Středoškolská technika 2018

Setkání a prezentace prací středoškolských studentů na ČVUT

Machine learning pro predikci teploty

Petr Šejvl

SPŠ a VOŠ Písek, Karla Čapka 402, 397 11 Písek

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Písku dne 27. 03. 2018 ……

Petr Šejvl

Anotace

Projektem je webová aplikace, která průběžně sbírá data o okamžité teplotě uvnitř domu a o venkovním počasí (teplota, oblačnost atd.) a ukládá je do MySQL databáze. Z dat se pak naučí pomocí machine learningu jak venkovní počasí ovlivňuje teplotu vevnitř. Díky těmto znalostem a předpovědi počasí na 120 hodin dopředu pak sám předpovídá teplotu uvnitř domu na 120 hodin dopředu.

Klíčová slova

Machine learning; normal equation; regrese; predikce

Annotation

The project is a web application which continuously records real-time data about temperature inside a house and the outside weather (temperature, clouds, etc.) and saves it to a MySQL database. With the usage of machine learning the application learns the influence of outside weather on the inside temperature. With such knowledge and using a weather forecast for 120 hours the application is able to predict the temperature inside the house up to 120 hours.

Keywords

Machine learning; normal equation; regression; prediction

Obsah

1 Úvod

Motivací k vytvoření tohoto projektu bylo riziko zamrznutí podlahového topení a následné prasknutí potrubí vytápěcího systému. Dům je vytápěn krbem, automatické zapnutí topení tedy není možné. Protože nelze vytvořit program, který by riziku bránil aktivně, zvolil jsem tedy alespoň pasivní způsob. Pomocí Arduina a modulu SIM900 budu sbírat data o vnitřní teplotě. Údaje o venkovní teplotě a dalším počasí pak získám pomocí API ze serveru https://openweathermap.org/. Díky tomu, že tento server nabízí předpověď počasí, mohu i já předpovídat teplotu uvnitř domu. Zjistím, jak data získaná z OpenWeatherMap (OWM) ovlivňují vnitřní teplotu. Pak získám předpověď počasí z OWM a budu moci předpovídat i teplotu uvnitř domu. Díky takovým předpovědím pak budu vědět, kdy je v domě třeba zatopit, aby se předešlo zamrznutí vody v podlahovém topení.

2 TEORETICKÝ ÚVOD

2.1 Machine learning

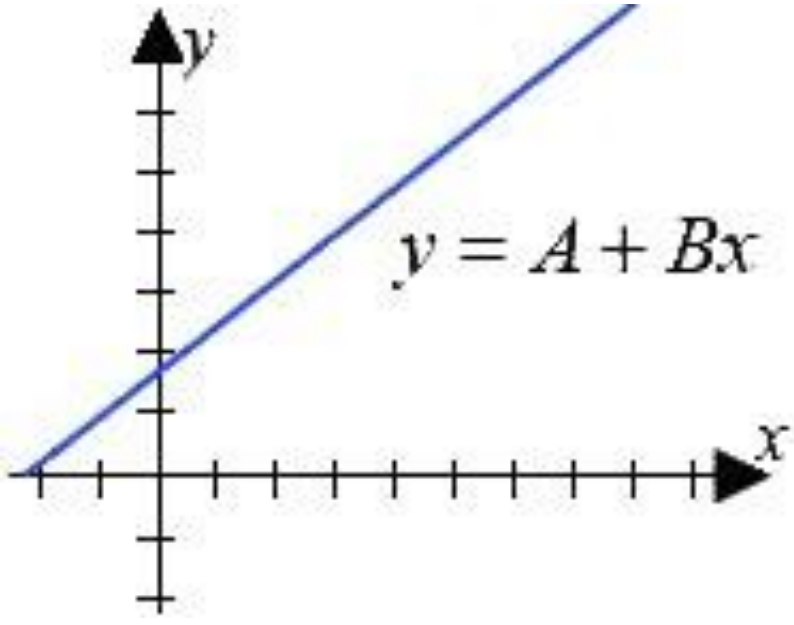
Machine learning (ML) je druh umělé inteligence (AI) se schopností se automaticky učit ze získaných dat, aniž by byla programově upravována. ML bere data a učí se z nich. Cílem je umožnit AI se učit bez zásahu programátora.

