. Oproti pevným pravidlům, které je obtížné obecně definovat, dokáže klasifikátor založený na neuronové síti klasifikovat i změny/události v prostředí, které se dříve ještě nevyskytly s určitou pravděpodobností. Neuronové sítě přináší možnost naučit se běžné činnosti uživatele a neupozorňovat tak na události, které jsou zřejmé. Na druhou stranu je potřeba již mít velké množství ohodnocených dat pro trénování. Pro jednodušší úlohy stačí pouhá kontrola klíčových/hraničních hodnot, např. pomocí triggerů. Ty se ale mohou spustět i v nežádoucích případech. Typickým příkladem je sprchování. V takovém případě není potřeba uživatele informovat o tom, že v koupelně je zvýšená vlhkost. Je tedy vhodné přístupy rozumně zkombinovat.

5.1 Návrh architektury

Aplikace na vyhodnocování událostí v domácnosti byla pojmenována *WIGO*, což je zkratka pro „What Is Going On“. Cílem je zpracovávat data ze senzorů a zobrazovat aktuální stav domácnosti uživateli. Návrh aplikace je zobrazen na obrázku 5.1.



Obrázek 5.1: Návrh architektury aplikace WIGO.

Při spuštění aplikace se načtou uložené natrénované neuronové sítě. Pokud tyto sítě nejsou k dispozici, nebo vznikne požadavek na znovunatrénování, načtou datové sady definované v konfiguračním souboru a data z datových sad se zpracují dle datového modelu. Tato data budou následně použita pro natrénování a vyhodnocení přesnosti vytvořených neuronových sítí. Z každé datové sady vznikne jedna klasifikační hluboká neuronová síť (angl. Deep Neural Network, DNN).

Jakmile jsou neuronové sítě natrénované/načtené, získá se z databáze nebo konfiguračního souboru seznam bran, k nim jejich seznamy senzorů a zařazení do typů místností. Dle seznamů se inicializuje lokální krátkodobé úložiště a začnou se číst data ze senzorů pomocí MQTT nebo ZQM protokolu. Data se budou ukládat do lokálního dočasného úložiště. Od chvíle, kdy je nashromážděno dostatečné množství dat, začnou se počítat potřebné parametry pro vstupní vektor a vyhodnocovat neuronovou síť. Pokud by bylo nutné tento proces sběru dat urychlit, historická data požadované doby se mohou načíst z databáze systému *BeeOn*.

Jakmile je k dispozici dostatek historických dat, začnou se na základě nově přichozích dat počítat potřebné parametry pro vstupní vektor dle datového modelu a předkládat síti k vyhodnocování. Výsledkem bude událost, která právě probíhá. Zkontrolují se kritické hranice hodnot a získané informace se poskytnou uživateli v uživatelském rozhraní. Pokud by bylo nutné uživatele informovat o kritickém problému, zašle aplikace zprávu do notifikačního centra (a tím uživateli).

5.2 Detekované události

V předchozí kapitole 4 bylo z vizuální analýzy vysledováno, že některé uživatelské úkony lze dobře vyčíst pouze z teploty a vlhkosti. Tyto úkony by tedy měly být automatizovaně rozpoznatelné klasifikátorem. Klasifikátor v oblasti strojového učení je algoritmus, který řeší problém třídění dat. Na vstup je předložen vektor dat a klasifikátor se snaží tato data klasifikovat do jedné z výstupních tříd. Cílem klasifikátoru tedy bude detekovat a rozpoznat z přichozích dat události, které se dějí v dané místnosti.

Následující seznam událostí představuje výstupní třídy klasifikátoru, do kterých se budou zařazovat vyhodocená data:

0. **unknown** – žádná známá událost,
1. **shower** – uživatel se sprchuje v koupelně,
2. **after shower** – uživatel již ukončil sprchování, ale v koupelně je stále vysoká vlhkost,
3. **user sleep** – detekován spánek,
4. **close to heat** – senzor příliš blízko zdroji tepla nebo na přímém slunci,
5. **cooking** – vaření (především vody),
6. **frying** – smažení nebo pečení,
7. **open window** – otevření okna (dveří).

Tento seznam byl vytvořen na základě analýzy popsané v kapitole 4, ze které vyplývá i následující popis událostí. Každá zmíněná událost bude reprezentovat jednu výstupní třídu klasifikátoru. Tyto jednotlivé události vynikají tím, že každá z těchto událostí/činností ovlivňuje teplotu a vlhkost výrazným specifickým způsobem.

Událost **shower** značí, že se v koupelně někdo sprchuje. Událost je detekována z rychlého růstu teploty a vlhkosti vyplývající z analýzy popsané v kapitole 4. Jakmile uživatel ukončí sprchování a opustí koupelnu, následuje událost **after shower**, která informuje o tom, že se někdo sprchoval cca v poslední hodině. Hodnoty jsou stále nadprůměrné, ale pomalu či rychle klesají. Délka této události záleží na rychlosti odvětrávání, které je však vždy pomalejší než vzrůst vlhkosti při sprchování.

V kuchyni budou detekovány události **cooking** a **frying**, tedy vaření a smažení nebo pečení. Vaření je charakteristické růstem teploty a se zpožděním i růstem vlhkosti. Smažení naopak způsobuje pokles vlhkosti. Pokud se ale vaří a smaží současně, zvýšená vlhkost z vaření znemožní detekci smažení.

V obývací místnosti nebo ložnici se budou detekovat tři události: spánek uživatele – **user sleep**, otevření okna – **open window** a expozice senzoru na slunečním světle – **close to heat**. V případě detekce události spánku se předpokládá, že když jde člověk spát, tak obvykle pozavírá dveře, okna a zatáhne závěsy (žaluzie), aby se během spánku cítil bezpečně a nebyl rušen. Tím se značně omezí většina rušivých elementů a měřené hodnoty vykazují jen minimální změny. Pokud má uživatel pootevřené okno a teplotní rozdíly mezi místností a venkem nejsou příliš odlišné, detekce je stále možná. Pokud je okno plně otevřené, přesnost detekce bude pravděpodobně nízká.

Během slunečného dne často zasvítí sluneční paprsky okny dovnitř pokoje a vyhřívají jej. Pokud je senzor nevhodně umístěn právě na osvětlené ploše, mohl by vyvolat zbytečné varování o vysoké teplotě. Totéž, když je senzor umístěn poblíž tepelného zdroje (elektronické topení, chladič notebooku, trafo, atd.). Teplota pozvolna, ale jistě stoupá a u vlhkosti je mírný ale znatelný pokles.

Pro detekci události otevření okna bude samozřejmě záležet na velikosti rozdílu aktuálního stavu vzduchu v místnosti a venku. Možné je detekovat pouze akci otevření, nikoliv stav, zda-li je okno otevřené nebo zavřené. V zimním období je charakteristika změn velmi výrazná, což znamená, že by otevření mělo být dobře detekovatelné. Problém může být

v letním období, kdy jsou teploty velmi podobné uvnitř i venku. Pokud ale bude místnost s uživatelem nějakou dobu uzavřená a pak uživatel otevře okno, stále lze očekávat výrazné změny vlhkosti.

Abychom tyto události mohli využívat v automatické detekci, je nutné nejdříve ověřit, zda-li je možné tyto události detekovat s dostatečnou přesností. Jaká přesnost se dá považovat za dostatečnou nelze jednoznačně říct, ale cílem je dosáhnout co nejvyšší. Přesnost klasifikátoru pro danou datovou sadu se dá vypočítat tak, že se klasifikátor natrénuje na trénovací sadě a ověří se vůči testovací sadě. V našem případě získáme tyto dvě sady vybráním zamíchaných vzorků z předzpracovaných dat a rozdělením na 80 % dat pro trénovací sadu a 20 % dat pro testovací sadu.

5.3 Datové sady

Naměřená data je potřeba ručně nebo částečně ručně anotovat. K hodnotám je tedy potřeba připojit informaci o tom, co se v okolí senzoru dělo v době měření. K tomuto účelu byly využívány doplňující senzory jako např. detekce přítomnosti uživatele pomocí bluetooth, senzor pohybu, detektor otevření dveří, atd. a ručně psaný deník činností. Tím byla vytvořena potřebná datová sada nutná pro natrénování klasifikačních sítí.

Datové sady slouží pro natrénování neuronových sítí vzorovými daty. Sít si podle těchto referenčních dat nalezne spojitosti mezi parametry a je pak schopná rozpoznávat a klasifikovat události z nových dat. Pro strojové učení byly datové sady vytvořeny z reálně naměřených dat. Data byla sbírána ve studenském bytě, rodinném bytě, rodinném domě a částečně v laboratoři. Následně byla ručně anotována podle dění v konkrétní místnosti. Data, která nebylo možné jednoznačně klasifikovat, byla eliminována pro vytvoření kvalitních jednoznačných dat.

Všechny datové sady jsou v uloženy jednotném textovém formátu CSV, kde oddělovačem je čárka a každý řádek obsahuje:

- **timestamp** – časová značka sběru hodnot v unixovém formátu,
- **temperature** – vnitřní teplota s přesností na desetiny,
- **humidity** – vnitřní relativní vlhkost s přesností na desetiny,
- **event** – typ události vyvolané uživatelem nebo samotným prostředím.

Inteval mezi měřeními byl nastaven na 15 s, reálně se pohybuje mezi 17–23 s. Každá datová sada reprezentuje jednu místnost. Jelikož některé detekovatelné události jsou si dosti podobné, jsou klasifikátory děleny podle místností. Největší podobnost se vyskytuje u detekce sprchování v koupelně a vaření vody v kuchyni. V případě sprchování rychle vzroste vlhkost a teplota. V případě vaření lze docílit stejného efektu, pokud se senzor vyskytuje příliš blízko vařící vodě. V případě jiných místností, kde takto rychlý vzrůst vlhkosti není očekáván, není potřeba rozpoznávat o jakou událost se jedná, ale hned uživatele upozornit na neočekávané chování v dané místnosti. Vytvořením více samostatných klasifikátorů se tedy zvýší přesnost detekce událostí typických pro danou místnost.

Je nutné počítat s tím, že data v datovém souboru nejsou ideální, jelikož vychází z reálného měření. Senzory někdy ztratili spojení s bránou na delší dobu (v rádech minut až hodin) nebo se vybitly baterie. Tím pádem data v tyto časové intervaly chybí. Dalším zdrojem chybějící dat v časových řadách je ruční eliminace nekvalitních či neznámých hodnot.

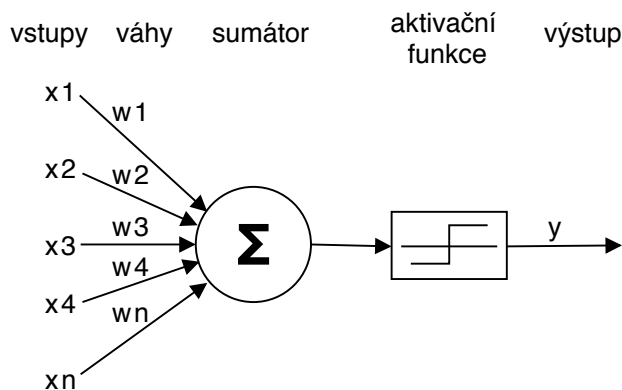
Navíc jeden datový soubor jednoho typu místnosti obsahuje data z více domácností, tedy intervaly časových řad se mohou překrývat.

Silným nedostatkem vytvořených datových sad je jejich neúplnost vůči ročním obdobím. Data byla sbírána konkrétně dva měsíce v zimním období (listopad, prosinec) a přibližně tři měsíce v jarním období (únor, březen, duben). Pro vytvoření komplexnějších datových sad by bylo zapotřebí sbírat a hodnotit data po celý rok z několika odlišných domácností.

V zimním období křivka teploty se zpožděním kopíruje zapínání a vypínání topení. Jedná se přibližně o jednostupňové odchylky v závislosti na vzdálenosti od tepelného zdroje. Získáváme tím sice informaci o spínání tepelného zdroje, ale může vést ke ztrátě jiných neznámých informací. Tento efekt ale překvapivě nemá vliv na vlhkost, tedy události vázané především na vlhkost nejsou ovlivněny.

5.4 Umělá neuronová síť

Umělá neuronová síť je silný výpočetní model využívaný pro strojové učení. Jedná se o strukturu umělých neuronů, která se snaží napodobit činnost lidského mozku, tedy biologickou neuronovou síť. Základním prvkem jsou neurony, které jsou vzájemně propojeny spoji ohodnocenými vahami. Tyto váhy se adaptují (učí se) na základě trénovacích dat. Model neuronu je zobrazen na obrázku 5.2.



Obrázek 5.2: Model umělého neuronu.

Každý neuron má x_1 až x_n vstupů ohodnocených vahami w_1 až w_n . Na základě těchto vah mohou být jednotlivé vstupy zvýhodněny nebo potlačeny. Tyto vstupy s váhy vstupují do sumátoru a vypočítá se z nich vnitřní stav neuronu $u = \vec{x} \cdot \vec{w}$. Výstupem je hodnota odvozena ze stavu podle aktivační funkce, která má definovaný práh a může být skoková, po částech lineární, spojitá, atd. V poslední době je pro hluboké učení velmi často využívána aktivační funkce ReLU¹. Funkce má výstup 0 pro hodnoty vstupu menší jak 0 a stejný výstup pro kladný vstup. Její vzorec je:

$$f(x) = \max(x, 0)$$

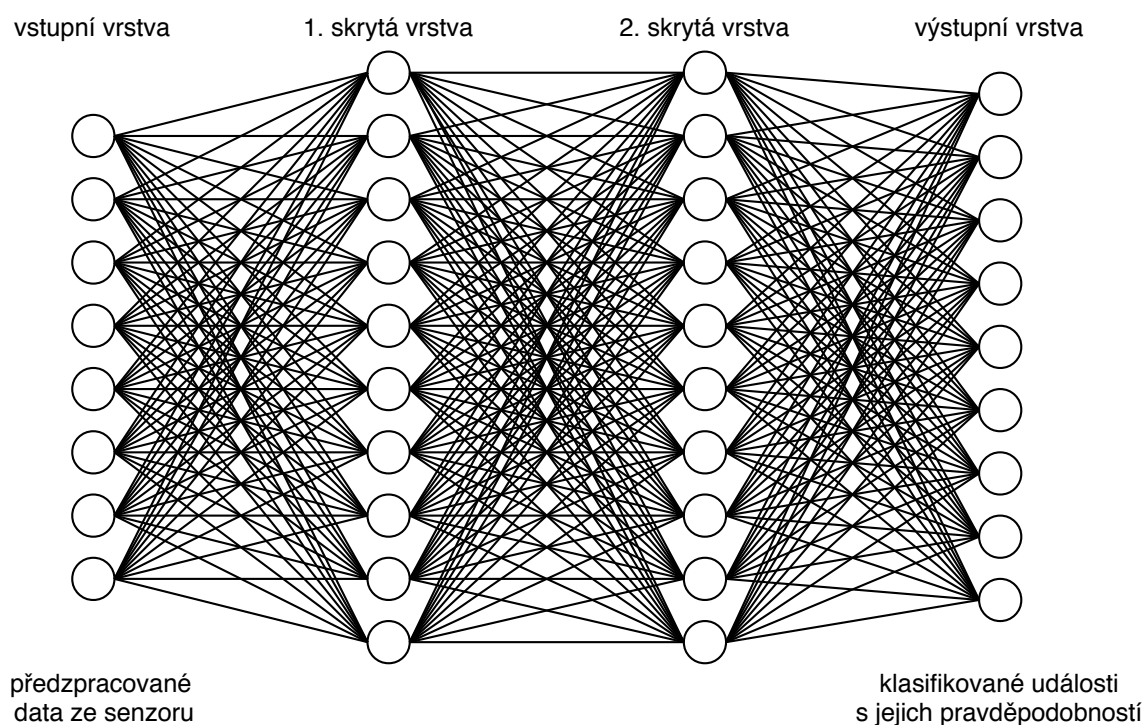
Neuron může mít libovolný počet vstupů, ale pouze jeden výstup. Síla neuronů se projeví až propojením neuronů mezi sebou do větších struktur, např. jako je na obrázku 5.3. Váhy mezi neurony si nastavuje algoritmus sám na základě učení, nedokážeme tedy přímo ovlivnit, jaké vztahy si neurony mezi sebou vytvoří.

¹ReLU - Rectified Linear Unit

Pro vytvoření sítě je potřeba velké množství dat. Tato data se typicky náhodně rozdělí na trénovací a testovací data v poměru 80:20, popřípadně 66:33. Na trénovacích datech se neuronová síť naučí a vytvoří si váhované spoje mezi neurony. Testovací data pak slouží k vyhodnocení, zda nedošlo k *přeučení sítě* – příliš silné vazby mezi neurony pro rozpoznání pouze jedné třídy.

Nejpoužívanější typy sítí jsou: vícevrstvé perceptronové sítě, Kohonenovy sítě, Bayesovské sítě lišící se jiným způsobem propojení neuronů. Neuronové sítě se používají na rozpoznávání a kompresi zvuků nebo obrazů, filtrování e-mailů, detekce podvodů, předvídání časových řad a spoustu další aplikací [5, 24].

K vytvoření sítě je potřeba určit vrstev a počet neuronů v každé vrstvě. Čím více neuronů síť disponuje, tím lepších výsledků může dosahovat, ale úměrně k tomu potřebuje odpovídající (velké) množství kvalitních trénovacích dat [8]. Na obrázku 5.3 je zobrazena předpokládaná vícevrstvá neuronová síť pro klasifikaci událostí jedné místnosti.



Obrázek 5.3: Předpokládaná vícevrstvá neuronová síť.

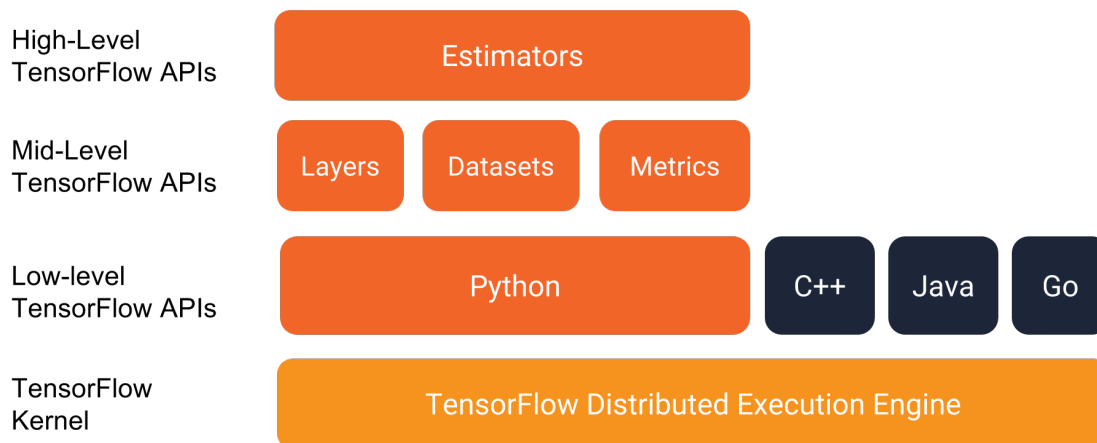
Na tvorbu neuronové sítě pro klasifikaci byla zvolena knihovna *Tensorflow* od společnosti Google (Google Brain Team), která byla prvně veřejně vydána v roce 2015 pod licencí *Apache 2.0 open source*.