Rozděluje se na supervised (s učitelem) a unsupervised (bez učitele)

- Supervised ML ví, k jakému výsledku by měl dojít. Takový model dostává data, která jsou rozdělena na "target" (cíl, požadovaný výsledek) a "features" (vlastnosti, kterému k výsledku povedou). Model se tak naučí, jak jednotlivé features ovlivní konečný výsledek.
- Unsupervised ML nezná žádné výsledky. Dostává pouze features, nejčastěji je pak na základně jejich podobnosti rozděluje do skupin.

2.2 Lineární regrese:

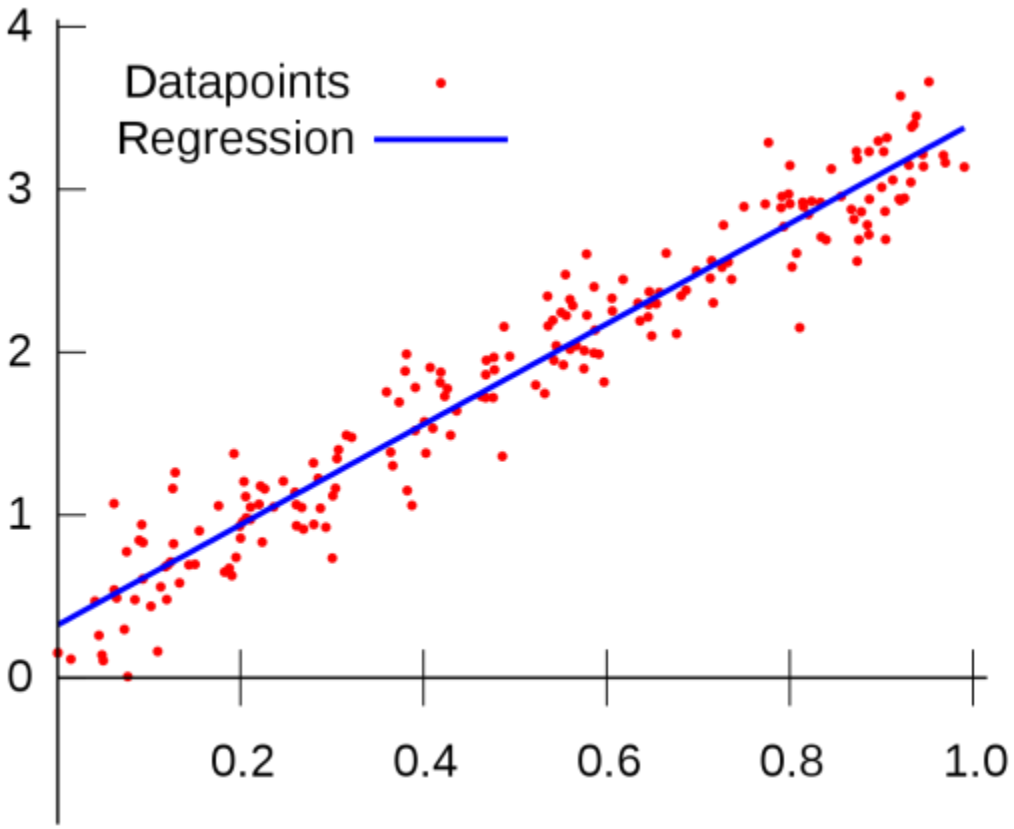
Jedná se asi o nejjednodušší ML model a tedy nejlepší ukázku v praxi. Využívá učení s učitelem. Lineární regrese se dá reprezentovat jako jednoduchá funkce y = a + b*x. Taková funkce pak lze zobrazit grafem:



Graf 1- jednoduchá lineární funkce

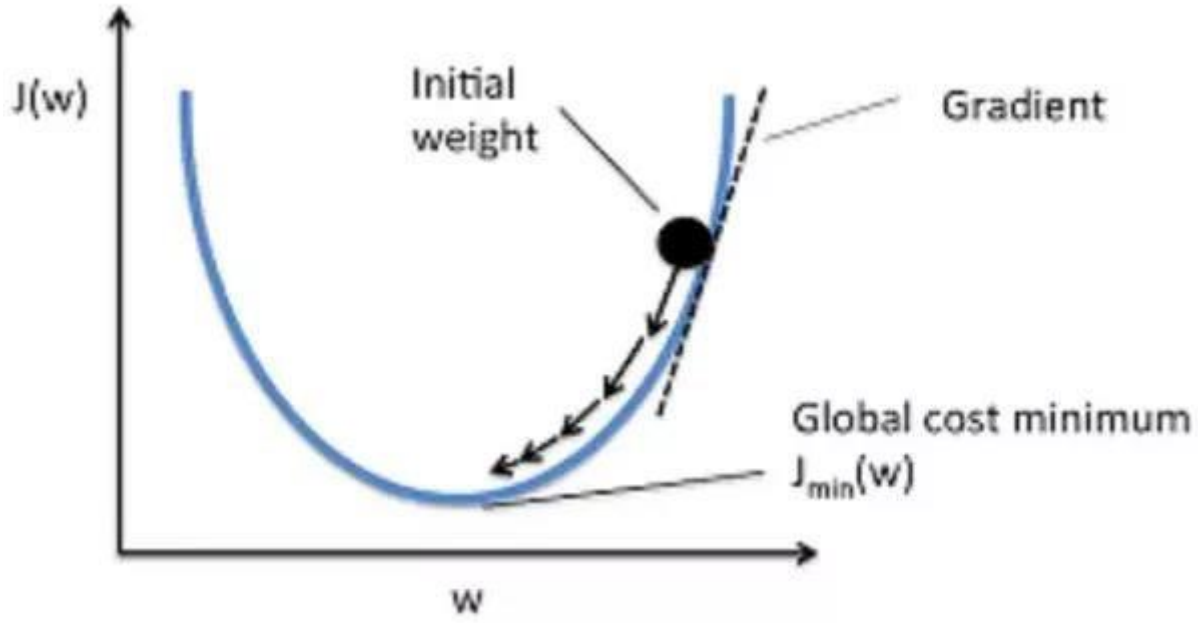
Model pak dostává data a upravuje svoje parametry ("a" a "b") tak, aby se co nejvíce přiblížilo skutečné cílové hodnotě (targetu). Pro zhodnocení přesnosti modelu se pak využívá "mean squeared error", což je průměr ze součtů všech rozdílů mezi námi vypočteným a

zadaným “targetem” na druhou. Zde je pak ukázka lineární regrese, kde body jsou jednotlivá měření a přímka představuje nejlepší predikce výsledků vzhledem k dodaným měřením



Graf 2 - jednoduchá lineární regrese

Nejčastěji využívaný způsob, jak se tyto parametr naučit je algoritmus nazývaný Gradient Descent. Ten upravuje parametry funkce tak, aby minimalizoval výše popsanou mean squared error. Úpravy probíhají pomocí derivací chybové funkce vzhledem k parametrům. Upravuje je tak dlouho, dokud se dostatečně nepřiblíží celkovému minimu této funkce. Ukázka postupného upravování na grafu:



Graf 3 - chybová funkce

(na ose y je hodnota chyby, na ose x hodnota parametru, který je upravován)

Nevýhodou toho algoritmu je pak fakt, že potřebuje více kroků k naučení parametrů a výpočet může být tedy poměrně zdlouhavý.

Další metodou je "Normal equation". Ta minimalizuje chybu v jednom kroku. Hodnoty parametrů se vypočítávají pomoci lineární algebry.

Srovnání algoritmů:

Gradient descent je lepší v případě velkého množství dat a features, normal equation je lepší pro menší množství dat.

V mém projektu proto pracuji s normal equation.

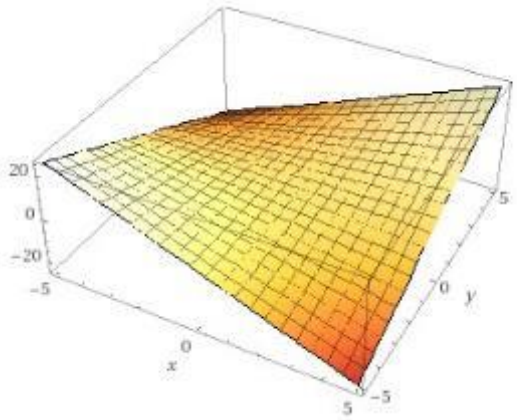
2.3 Multivariable regression

Model využívá více než jedné proměnné. Funkce pak může vypadat nějak takto: $a + b_1*x_1 + b_2*x_2 + b_3*x_3 + ...+ b_n*x_n$. Graf takové funkce je pak vícedimenzionální, konkrétně má $n + 1$ dimenzí. Model se jednotlivé parametry učí stejným způsobem jako u jednoduché lineární regrese. Díky více parametrům rozhodně můžeme získávat přesnější výsledky, ovšem nese to s sebou i riziko přesného opaku. Pokud bude náš model využívat velké množství features (proměnných), ale dataset, z kterého se bude učit, bude malý, může lehce nastat "overfitting". To znamená, že se model naučí parametry špatně a nebude adekvátně reagovat na nová data. Abychom se takového problému vyvarovali, doporučuje se asi 30 % z datasetu použít pouze na testování modelu. Testování funguje tak, že se tato testovací data nevyužívají k učení, ale

pouze se na nich poté vyzkouší už naučený model a sleduje se mean squared error. U testování nás pak zajímá rozdíl mezi mean squarred error pro vzorky, na kterých se model učil a pro vzorky, které předtím neviděl (testovací)

Ukázka, jak by mohl vypadat graf multivariable regression:

$y = a + b_1*x_1 + b_2*x_2$

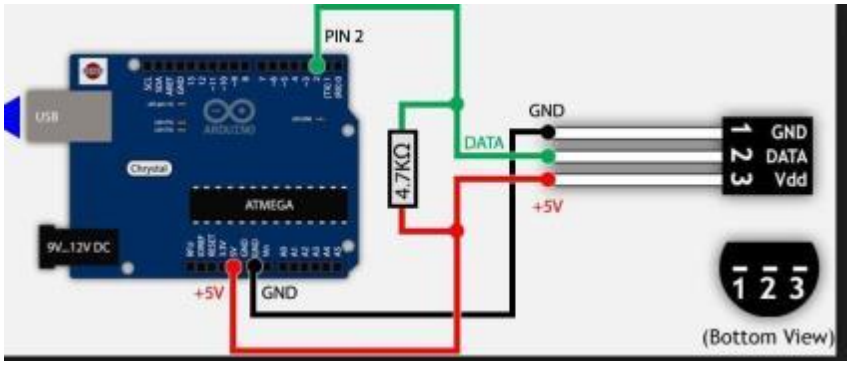


Graf 4 - graf funkce s více proměnnými

3 ARDUINO, SIM900, 18B2

Obrázek 1 - hardwarové díly

- Arduino
  - Malý jednodeskový počítač založený na mikroklontrolerech ATmega
  - Dají se s ním ovládat I/O periferie
- Sim900
  - GSM modul
  - Umožňuje arduinu komunikovat přes mobilní sítě, posílat sms, zprostředkovávat telefonní hovory a přístup na internet přes GPRS
  - Ovládá se pomocí AT příkazů, které posíláme přes sériovou linku
- 18B2
  - Teploměr
  - Připojuje se na zdroj napájení 5V na arduinu, na zem a na jeden z digitálních pinů