Tato matematická knihovna v dnešní době patří mezi populární knihovny strojového učení. Pro své jednoduché používání a silné výpočetní možnosti je tato knihovna využívána téměř na všechny služby poskytované Googlem, tak také velkými korporacemi jako např. Coca-Cola, Twitter, Intel, Lenovo, AMD, NVIDIA, eBay, atd. Nabízí celou řadu různých nástrojů, které umožňují vytvářet modely na požadované úrovni abstrakce. Jádrem knihovny je napsáno v jazyce C++ a poskytuje nízko-úrovňové API² pro jazyky Python, C++, Go, Java, Haskell, atd. Pro jazyk Python navíc nabízí středně-úrovňové API pro zpracovávání

²API – Application programming interface

datasetů a metrik, a vysoko-úrovňové API pro tzv. Estimators (česky *rozpozávače*). Úrovně API jsou rozbrázeny na obrázku 5.4.

Tensorflow je možné sputit na více CPU a GPU, na Linuxu, MacOS nebo Windows, ale také i na mobilní platformách jako Android a iOS. Pro své výpočty využívá grafy datových toků, které pracují s vícerozměrnými datovými poli pojmenované jako „tenzory“ [25, 7]



Obrázek 5.4: Struktura knihovny Tensorflow. Zdroj: www.tensorflow.org

5.5 Úprava vstupních dat

Pro detekci a klasifikaci zmíněných událostí bude využita vícevrstvá neuronová síť. Na vstup sítě se budou předkládat upravená (předzpracovaná) data ze senzorů. Je tedy potřeba definovat a vytvořit vhodné vstupní parametry. Hodnoty ze senzorů přichází jako časová řada s nepravidelnými intervaly mezi vzorky. Příchozí data obsahují tři informace: časovou značku, teplotu a vlhkost.

Z časové značky získáme informaci, o jakou část dne se jedná (ráno, odpoledne, noc, atd.). Klasifikátor časovou značku zpracuje v podobě desetinného čísla t ve tvaru:

$$t = h + transform(m)$$

kde h je počet hodin a funkce $transform(m) \in \langle 0, 1 \rangle$ transformuje počet minut v rozsahu 0–59 do intervalu $\langle 0, 1 \rangle$. Pokud bychom hodinu a minutu reprezentovali samostatně jako oddělené parametry, mohlo by dojít k tomu, že by se parametru minuty mohl přiřkládat větší význam, než je nutné. Parametr měsíce, reprezentované celým číslem, nepřímou přináší informaci o ročním období.

Hodnoty teploty a vlhkosti nemůžeme předkládat na vstup přímo, jelikož každá domácnost je odlišná a samotná data mnoho neříkají. Informace o změně prostředí je nesena v diferenci hodnoty, tedy surová data (teplota a vlhkost) nesou informaci o prováděné události pouze v seřazené sekvenci. Zajímá nás tedy jak se data mění v čase. Tyto změny mají stejný průběh v libovolné domácnosti. Potřebujeme tedy data nejdřív požadovanou dobu (dle požadavků parametrů) dočasně ukládat. Výpočet změn se provede derivací, Získá se tak informace, zda-li veličina (teplota či vlhkost) roste či klesá a jak rychle.

Jedna z možností na získání vstupních parametrů z teploty a vlhkosti je rozsekání definovaného časového intervalu hodnot na krátké úseky, tedy např. výpočet derivací pro intervaly 0.–1. minuta, 1.–2. minuta, atd. Tím bychom ale získávali výsledek klasifikace jen jednou za sekvenci a riskovali, že se nedetekují význačné události na hraně mezi sekvencemi [26].

Jako parametr nelze použít rozdíl posledních dvou hodnot, protože mezi jednotlivými měřeními a posláním na server mohou nastat výpadky. Mezi těmito dvěma hodnotami tedy mohou být různě velké časové rozdíly. Parametry se tedy budou vytvářet derivací z různých časových intervalů. Tím pádem nebude nutné data vyhlazovat, jak bylo popsáno v sekci 4.3.1 předzpracování dat. Testovány budou parametry vytvořené z dat o délce např. posledních 30 min, 10 min, 5 min, 2 min, 1 min, 30 s pro teplotu i vlhkost.

Dalším vstupním parametrem bude odchylka od průměru. Aplikace si bude pamatovat průměr posledních sedmi dní, aby byl pokrytý celý týden změn. Od tohoto průměru bude vypočítán rozdíl s aktuální hodnotou.

Další vstupní parametry by mohli být vytvořeny z externích zdrojů. Např. pro detekci otevření okna je užitečné znát venkovní teplotu a vlhkost z nejbližší meteostanice. To by ale vyžadovalo na serveru samostatný modul, řešící logiku získávání dat z meteo-serverů, jejich uchovávání a rozdělení dle GPS oblastí. Tento modul v systému zatím neexistuje, proto tato možnost obohacení vstupu pro neuronovou síť není v této práci více rozebírána.

5.6 Vstupní vrstva neuronové sítě

Data z načtené datové sady je potřeba upravit do vhodného formátu, tedy předzpracovat, než se budou předkládat neuronové síti, jak v případě učení, tak v případě vyhodnocování. Zde existuje velké množství kombinací parametrů.

Hlavní parametry budou vypočítávány derivací pro zvolený časový interval dat. Sumou rozdílů hodnot seřazené dle časové značky se získá metrika informující o růstu nebo poklesu a rychlosti této změny.

Neuronová síť interně pracuje s normalizovanými hodnotami. Normalizace dat tedy může vést k lepším výsledkům a normalizace se provede transformací hodnot na interval $(0, 1)$. Pro tento převod je potřeba určit minimální a maximální hraniční hodnoty. Pro teplotu se v databázi vyskytují hodnoty přibližně od -10°C až po 50°C . Avšak zpracovávat se budou pouze rozdíly, tedy výsledná hodnota derivace se normalizuje v rozsahu -5 až 5°C . Vlhkost se normalizuje pro změny v rozsahu -20 až 20% .

Vstupní vektory parametrů, u kterých bude měřena přesnost jsou následující:

1. hour, minute, temp1m, hum1m, event
2. hour, minute, temp2m, temp1m, hum2m, hum1m, event
3. month, time, temp2m, temp1m, hum2m, hum1m, event
4. month, time, temp2mNorm, temp1mNorm, hum2mNorm, hum1mNorm, event
5. temp2m, temp1m, hum2m, hum1m, event
6. temp2mNorm, temp1mNorm, hum2mNorm, hum1mNorm, event
7. temp5m, temp2m, temp1m, hum5m, hum2m, hum1m, event

8. temp10m, temp5m, temp2m, temp1m, hum10m, hum5m, hum2m, hum1m, event
9. temp2m, temp1m, temp30s, hum2m, hum1m, hum30s, event
10. temp30m, temp10m, temp5m, temp2m, temp1m, temp30s, hum30m, hum10m, hum5m, hum2m, hum1m, hum30s, event
11. temp10m, temp5m, temp1m, hum10m, hum5m, hum1m, event
12. temp5m, temp4m, temp3m, temp2m, temp1m, hum5m, hum4m, hum3m, hum2m, hum1m, event
13. tempAvgDiff, temp10m, temp5m, temp2m, temp1m, humAvgDiff, hum10m, hum5m, hum2m, hum1m, event
14. tempAvgDiff, temp5m, temp2m, temp1m, humAvgDiff, hum5m, hum2m, hum1m, event

Hodina (hour) je celočíselné číslo v rozsahu 0–23, minuta (minute) v rozsahu 0–59 a měsíc (month) v rozsahu 1–12. Čas (time) je spojení hodiny a minuty do desetinného čísla v rozsahu 0.0–23.99. Pro teplotu (temp_) a vlhkost (hum_) byli počítány derivace různých časových intervalů. Délka intervalu je zaznamenána v názvu parametru, např.: *hum3m* značí derivaci vlhkosti za poslední 3 minuty. Derivace hodnoty, které byly normalizovány jsou označeny příponou *Norm*. Poslední typ parametru je *_AvgDiff*, který značí rozdíl aktuální hodnoty od průměrné hodnoty za posledních 7 dní. Výsledky jednotlivých vstupních vektorů jsou vyhodnoceny v 7. kapitole výsledků.

5.7 Detekovaná událost a kontrola kritických hranic

Jeden z cílů výsledné aplikace je i hlídání kritických hodnot nebezpečné pro člověka. Po vyhodnocení pravděpodobné množiny událostí, které se aktuálně odehrávají ve sledovaném prostředí, může následovat kontrola kritických hranic hodnot. Kritické hodnoty jsou stavy prostředí, které není možné neuronovou sítí naučit, jelikož nejsou předem známé. Navíc je potřebujeme detekovat ihned a s vysokou jistotou. Účelem této úlohy je kontrola bezpečnosti prostředí a zdravého klimatu v domácnosti.

Obecně, pokud nebude detekována žádná známá událost, ale bude překročena definovaná hraniční hodnota, bude uživatel na tento problém hned upozorněn (notifikací, světlem, atd.).

V případě detekování známé události a překročení hraniční hodnoty záleží vyhodnocení na typu místnosti. Uživatel bude upozorněn až v případě, že překročení hodnoty trvá dlouho a je stále rostoucí. Problém je definovat, co znamená dlouho. Například uživatel se může sprchovat několik desítek minut, nebo se může vystřídat více uživatelů rychle za sebou. To vede k dlouhodobě zvýšené vlhkosti. V takovém případě bude uživatel upozorněn po jedné hodině od první detekce problému.