Obrázek 2 - zapojení teploměru

3.1 Zapojení

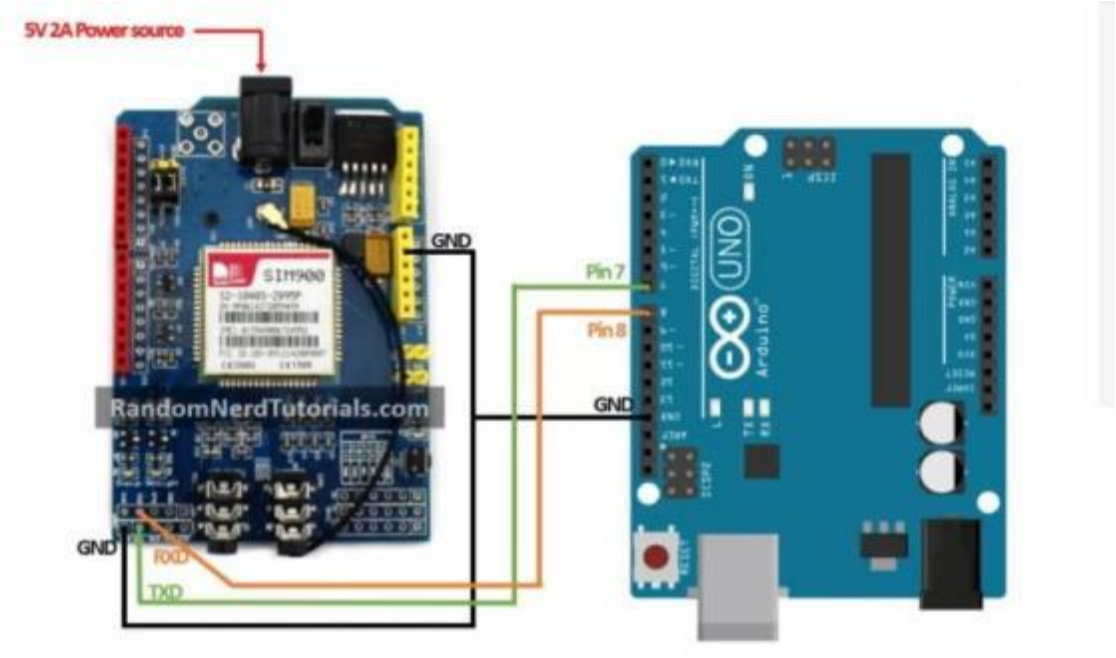
Na arduino vytvoříme Software Serial na pinech 7, 8 a spojíme je se SIM900

7 => TXD

8 => RXD

GND => GND

Viz obrázek:



Obrázek 3 - zapojení SIM900

Teploměr připojíme také podle obrázku výše, napájení a data budou paralelně spojeny přes 4,7 ohmový rezistor, ke čtení dat z teploměru využijeme knihovny OneWire a DallasTemperature.

3.2 Popis fungování

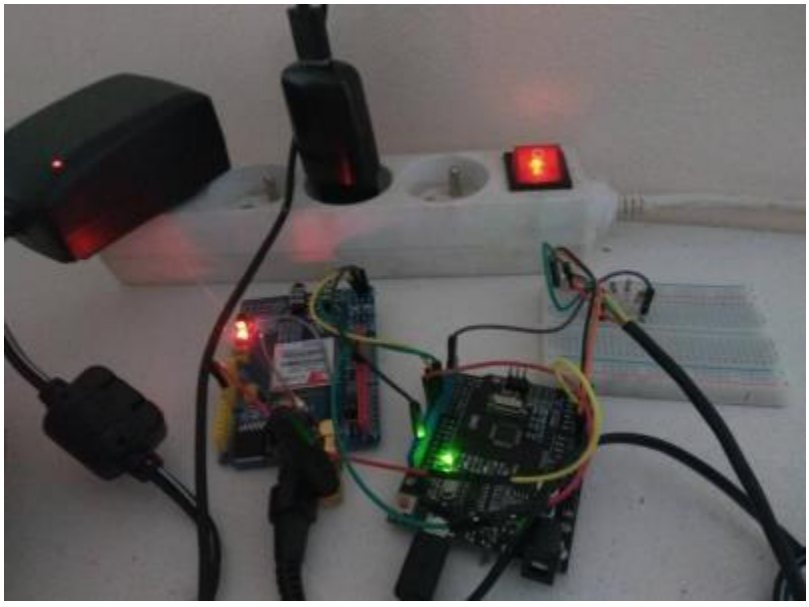
Arduino odesílá přes sériovou linku AT příkazy na SIM900, díky kterým SIM900 odešle GET request na server, kde v url uvádí dvě proměnné. Jedná se o naměřenou teplotu, kterou server zpracuje, a o API klíč, který potvrdí, že ten kdo data posílá na to má skutečně právo.