Aby uživatel nebyl zahlcen notifikacemi, bude vytvořeno více úrovní notifikací. Množství propuštěných notifikací a agregace bude mít na starosti již notifikační centrum.

První úroveň bude pouze informovat o nevhodném prostředí, tedy, že byly naměřeny hodnoty vyšší nebo nižší než obvykle. Tyto hodnoty by se měly časem přizpůsobit podle dlouhodobého měření v domácnosti. Uživatel bude moci tyto hodnoty upravit dle vlastních preferencí nebo tuto úroveň notifikací úplně vypnout. Ve výchozím nastavení budou

nastaveny minimální a maximální hodnoty na 18 °C a 25 °C pro teplotu a 40 % a 60 % pro vlhkost.

Druhá úroveň bude varovat na nevhodné životní podmínky, kterým je potřeba věnovat pozornost. Hodnoty byly stanoveny dle norem, popsané kapitole analýzy a empirických zjištění z měření. Minimální a maximální hodnoty na 15 °C a 29 °C pro teplotu a 25 % a 70 % pro vlhkost.

Kapitola 6

Implementace navrženého řešení

Tato kapitola popisuje všechny potřebné informace k detekci definovaných událostí popsaných v předchozí kapitole. Vytvořený program běží na serveru, kde má možnost přístupu k aktuálním datům ze senzorů. Přístup k databázi s historickými daty není nutný, ale může být výhodný, pokud bychom požadovali výsledky vyhodnocování ihned po spuštění programu.

6.1 Jádro klasifikátoru

Pro vytvoření a natrénování neuronových sítí je využita knihovna TensorFlow zmíněná v návrhu v kapitole 5. Konkrétně je využit estimátor `DNNClassifier` z nejvyšší API vrstvy. Díky něj je možné jednoduše vytvořit vícevrstvou (hlubokou) neuronovou síť dle zadaných argumentů.

Mezi povinné argumenty patří:

- *feature_columns* – definování vstupního vektoru (jména sloupců, typ hodnot, atd.),
- *hidden_units* – definování počtu skrytých vstev sítě a počet neuronů v každé vrstvě,
- *n_classes* – počet výstupních tříd (velikost výstupní vrstvy).

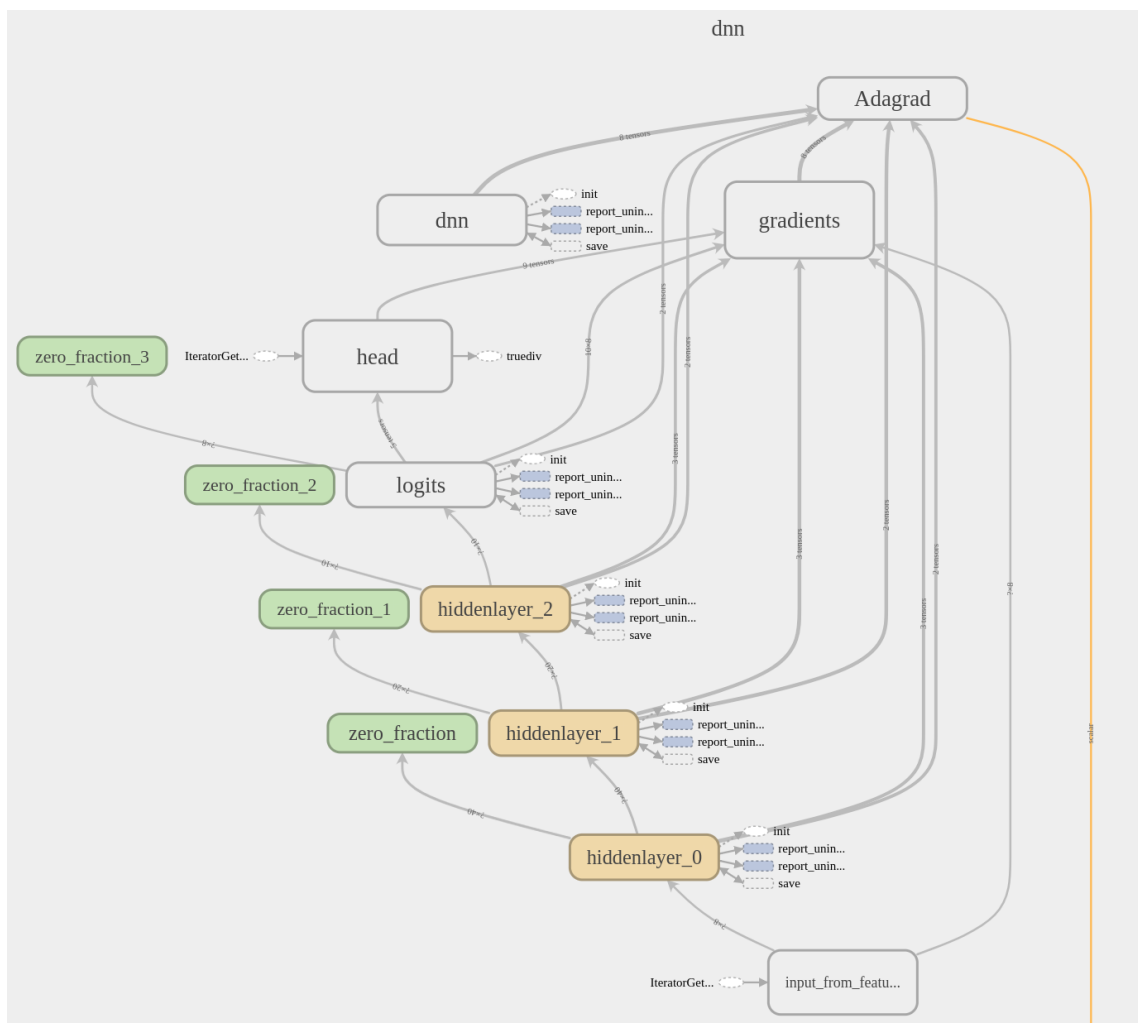
Největší vstupní vektor pro síť má až 12 parametrů a výstupní vektor 8 tříd. Hloubka sítě byla zvolena na tři skryté vrstvy. Počet skrytých neuronů by měl být přibližně 2/3 velikosti vstupní vrstvy plus velikost výstupní vrstvy. Pro jednotlivé skryté vrstvy tedy bylo zvoleno 20, 20, 10 neuronů [10].

Ostatní argumenty jsou volitelné nebo předdefinované. Argumentem *model_dir* se definuje cesta adresáře, kam se má uložit natrénovaná síť. Při novém spuštění programu je tedy pak možné tuto natrénovanou síť již pouze načíst. Argument *activation_fn* definuje aktivační funkci pro neurony v síti. Ve výchozím nastavení je využívána velmi rozšířená funkce ReLU zmíněná v kapitole 5.4.

Pro natrénování sítě požaduje metoda *train()* trénovací data a argumenty: *batch_size*, který definuje kolik dat se má načíst pro každou iteraci učení, a *train_steps* určující počet kroků. Výsledkem vyhodnocení sítě metodou *evaluate()* jsou hodnotící metriky: *přesnost (accuracy)* a *ztráta (loss function)*. Přesnost určuje, v kolika případech síť klasifikovala vstupní vektor do správné třídy. Ztrátová funkce mapuje sadu hodnot parametrů pro síť

na skalární hodnotu, která udává, jak dobře tyto parametry dosáhnou úlohy, kterou má síť provádět¹.

Součástí knihovny TensorFlow je i grafický nástroj TensorBoard, sloužící pro snadnější pochopení, ladění a optimalizaci programů využívající TensorFlow. Pomocí TensorBoard je možné vykreslit kvantitativní metriky a vizualizovat neuronovou síť jako graf. Na obrázku 6.1 je podstatná část grafu zobrazující natrénovanou síť klasifikující události v koupelně. Nejspodnější blok představuje vstupní vektor. Odspodu nahoru jsou zobrazeny bloky skryté vrstvy a nahoře pomocné logické bloky agregující všechny informace do výstupního vektoru.



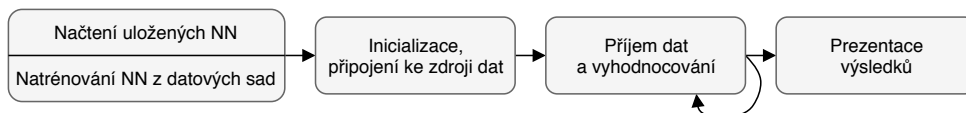
Obrázek 6.1: Repräsentace vytvořené DNN nástrojem TensorBoard.

6.2 Hlavní činnost aplikace

V rámci této práce byl vytvořena aplikace v Pythonu 3.6 typu „proof of concept“ pojmenována jako *WIGO*. Aplikace pro svou činnost využívá, kromě standartních pythonových knihoven, také navíc knihovny: tensorflow, numpy, pandas, paho.mqtt.

¹Definice ztrátové funkce – <http://theanets.readthedocs.io/en/stable/api/losses.html>

Knihovna *Tensorflow* slouží primárně pro strojové učení, tedy tvorbu neuronových sítí. *Numpy* je balíček matematických metod pro vědecké výpočty a práci s multidimenzionálními poli. V aplikaci je využíváno především na výpočet derivací. *Pandas* poskytuje rychlé a flexibilní datové struktury navržené pro snadnou a intuitivní práci. Prakticky se využívá pro analýzu dat. V aplikaci je použita pro práci s trénovacími a testovacími daty. *Paho.mqtt* je klientská knihovna v jazyce Python, která implementuje protokol MQTT² pro připojení na MQTT broker a posílání zpráv mezi klienty. Knihovna je využita pro příjem dat ze senzorových sítí.



Obrázek 6.2: Zjednodušené schéma úkonů aplikace WIGO.

Program *WIGO* má tři základní logické části:

- vytvoření/načtení neuronových sítí,
- sběr a vyhodnocování,
- reprezentace výsledku.

Na obrázku 6.2 je zobrazeno zjednodušené schéma posloupnosti úkonů. V první části se načte zadaná datová sada, natrénuje a ohodnotí se neuronová síť. Pokud již síť byla vytvořena dříve a datová sada nebyla změněna, načte se uložená neuronová síť z posledního natrénování. Z každé jedné datové sady se vytvoří jedna samostatná síť/klasifikátor.

V druhé části se provede inicializace krátkodobého úložiště a provede se připojení na MQTT broker. Zde naslouchá příchozí data ze senzorů, přeposílané přes bránu na server. Každá brána posílá jednotné JSON zprávy na broker se svým unikátním ID – *gateway_id*. Pokud je toto id brány známé (z konfiguračního souboru), přečte se obsah zprávy a zjistí se, v jaké místnosti se nachází senzor. Na základě této informace se vybere klasifikátor, data se vyhodnotí a výsledek uloží do krátkodobého úložiště.

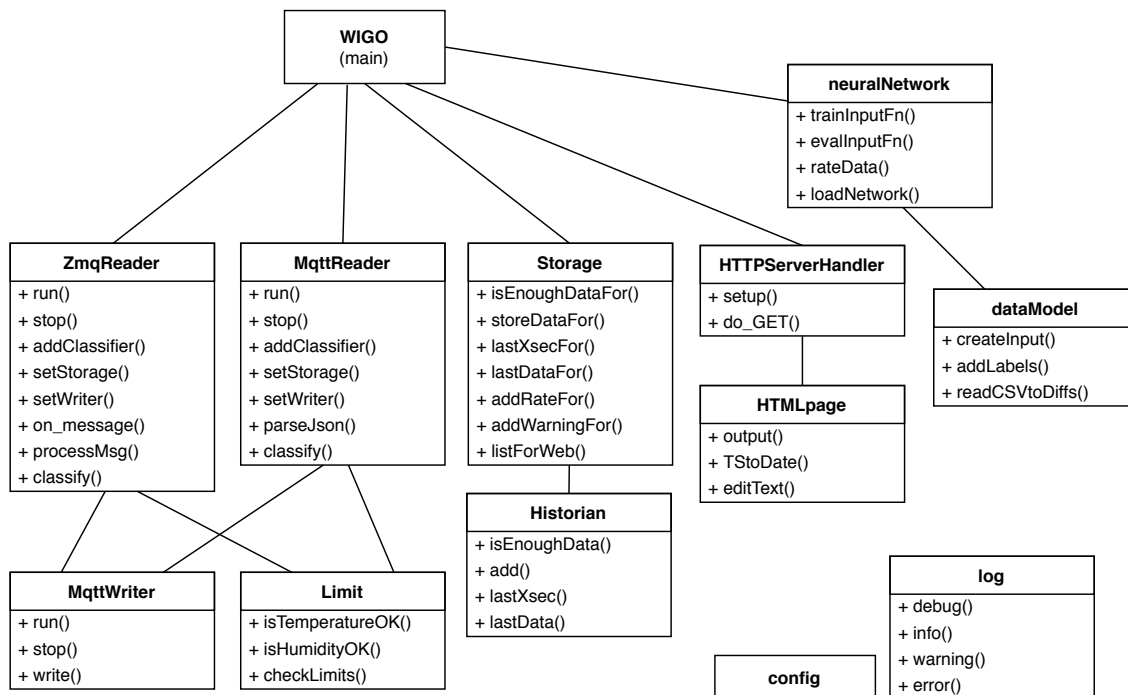
Třetí část je reprezentační, tedy zobrazení výsledku uživateli. K tomuto účelu bylo vytvořeno webové rozhraní, kde může uživatel vidět aktuální hodnoty, čas posledního měření, odhad probíhané události s procentuální pravděpodobností. Notifikace o změně události v místnosti a překročení kritických hodnot, jsou publikovány na MQTT, kde je může číst notifikační centrum.

Pro spuštění programu *WIGO* nejsou potřeba žádné argumenty, jen je nutné mít nainstalované všechny potřebné knihovny. Seznam senzorů, které se mají sbírat a vyhodnocovat, se nachází v *devices.py*. Veškeré potřebné nastavení a informace se načítají ze souboru *config.py*.

6.3 Popis tříd

V této části je popis nejdůležitějších tříd a pomocných funkcí. Diagram tříd je zobrazen na obrázku 6.3. V diagramu jsou u jednotlivých tříd zmíněny nejdůležitější veřejné metody.

²MQTT – protokol, který poskytuje posílání zpráv pomocí modelu publikování a odběru.



Obrázek 6.3: Diagram třít aplikace WIGO.

- **Wigo**

Hlavní soubor programu. Spouští jádro TensorFlow a nastavuje jeho loggovací úroveň. Následuje načtení potřebných klasifikátorů a inicializace krátkodobého úložiště. Dále vytvoří instanci třídy pro příjem dat, předá ji reference na vytvořené klasifikátory a úložiště, a spustí čtení dat ze senzorů. Jako poslední se spustí HTTP server, který spustí smyčku pro přijímání žádostí na stránku.

- **Historian**

uchovává nasbírané data jednoho senzoru po definovanou dobu, výchozí doba je 10 min, jelikož více pro tvořené parametry není potřeba. Data jsou ukládána ve frontě podle časové značky. S každým přidáním nových dat se vymažou data starší než 10 min. Třída poskytuje metodu *lastXsec()* pro získání derivace hodnot za posledních *X* sekund, metodu *lastData()* pro získání posledních hodnot a časové značky pro zobrazení na webové stránce.

- **Storage**

Storage je manažer obsluhující krátkodobé úložiště. Spravuje data v objektech *Historian* pro každé zařízení a další informace jako výsledky vyhodnocení, případné notifikace, atd.

- **MqttReader**

Třída, zajišťující připojení a komunikaci s MQTT broker. Přijímá data od senzorů a vykonává operace pro vyhodnocení dat. Příchozí zprávy jsou přijímány metodou *on_message()*. Zde se zkontroluje, zda-li je ID brány v seznamu pro vyhodnocování. Pokud ano, přečte se zpráva v metodě *parseJson()* a vyparovaná data se předají ke klasifikaci metodě *classify()*. Zjistí se, zda-li je k dispozici dostatek dat v krátko-

dobém úložišti, načtou se a vyhodnotí příslušným klasifikátorem. Výsledek se uloží do krátkodobého úložiště pro možnost zobrazení uživateli.

- **ZmqReader**

Třída, zajišťující připojení a komunikaci protokolem ZeroMQ. Pro příjem senzorických dat přímo na serveru. Serverová aplikace přijímá data ze zaregistrovaných bran a poskytuje pomocí ZeroMQ. Komunikace probíhá na modelu push-pull. Server je připojen na *push* socket, který distribuují odesílaná data na klienty (tato třída) kteří jsou připojeni na *pull* socket. Následné zpracování je stejné jako ve třídě `mqtReader`.

- **MqttWriter**

Třída, zajišťující připojení a komunikaci s MQTT broker společný s notifikačním centrem. Výsledky vyhodnocení jsou posílány (publikovány) ve formátu JSON. Každá zpráva obsahuje ID brány, ID zařízení, detekovanou událost, její přesnost, informační či varovnou zprávu a kód zprávy.

- **HTMLpage**

Obsahuje třídu `HTTPServerHandler`, která přijímá požadavky na získání webové stránky se všemi informacemi – aktuální data a detekovanou událost, popřípadně zprávu notifikace. Struktura `HTMLpage` obsahuje kostru webové stránky a metodu pro vyčtení informací z krátkodobého úložiště.

- **Limit**

Třída obsahující hraniční hodnoty pro měřené fyzikální veličiny dvou úrovní. První úroveň pouze informuje o nestandardní naměřené hodnotě, která je vyšší nebo nižší než obvykle. Druhá úroveň varuje na nevhodné životní podmínky, kterým je potřeba věnovat pozornost. Hodnoty byly stanoveny dle norem, popsane kapitole 4 a empirických zjištění z měření. Hlavní metodou je `checkLimits()`, která vrací kód problému a zprávu.

- **neuralNetwork**

Soubor pomocných funkcí obsluhující neuronové sítě. Funkce `loadNetwork()` zjistí, zda-li již existuje uložená natrénovaná síť. Pokud ano, načte ji a vrátí na výstup. Pokud ne, načtou se CSV data z datové sady, data se upraví dle datového modelu, náhodně zamíchají a rozdělí na trénovací a testovací části. Inicializuje se estimátor `DNNClassifier` s parametry definované v konfiguračním souboru a natrénuje se trénovacími daty. V posledním kroku se vyhodnotí vytvořený klasifikátor na testovacích datech a předá na výstup. Vyhodnocování příchozích dat ze senzorů se provádí v `rateData()`. Argumenty funkce jsou klasifikátor, ID klasifikátoru a data k vyhodnocení. Výsledkem klasifikace je vektor pravděpodobností pro každou událost, přičemž výstupem celé funkce je pouze název a pravděpodobnost nejlépe ohodnocené události. Ostatní funkce `train_input_fn()` a `eval_input_fn()` slouží pro konvertování dat do interní reprezentace sítě.

- **dataModel**

Zde se provádí předzpracování dat. Obsahuje hlavní funkci `createInput()`, která vytváří vstupní vektor pro síť. Z krátkodobého úložiště se pro každý parametr načte potřebné množství dat, vypočítá derivace nebo rozdíl a popřípadně se normalizuje. Dále jsou vytvořeny potřebné časové parametry z časové značky.