Protože SIM900 občas vypadne z telefonní sítě, musíme před pokusem o odeslání dat zkontrolovat, je-li připojen a popř. ho znovu připojit. Jelikož na inicializování sítě je potřeba nějaký čas, o který by se pak opožďovala všechna měření, musíme si ho počítat. Následující odesílání data pak proběhne o to dřív, aby se zpoždění nesčítala.

Abych mohl zapínat SIM900 softwarově, bylo třeba spájet takto dva piny:

Obrázek 4 - pájení

Celkové zapojení přímo na pracovišti:



Obrázek 5 - zapojení na pracovišti

3.3 Server-side

Kód na straně serveru, který data přijímá, nejprve zkontroluje API klíč, aby nedošlo k neautorizovanému přístupu. Poté zkontroluje, zda skutečně přišla i teplota a připojí se k databázi. Nyní se pokusí načíst API, kvůli možné chybovosti se načtení musí zkontrolovat a popřípadě znovu zavolat tato funkce od začátku. Pokud vše proběhne v pořádku, funkce

projde API, získá z něj námi požadované údaje a spolu s obdrženou vnitřní teplotou a aktuálním časem je uloží do databáze.

4 HLAVNÍ MYŠLENKA A POPIS ALGORITMU

Pro zjednodušení popisu práce algoritmu bych rád vysvětlil některé pojmy, které budu používat:

“Skupina měření” = časově uspořádaná měření (od prvního k poslednímu) jednoho chladnutí. Začátkem skupiny měření je tedy první záznam, který není algoritmem vyhodnocen, jakože by byl uložen ve chvíli, kdy se v domě topilo. Koncem skupiny měření je pak záznam před následujícím zatopením, teda záznam před záznam, který je algoritmem vyhodnocen, že byl uložen ve chvíli, kdy se v domě topilo. Ve zkratce se jedná o časově uspořádané záznamy mezi jednotlivými topeními (tj. záznamy chladnutí).

“Vzorky pro model” = data vybraná tak, že z nich lze model učit. Vzorky pro model budeme vybírat tak, že v každé skupině měření najdeme vždy jednu teplotu blízkou aktuální teplotě. Následující měření v každé skupině měření pak popisuje, jaká byla situaci za 3 hodiny. Vezmeme tedy všechny tyto záznamy (ty, co popisují, jaká byla situace za 3 hodiny) a předáme je modelu, aby se z nich naučil parametry. Tím jsme získali vzorky pro model na predikci 3 hodiny dopředu. Vzorky pro model pro následující dobu (za 6 hodin, 9 hodin atd.) pak vytvoříme stejným způsobem.

Cílem tohoto projektu je s maximální přesností předpovídat jaká bude teplota uvnitř domu v následujícím čase. Protože OWM (Open Weather Map), jejichž API využívám k získání údajů o venkovní situaci, nabízí předpověď počasí po 3 hodinách na 120 hodin dopředu, tak i můj projekt předpovídá vnitřní teploty vždy po 3 hodinách maximálně 120 hodin dopředu.