- **log**

Logovací knihovna, která obsahuje tři úrovně logování: info, warning a error. Logovací zprávy se zapisují na standardní výstup a současně do zadaného logovacího souboru. Zprávy mají formát: *datum_čas [úroveň logování] soubor.py/funkce: zpráva*. Místo vyvolání logovací zprávy je tedy jednoduše dohledatelné.

- **config**

config.py je konfigurační soubor obsahující všechny volitelné parametry: nastavení každé klasifikační neuronové sítě, adresy datových sad, adresu MQTT brokeru, port pro webovou stránku, adresu logovacího souboru, atd.

Zdrojem sensorických dat může být MQTT, kde se data posílají přímo z brány, nebo ZeroMQ, kde data projdou přes serverovou aplikaci. Výběr protokolu se nastavuje v *config.py* a není možné využívat oba protokoly současně.

6.4 Možná rozšíření

Jelikož je aplikace vytvořena ve skriptovacím jazyce Python, je možné ji rychle a jednoduše upravovat a rozšiřovat. Pokud bychom měli přístup do databáze systému k historickým datům, bylo by možné načíst tyto data už při startu aplikace a nebylo by nutné čekat, až se nasbírá požadované množství. Tím by aplikace mohla začít vyhodnocovat nově příchozí data hned po startu.

Pokud by bylo potřeba zpracovávat velké množství sensorů, je možné vyhodnocování provádět hromadně. To by byla výhodná optimalizace, pokud by klasifikátorů bylo více. Estimátor DNNClassifier tuto možnost vyhodnocování většího množství dat nabízí. Jelikož nejkratší interval mezi posílanými daty ze sensorů je 15 s, stačí provádět vyhodnocování ve stejném intervalu. Nově příchozí data by se předzpracovala dle datového modelu a zařadila do fronty pro vyhodnocení. Klasifikátory by se pak sekvenčně střídali. Možná komplikace by mohla nastat při zpětném přiřazení dat ke správným sensorům.

Jako další vylepšení může být přidání relevantních externích hodnot, např. teplota z nejbližší meteorologické stanice. Díky tomu by aplikace věděla, jaký je (v zimě) rozdíl teplot mezi domácností a venkem. Mohla by tím uživateli doporučit otevření okna na vhodnou dobu pro vyvětrání vydýchaného vzduchu s minimální ztrátou tepla.

Inteligentní systém může mít samozřejmě i další senzory měřící jiné veličiny, než jen teplotu a vlhkost. V případě, že bychom chtěli tyto veličiny také přidat do vyhodnocování, stačí je vhodně předzpracovat a přidat do vstupního vektoru jako další parametry. Současně s tím se také musí přetrénovat neuronová síť nebo vytvořit její nová instance.

K dalšímu zpřesnění klasifikace může pomoci sám uživatel. Ve výjimečných případech se systém může prostřednictvím uživatelského rozhraní uživatele dotázat, zda aktuálně vyhodnocená událost byla detekovaná správně. Tato informace se přidá do datové sady a neuronová síť se přetrénuje.

Kapitola 7

Výsledky

Tato kapitola shrnuje a zobrazuje výsledky navržených řešení předzpracování dat, pro dosažení co nejvyšší přesnosti klasifikace pro jednotlivé datové sady a přesnosti jednotlivých událostí. Výsledky jsou prezentovány v následujících tabulkách a poté zhodnoceny.

Testovány byly datové sady z koupelny, kuchyně a obývacího pokoje. Sada koupelny obsahuje události *sprcha* a *po sprše*, sada kuchyně obsahuje události *vaření* a *smažení/pečení*, sada obývacího pokoje obsahuje události *spánek*, *blízko teple/na slunci* a *otevření okna*.

7.1 Přesnost datových sad

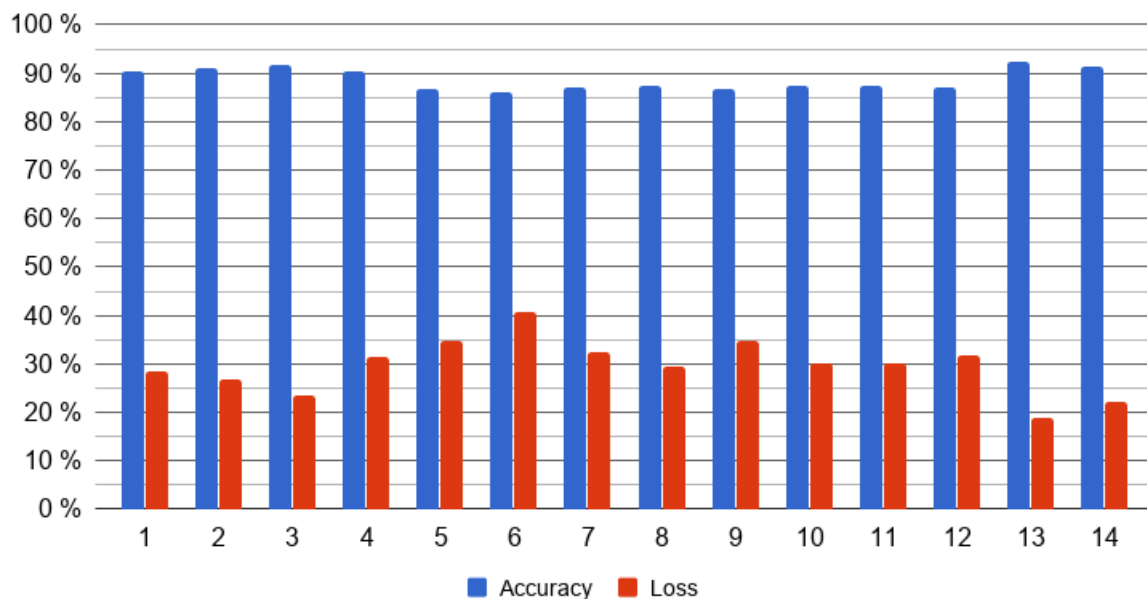
Načtená data z datových sad byla načtena a předzpracována do specifických vektorů sloužící jako vstupní vrstva neuronové sítě popsané v kapitole 5.6 na straně 31. Výsledkem trénování a testování vytvořené sítě je její přesnost a ztrátová funkce, přičemž cílem je získat co nejvyšší přesnost s minimální ztrátou. Tímto způsobem byl nalezen nejvhodnější vstupní vektor.

	bathroom		kitchen		livingroom	
	přesnost	ztráta	přesnost	ztráta	přesnost	ztráta
1.vektor	91.03	26.54	95.9	17.52	84.06	40.9
2.vektor	92.71	23.57	95.68	16.93	84.71	39.87
3.vektor	92.96	20.15	95.86	15.65	86.24	35.07
4.vektor	89.44	37.3	95.76	18.46	85.63	39.01
5.vektor	93.01	20.18	95.84	15.57	71.13	68.42
6.vektor	91.12	29.4	95.87	20.24	71.02	72.81
7.vektor	94.19	16.8	96.05	13.86	70.59	66.49
8.vektor	94.9	13.7	96.45	12.18	71.03	62.66
9.vektor	93.35	19.98	95.82	15.42	70.76	68.64
10.vektor	94.74	14.41	96.25	12.34	70.91	63.32
11.vektor	94.7	14.06	96.39	12.58	70.68	63.51
12.vektor	94.24	16.4	96.25	13.01	70.52	66.1
13.vektor	96.89	8.49	96.64	10.0	83.88	38.28
14.vektor	96.43	9.82	96.08	11.69	81.5	44.78

Tabulka 7.1: Procentuální přesnosti a ztráty datových sad s vytvořenými vstupními vektory.

Tabulka 7.1 zobrazuje, jakých přesností a ztrát (v procentech) bylo dosaženo s vytvořenými vstupními vektory pro každou datovou sadu. První hodnota značí přesnost sítě, tedy její schopnost správně klasifikovat vstupní data. Druhá hodnota je výsledek ztrátové funkce, tedy kolik procent průměrně chybělo ke stoprocentní jistotě klasifikované události.

Z výsledků vyplývá, že nejlepší klasifikace byla dosažena se vstupním vektorem obsahující parametr: rozdíl oproti průměru a derivace hodnot za posledních 10, 5, 2 a 1 minut pro teplotu i vlhkost. V tabulce tučně zvýrazněno. Dále se ukázalo, že z datové sady obývací místnosti se neuronová síť nedokáže spolehlivě naučit a klasifikovat, jelikož v žádném z případů přesnost nepřesáhla ani 90 %.



Obrázek 7.1: Průměrné přesnosti a ztráty všech datových sad s vytvořenými vstupními vektory.

V grafu na obrázku 7.1 jsou zobrazeny hodnoty přesností (modré) a ztrát (červené) zprůměrované ze všech datových sad. Je zde vidět, že 13. a 14. vstupní vektor dosahují nejlepších výsledků, avšak pro obývací pokoj vychází lépe 3. vstupní vektor.

7.2 Přesnost detekce událostí

Pro zjištění nedokonalostí datových sad byly jednotlivé události z datových sad vyfiltrovány a otestovány samostatně. Tím tedy bylo vytvořeno a natrénováno sedm neuronových sítí.