Ve chvíli, kdy chceme znát predikci vnitřní teploty, máme k dispozici: aktuální vnitřní teplotu, poslední dobu, jak dlouho byl dům vytápěn, předpověď počasí, aktuální situaci venku a záznamy předchozího průběhu vnitřní teploty s tehdejším venkovním počasím.

První co musíme udělat, je seskupit předchozí záznamy, aby šly využít k naučení ML algoritmu. Projdeme tedy záznamy z databáze a rozdělíme je podle výše vysvětleného systému do skupin měření. Záznam pořízený ve chvíli, kdy se v domě zatopilo, pozná algoritmus díky okamžitému velkému nárůstu vnitřní teploty. Takový záznam pak patřičně označí. Algoritmus musí dále zjistit, jak dlouho se topilo. To zjistí tak, že bude procházet další záznamy až do doby, kdy vnitřní teplota klesne pod 15 stupňů C. Všem záznamům, které takto prošel, přiřadí označení topení. Ve chvíli, kdy teplota klesne pod 15 °C, se “vrátí zpět” a všem záznamům od tohoto poklesu do poslední nejvyšší teploty (bez) odebere označení topení. Dobou topení pro jednotlivé skupiny měření je tedy počet přímo předcházejících měření, označených jako topení.

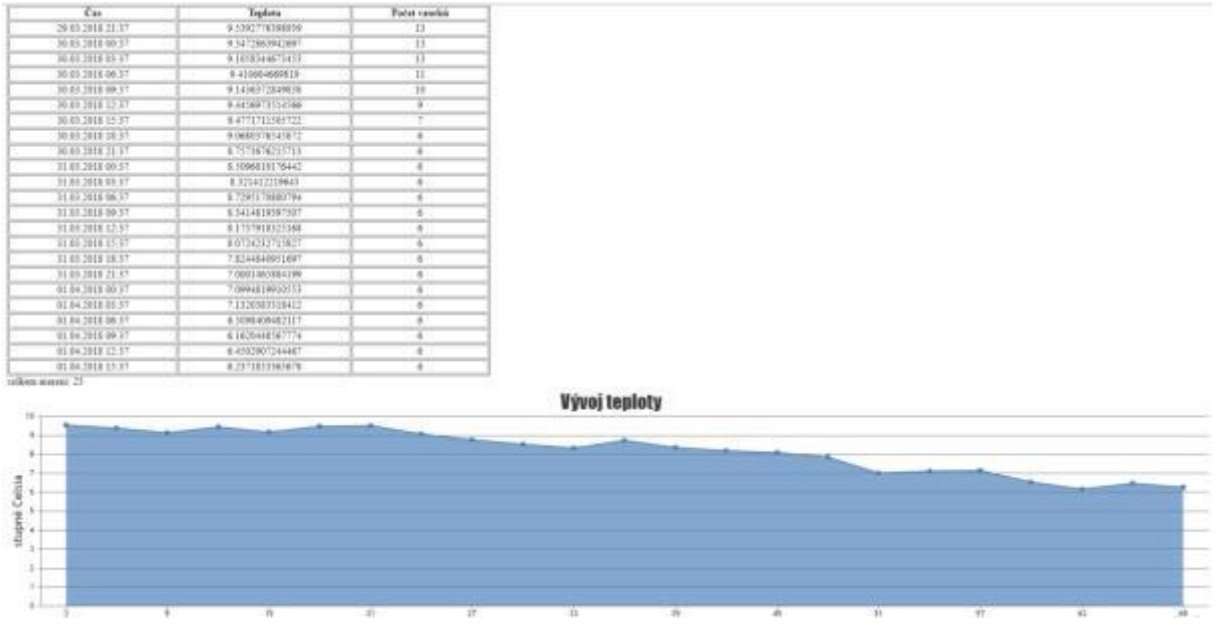
Ve chvíli, kdy máme takto zpracované záznamy, získáme z nich vzorky pro model. Vzorky pro model rozdělíme na features (konstanta 1, venkovní teplota, doba vytápění před chladnutím) a na targets (vnitřní teplota). Využíváme výše popsaného modelu multivariable regression, který se díky těmto vzorkům naučí parametry, jak je výše popsáno.