Tabulka 7.2 zobrazuje, jakých přesností a ztrát (v procentech) bylo dosaženo pro jednotlivé sledované události. Porovnání nejlepších dosažených výsledků detekce jednotlivých událostí je zobrazen v grafu na obrázku 7.2. Z výsledků je patrné, že detekce spánku je velmi špatně detekovatelná ve všech případech vstupních vektorů. To je také příčinou nedostatečné přesnosti a velmi vysoké ztrátovosti datové sady obývacího pokoje. Naopak jako velmi pozitivní se ukazují detekce událostí sprchy a otevření okna.

	shower		after shower		sleep		close to heat	
	přesnost	ztráta	přesnost	ztráta	přesnost	ztráta	přesnost	ztráta
1.vektor	97.76	8.76	92.95	19.94	87.09	29.72	97.28	9.55
2.vektor	98.63	5.82	93.83	18.52	88.38	26.54	97.33	9.26
3.vektor	98.48	5.66	94.04	17.99	89.32	23.88	97.72	7.63
4.vektor	96.63	13.46	92.99	23.55	88.98	24.28	97.25	8.56
5.vektor	98.41	6.2	94.18	15.35	74.22	54.62	97.31	10.5
6.vektor	96.6	13.81	92.82	23.66	73.91	57.46	97.28	12.47
7.vektor	98.72	4.5	95.24	12.01	74.07	52.88	97.27	9.93
8.vektor	98.85	3.93	95.33	11.1	74.82	50.9	97.39	8.71
9.vektor	98.52	5.88	94.86	13.73	74.5	53.9	97.22	10.77
10.vektor	98.97	3.65	95.56	10.56	74.91	50.38	97.4	8.91
11.vektor	98.8	4.02	95.64	10.76	74.45	51.31	97.36	9.03
12.vektor	98.76	4.5	95.01	12.71	74.05	51.98	97.47	9.62
13.vektor	99.01	2.98	97.35	6.62	86.51	31.34	98.1	5.68
14.vektor	98.93	3.18	97.19	7.47	85.58	32.91	97.64	7.1

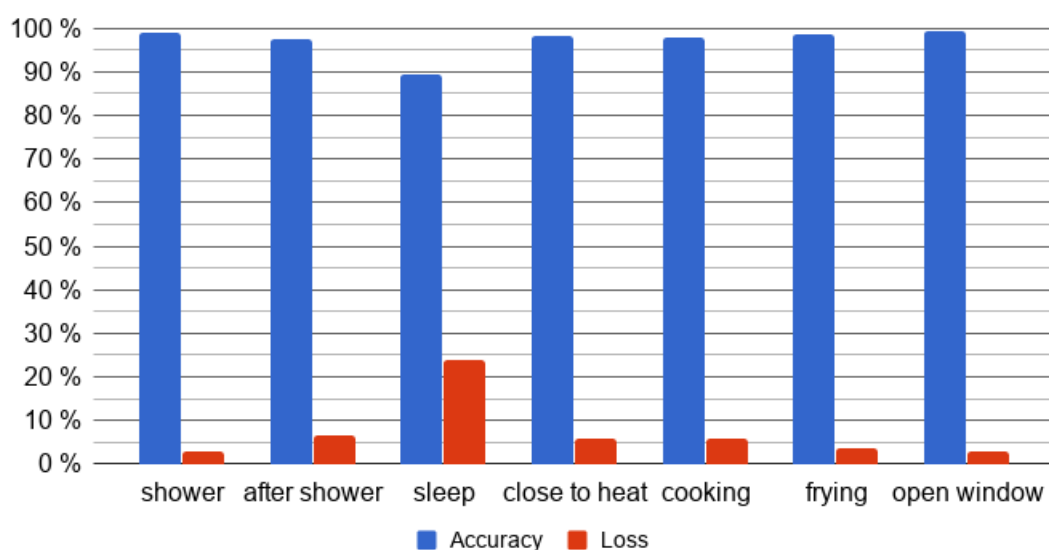
	cooking		frying		open window	
	přesnost	ztráta	přesnost	ztráta	přesnost	ztráta
1.vektor	97.38	10.84	98.57	6.46	99.03	4.77
2.vektor	97.23	10.46	98.4	6.61	98.94	6.61
3.vektor	97.47	9.31	98.64	5.92	99.11	4.11
4.vektor	97.29	11.0	98.45	6.85	98.94	5.31
5.vektor	97.37	9.47	98.54	5.93	99.03	4.68
6.vektor	97.3	12.47	98.39	8.26	99.07	4.81
7.vektor	97.53	8.21	98.53	5.43	99.16	3.76
8.vektor	97.82	7.51	98.56	4.86	99.22	2.91
9.vektor	97.65	8.7	98.48	6.01	98.98	4.87
10.vektor	97.88	7.21	98.59	4.87	99.15	3.24
11.vektor	97.62	7.79	98.5	5.0	99.1	3.51
12.vektor	97.56	8.0	98.49	5.49	99.13	3.38
13.vektor	97.98	5.89	98.63	3.75	99.25	2.81
14.vektor	97.88	6.33	98.48	4.56	99.29	2.86

Tabulka 7.2: Procentuální přesnosti a ztráty jednotlivých událostí se vstupními vektory.

7.3 Zhodnocení

Z navržených parametrů, popsaných v sekci 5.5 Úprava vstupních dat, se parametry vytvořené z časové značky ukázaly jako nevhodné. Parametr hodiny je užitečný pouze u detekce spánku, jelikož datové sady obsahují spánkové intervaly pouze přes noc. Parametr minuty má převážně rušivý efekt a parametr měsíce naopak žádný efekt – to mohl být způsobeno nedostatečně obsáhlou datovou sadou.

Také předpoklad zlepšení výsledků normalizací hodnot nebyl úspěšný. To mohlo být způsobeno nevhodným zvolením maximálních a minimálních hodnot. Problém je určit tyto

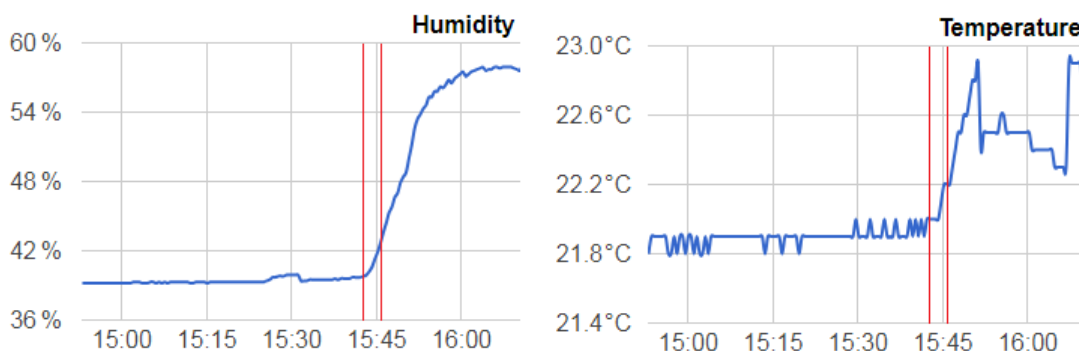


Obrázek 7.2: Nejlepší dosažené výsledky detekce jednotlivých událostí.

hranice správně. V případě velkého rozsahu se po normalizaci mohou ztratit malé změny hodnot a v případě malého rozsahu mohou tyto hraniční hodnoty překročit.

Jako velmi významný parametr ovlivňující správnost klasifikace se ukázal rozdíl aktuální hodnoty s průměrem. To je způsobeno tím, že průměrná hodnota je specifická pro měřenou oblast a tím se dosahuje vyšší přesnosti klasifikace.

Z tabulky 7.2 popisující přesnosti detekce jednotlivých událostí vyplývá, že vytvořením samostatných neuronových sítí pro každou jednu událost dosáhneme mnohem lepších výsledků. Stále je však nutné znát místnost, ze které data přichází.



Obrázek 7.3: Zpoždění detekce události v případě sprchování.

Reálným nasazením a testováním aplikace WIGO bylo zjištěno, že kvůli zpoždění měření senzorů a zpožděním samotné klasifikace se detekce některých rychlých událostí (sprcha, otevření okna) projevila až po 3 minutách. Příklad takového zpoždění je znázorněn v grafech na obrázku 7.3, kde první červená čára značí začátek události (sprchování) a druhá čára detekování události. Neuronová síť se pravděpodobně naučila ignorovat šum senzorů, čímž z počátku ignoruje začátek i známých událostí.

Dále bylo zjištěno, že když se v kuchyni smaží a vaří současně, klasifikátor detekuje pouze událost vaření. To je pravděpodobně způsobeno tím, že smažení je specifické růstem teploty a poklesem vlhkosti, avšak vaření růstem obou typů hodnot.

V případě měření v malém bytě se projevilo ovlivňování vzduchu mezi místnostmi. Pokud se uživatel osprchoval a následně nechal otevřené dveře koupelny, vlhkost a teplo se rozptýlily i do ostatních místností. To výjimečně krátkodobě vyvolalo detekování chybné události.

Aplikace WIGO se třemi načtenými neuronovými sítěmi přijímající data deseti senzorů spotřebuje cca 200 MB paměti RAM. Každá uložená neuronová síť na disku zabírá cca 80 MB.

Kapitola 8

Závěr

V této práci byl představen obecný inteligentní systém a detailně popsán domácí inteligentní systém *BeeeOn*, jeho dostupné senzorické prvky a další možnosti systému. Dále byly popsány fyzikální veličiny související s domácím prostředím, jejich vlastnosti, chování a možná využití k vytěžení nových informací užitečných pro komfort člověka. Pro sběr senzorických dat byl využit zmíněný inteligentní systém *BeeeOn*, který byl umístěn ve čtyřech různých lokalitách – laboratoř, studentský a rodinný byt a rodinný dům. Pro každé sběrné místo byla sepsána charakteristika tohoto místa a rozmístění použitých senzorů.

Senzorická data byla manuálně analyzována a výsledky analýzy jsou popsány a shrnuty v kapitole 4. Na základě analýzy byly odvozeny události, které jsou výsledkem klasifikace. Události jsou klasifikovány z minima informací z domácnosti, konkrétně pouze z teploty a vlhkosti v místnosti, což může být zajímavé pro koncového uživatele z finančního pohledu. Nalezené události, které mají charakteristický průběh, jsou např. sprchování, otevření okna, spánek, atd. Na základě těchto zjištění byly vytvořeny ručně anotované datové sady.

Vytvořený klasifikátor, pojmenovaný *WIGO*, si vytvoří potřebné instance hlubokých neuronových sítí a natrénuje je daty z datových sad, nebo si může načíst již natrénované síť. Cílem klasifikátoru je detekovat události v domácnosti z aktuálních dat. Data ze senzorů jsou do klasifikátoru zasílaná pomocí protokolu MQTT nebo ZeroMQ. Před klasifikací se data předzpracují do podoby vstupního vektoru popsaného v návrhu v kapitole 5. Následuje vyhodnocení a kontrola zvýšených hladin hodnot. Výsledkem je detekovaná událost, její pravděpodobnost a informace, zda jsou měřené hodnoty ve zdravotně nezávadných normách.

Detekované události mohou být použity jako vstupy pro navazující systém automatizovaného řízení domácnosti. V důsledku toto může celý systém přispět ke zlepšení životních podmínek a ve výsledku vést ke zlepšení životních podmínek. Například v případě sprchování mohou být překročeny nežádoucí úrovně měřených hodnot. Reakcí na tuto událost může být např. spuštění odvětrávání, popř. nemusí k žádné reakci dojít. V ostatních případech je potřeba uživatele upozornit na závažnou nebo nežádoucí situaci.

Z přesností klasifikace se jako nejlépe detekovatelné události jeví sprcha a otevření okna. I přesto, že *BeeeOn* senzory generují lehce zašumněná data, je klasifikátor stále schopný správné detekce i bez nutnosti vyhlazení těchto dat. Z výsledků také vyplývá, že vytvořením neuronových sítí pro jednotlivé události vede k přesnějším výsledkům. K dalšímu zpřesnění klasifikace může také pomoci interakce s uživatelem dotazování na správnost vyhodnocení.

Pokračováním této práce bude zaměření se na zlepšení zdravotních podmínek v kancelářích a firemním prostředí.

Literatura

- [1] Bencko, V.: *Hygiena: Učební texty k seminářům a praktickým cvičením. 2. přepracované a doplněné vydání vydání. s. 126-128.* Praha: Karolinum, 2002, ISBN 80-7184-551-5.
- [2] Czechklima.cz: *Teploty vhodné pro bydlení.* [Online; 4.1.2018].
URL <https://www.czechklima.cz/novinky/teploty-vhodne-pro-bydleni>
- [3] Eon.cz: *Jaké jsou optimální teploty v místnostech?* [Online; 28.12.2017].
URL <https://www.eon.cz/radce/jake-jsou-optimalni-teploty-v-mistnostech>
- [4] Fioletov V., Kerr J., Fergusson A.: *The UV Index: Definition, Distribution and Factors Affecting It.* Canadian Journal of Public Health, 101 (4): 15-19, 2010.
URL <http://journal.cpha.ca/index.php/cjph/article/viewFile/1905/2203>
- [5] František Vítězslav Zbořil, D. I. C.: *Biologický a umělý neuron, umělé neuronové sítě.*, studijní materiály k semináři *Soft Computing*. Fakulta informačních technologií VUT v Brně, 2017.
- [6] Fryč, A.: *Hygienické limity hluku.* [Online; 9.1.2014].
URL <https://www.akusticka-pena.cz/novinky/hygienicke-limity-hluku/>
- [7] Google Brain Team: *About Tensorflow.* [Online; 27.2.2018].
URL <https://www.tensorflow.org/about/>
- [8] Görner, M.: *Tensorflow and deep learning - without a PhD*, Devovx Belgium – The Developers Community Conference. [Online; 8.11.2016].
URL <https://www.youtube.com/watch?v=vq2nnJ4g6N0>
- [9] Hanneman K.K., Cooper K.D., Baron E.D.: *Ultraviolet immunosuppression: mechanisms and consequences.* Dermatologic Clinics, 24 (1): 19–25., 2006.
URL <https://www.sciencedirect.com/science/article/pii/S0733863505001014>
- [10] Heaton J., Ph.D.: *The Number of Hidden Layers* 01.06.2017. Heaton Research, 2017, [Online].
URL <https://www.heatonresearch.com/2017/06/01/hidden-layers.html>
- [11] Hollan, J.: *Jak osvětlovat budovy veřejných institucí šetrně k životnímu prostředí,* Hvězdárna a planetárium M. Koperníka v Brně 2007. [Online].
URL http://amper.ped.muni.cz/light/texty_html/osv_kanc.html
- [12] Inoutic.cz: *Větrání - Vlhkost vzduchu a tvorba plisní.* [Online; 14.1.2018].
URL <http://www.inoutic.cz/tipy-fakta/vlhkost-vzduchu-tvorba-plisni>

- [13] Jan Pešta, I.; Viktor Zwiener, I. P.: *Hlukové limity, měření hluku a hlukové studie*, 2012. [Online].
URL <https://atelier-dek.cz/hlukove-limity-mereni-hluku-hlukove-studie-194>
- [14] Ladislav Monzer, I. a. C.: *Umělé osvětlení v obytných prostorech, 2. část – Volba intenzity osvětlení*, Časopis Světlo 03/2002. [Online].
URL <http://www.odbornecasopisy.cz/svetlo/casopis/tema/umele-osvetleni-v-obytnych-prostorech-2-cast-volba-intenzity-osvetleni--16854>
- [15] Lunos.cz: *Obecně o bytovém větrání*. [Online; navštíveno 04.12.2017].
URL <http://www.lunos.cz/o-systemu/>
- [16] Novotný, I. P.: *Vliv oxidu uhličitého (CO₂) na kvalitu vnitřního prostředí*, 30.5.2017. [Online].
URL <http://vetrani.tzb-info.cz/vetrani-skol/15831-vliv-oxidu-uhliciteho-co2-na-kvalitu-vnitrnihoprostredi>
- [17] Osvald Modrlák, D. I. C.: *Teorie automatického řízení II., FUZZY ŘÍZENÍ A REGULACE*. Fakulta mechatroniky a mezioborových inženýrských studií, [Online; 19.4.2002].
URL <https://www.kirp.chnf.stuba.sk/~bakosova/wwwRTP/tar2fuz.pdf>
- [18] Petr Mejzlik, D. E.: *Zdraví a lidské tělo - Vnímání hluku a vibrací*. [Online; 13.1.2018].
URL <http://www.x-idea.cz/vnimani-hluku-a-vibraci/>
- [19] Pocasicz.cz: *Jaký vliv má tlak vzduchu na vývoj počasí*, 29.06.2015. [Online].
URL <http://www.pocasicz.cz/aktuality-o-pocasi/aktuality-471/jaky-vliv-ma-tlak-vzduchu-na-vyvoj-pocasi-463>
- [20] Sterling, E.; Arundel, A.; T.D. Sterling, P.: *Criteria for human Exposure to Humidity in Occupied Buildings*, ASHRAE Transactions, Vol.91, Part 1, 1985: str. 611-622. [Online].
URL <http://www.pro.net/sterlingiaq.com/html/photos/1044922973.pdf>
- [21] TZBinfo.cz: *Vyhláška č. 194/2007 Sb. kterou se stanoví pravidla pro vytápění a dodávku teplé vody*, 2007. [Online].
URL <http://www.tzb-info.cz/pravni-predpisy/vyhlaska-c-194-2007-sb-ktouhou-se-stanovi-pravidla-pro-vytapeni-a-dodavku-teple-vody>
- [22] TZBinfo.cz: *Zákon č. 406/2000 Sb. o hospodaření energií*, 2017. [Online].
URL <http://www.tzb-info.cz/pravni-predpisy/zakon-c-406-2000-sb-o-hospodareni-energi>
- [23] Tůmová, M.: *Jaká je ideální teplota domácnosti?* [Online; 27.10.2012].
URL <http://www.uspornadomacnost.cz/vytapeni/jaka-je-idealni-teplota-domacnosti.html>
- [24] Uldrich, M.; Jurczyk, T.: *Neuronové sítě a jejich využití*. [Online; 12.3.2014].
URL <https://www.systemonline.cz/clanky/neuronove-site-a-jejich-vyuziti-1.htm>

- [25] Unruh, A.: *What is the TensorFlow machine intelligence platform?*, opensource.com, 9.11.2017. [Online].
URL <https://opensource.com/article/17/11/intro-tensorflow>
- [26] Zendulka Jaroslav, D. I. C.: *Získávání znalostí z netradičních typů dat*, studijní materiály k semináři *Získávání znalostí z databází*. Fakulta informačních technologií VUT v Brně, 2017, str.36-39.
- [27] Zikán, I. Z.: *Oxid uhličitý - utajený nepřítel*, 7.2.2011. [Online].
URL <http://vetrani.tzb-info.cz/vnitri-prostredi/7132-oxid-uhlicity-utajeny-nepritel>
- [28] ŠetřímEnergii.cz: *Udržujte v interiéru správnou vlhkost vzduchu*. [Online; 4.1.2018].
URL <http://www.setrimenergii.cz/mikroklima/spravna-vlhkost-vzduchu-v-interieru>
- [29] Žák, M.: *Tlak vzduchu výrazně klesá – je to zátěž pro lidi?* [Online; 29.1.2015].
URL <http://www.in-pocasi.cz/clanky/teorie/tlak-vzduchu-pokles-29.1.2014/>

Příloha A

Obsah CD

- Tato práce v PDF i v TEX.
- Výsledná aplikace WIGO pro klasifikaci událostí v domácnosti.
- Naměřená data, datové sady, poznámky a pomocné skripty.