Vždy, když získáme parametry na určitou situaci (za 3, 6, 9 hodin atd...), vynásobíme je s údaji z předpovědi (získanými z API) a sečteme je. A tak je hotová první predikce teploty.

Takto budeme uskutečňovat predikce, dokud to půjde. To znamená maximálně 120 hodin dopředu (tj. 40 predikcí), nebo dokud bude mít situace více než 2 vzorky pro model (2 by zničily kvůli extrémní nespolehlivosti strukturu grafu, dokud jich není více jak 10, nedá se na predikci příliš spoléhat, ale alespoň nezničí graf).

Předpovězené hodnoty tedy zapíšeme do tabulky a zobrazíme v grafu. Predikce je tímto hotova.

Zobrazujeme ji na jednoduchém front-endu na webu (pozn. datumy neodpovídají skutečnosti):

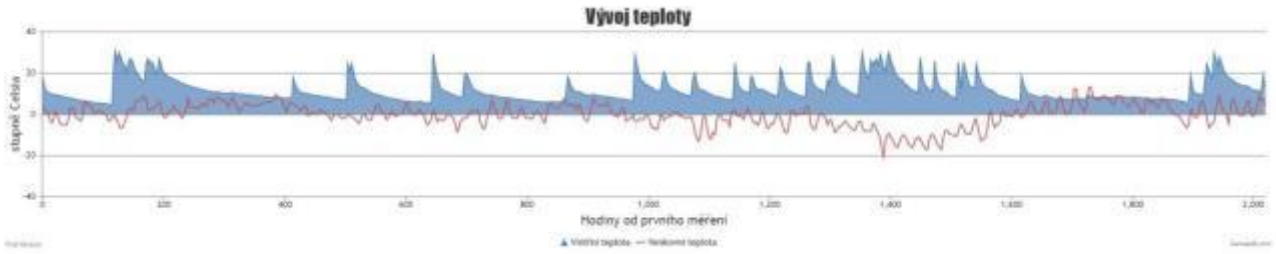


Obrázek 6 - ukázka predikce

5 ZÁVĚR

Díky ukládání dat a jejich zpracování můžeme předpovídat situaci uvnitř domu. Projekt stále sbírá data pro ML model, takže se predikce budou stále zpřesňovat. Díky predikcím tedy víme, za jak dlouho bude třeba v domě zatopit a můžeme sledovat, jak se situace uvnitř obvykle vyvíjí. Webové rozhraní pak nabízí jednoduchý a dostupný způsob prezentace výsledků.

Ukázka celkového vývoje teploty:



Obrázek 7 - ukázka celkového vývoje

Vše se nachází na webových stránkách: http://forecast.ondrej-sejvl.cz/

6 POUŽITÁ LITERATURA

1. tutoriál pro práci se SIM900 a schémata zapojení, dostupné z: https://randomnerdtutorials.com/sim900-gsm-gprs-shield-arduino/
2. machine learning z Coursera.org, zdroj pro teoretický úvod dostupný z: https://www.coursera.org/learn/machine-learning/home/welcome
3. grafy na webu, dostupné z: https://canvasjs.com/

7 SEZNAM OBRÁZKŮ A GRAFŮ

Grafy:

Obrázky:

8 PŘÍLOHY

1. Forecast.php
   - Digitální příloha
   - Kód. který pracovává data a vytváří predikci. Tu zapíše do tabulky a zobrazí na grafu
2. Arduino kód
   - Digitální příloha
   - Kód, který je nahrán na Arduinu. Měří teplotu a odesílá ji na server
3. Ukázka prvních 4 stran z tabulky, ve které jsou záznamy rozděleny do skupin měření a je vypočteno, jak dlouho se topilo