

# Light Learning Estimator Challenge

Ontwikkel een light-weight lokale voorspeller die op basis van slimme meter gegevens een lokale voorspelling deelt.



Title	Light Learning Estimator (LLE) Challenge	
	Een beschrijving van een 'challenge' om te komen tot een continue lerend algoritme, dat een accurate voorspelling produceert voor een huishouden met lichte hardware.	
Versie	21 dec 2021	
Contact	Leon de Jong Hans Fugers	Leon.de.jong@alllander.com hans.fugers@alllander.com

## Achtergrond van de challenge

### Alliander geeft vorm aan de energietoekomst van Nederland

Wij staan voor een energievoorziening die iedereen onder gelijke condities toegang geeft tot betrouwbare, betaalbare en duurzame energie. Dat is waar wij iedere dag aan werken. Door ons netwerk continu te verbeteren bereiden we ons voor op de toekomst. Een toekomst waarin iedereen duurzame energie kan gebruiken, produceren en delen.

Alliander ontwikkelt en beheert energienetten. Via onze kabels en leidingen ontvangen ruim drie miljoen Nederlandse huishoudens en bedrijven elektriciteit, gas en warmte. We beheren meer dan 90.000 km elektriciteitsnet en 40.000 km gasnet en zijn er trots op dat onze netten tot één van de betrouwbaarste ter wereld behoren. Onze 7000 collega's zorgen ervoor dat het licht brandt, de huizen warm zijn en de bedrijven draaien. Dit doen we vanuit het maatschappelijk belang om energie voor iedereen betrouwbaar, betaalbaar en bereikbaar te houden.

Deze opdracht is opgesteld door het research center en draagt direct bij aan de betrouwbaarheid van ons energie net. Het Research Center for Digital Technologies geeft vorm aan het digitale netwerkbedrijf van de toekomst. Via deze challenge onderzoeken we hoe we met kleine voorspellingen, namelijk die van de individuele huishoudens, bij elkaar opgeteld een betere prognose kunnen maken voor het verbruik van morgen. Met een betere voorspelling zijn wij beter in staat in te spelen op eventuele problemen. Jij draagt via deze challenge dus direct bij aan de energie transitie!

### De Light Learning Estimator

De Light Learning Estimator (**LLE**) maakt periodiek een voorspelling voor ieder individueel huishouden. Deze huishoudens geven cumulatief inzicht in eventuele problemen op wijkniveau. Deze software moet gaan werken op light weight hardware, voor onze test een Raspberry Pi 4. De software geeft twee keer per dag een prognose af van jouw energieverbruik en energie opwek. Wanneer er iets in jou gebruik veranderd, door bijvoorbeeld de aanschaf zonnepanelen of een laadpaal aan huis, dan stelt de estimator zich binnen enkele dagen bij en wordt de voorspelling weer accuraat.

De LLE is een continuous learning, voorspellend algoritme dat draait op lichte hardware dat zich bevindt aan de uiteinden van het netwerk. De voordelen voor gebruikers en Alliander/Liander op een rijtje:

Voordelen Alliander:

- Beter inzicht in verwacht energieverbruik van huishoudens, en dus wijk en het elektriciteitsnet
- En daarmee het beter kunnen verhelpen van storingen

Voordelen voor gebruikers:

- Privacy-gevoelige data blijft binnen de slimme meter van het huishouden

In deze challenge gaan jullie aan de slag om dit stukje software te ontwikkelen. Hier staat natuurlijk ook iets tegenover:

**Naast eeuwige roem, is de hoofdprijs €1.000,-. De deadline voor inzenden is 6 februari 2022, 23:59.**  
Veel plezier!

## Inhoud

Achtergrond van de challenge .....	2
Uitgangspunten .....	4
Deadline challenge .....	4
Prijzengeld .....	4
Uitgangsmateriaal .....	4
Ontwikkeling van oplossingen .....	4
Inleveren oplossing .....	5
Verificatie en uitslagen .....	5
Trainingsdata .....	5
Gebruiksdata .....	5
Weerdata .....	6
Format .....	6
Opgeleverde product .....	6
Continuous learning .....	7
Toetsing .....	7
Relevante toetsdagen .....	8
Analyse kwaliteit voorspelling .....	8
De winnaar .....	9
Special entry .....	9
Uitkomsten en terugkoppeling .....	10
Bijlages .....	11
Bijlage 2 : Format gebruikersdata .....	11
Bijlage 3 : Tabel applicaties huizen .....	11
Bijlage 4 : Format weerdata .....	12
Bijlage 6: Hardware en OS Raspberry Pi .....	13
Docker .....	13
Bijlage 7: Details voorspelling en measurement .....	14
Details: voorspelling .....	14
Details: Historische meetwaarden oproepen .....	14

## Uitgangspunten

Het idee is dat er een apparaat (agent) wordt aangesloten op de slimme meter van een woonhuis, die een accurate voorspelling leert maken van het huishouden en zich aanpast aan veranderende omstandigheden in het energie-gebruik (continue lerend).

Een uitgangspunt is dat de winnende oplossing qua algoritme en architectuur openbaar worden gemaakt, In overleg kan dit qua timing later worden openbaar gemaakt om er eerst over te publiceren in andere media, onder referentie naar deze challenge.

Alliander blijft ten allen tijden eigenaar van de geleverde inzendingen.

Daarnaast moet de winnende oplossing kunnen uitvoeren op beperkte hardware, hierin nemen wij de Raspberry Pi 4 als uitgangspunt. Naast dat de kwaliteit van de voorspelling wordt getoetst, wordt ook gekeken of de oplossing voldoende snel kan draaien op deze Raspberry Pi.

Het generieke tijdsraam voor data is 5 min. We verwachten dus een voorspelling per 5 min voor de komende 24 uur. Ook wordt de toetsing gedaan over 5 min. Dus bij een hogere frequentie van input wordt deze gemiddeld ( of opgeteld) over de 5 min.

## Deadline challenge

De einddatum van de challenge is zondag 6 februari 2022, 23:59. Vervolgens vindt op maandag 21 februari terugkoppeling plaats over de winnaars.

## Prijzengeld

De Winnaar ontvangt naast eeuwige roem ook 1000 euro. Je wint door in de toetsingsperiodes de beste voorspellingen te doen. Verdere details staan verderop.

De tweede plaats ontvangt 500 euro.

## Uitgangsmateriaal

De datasets van Slimme meter-data valt onder een Non-Disclosure-overeenkomst (NDA) en mag enkel gebruikt worden voor het ontwikkelen van een oplossing. De data dient na afloop te worden verwijderd, zoals opgenomen in de te ondertekenen verwerkingsovereenkomst [\[zie hier\]](#) waarom dit nodig is].

We definiëren de hardware-specificaties welke de dataset zullen verwerken. Bij vragen en/of aanvullende informatie wordt deze aan alle deelnemers verstrekt.

Alliander geeft een score-mechanisme voor de voorspellingen op basis waarvan we een winnaar zullen aanwijzen.

## Ontwikkeling van oplossingen

De partijen ontwikkelen een oplossing welke op de gespecificeerde hardware voorspellingen genereert.

De deelnemer levert de oplossing aan in de vorm van een werkend Python-script (ver 3.10), R-script (ver 3.6.0) of docker-compose, globale beschrijving van de oplossingsaanpak en installatie-instructies voor test-data, welke we op de gespecificeerde hardware kunnen laten werken en de eigen evaluaties van de voorspelling met de scores.

# Lite Learning Estimator Challenge

We verwachten een ingangscontrolle van testdata, die de correcte verwerking van het geleverde algoritme garandeert.

## Inleveren oplossing

Oplossingen kunnen worden ingeleverd op twee manieren:

- Upload de oplossing op je persoonlijke repo en geef ons toegang
- Mailen naar [leon.de.jong@alliander.com](mailto:leon.de.jong@alliander.com)

## Verificatie en uitslagen

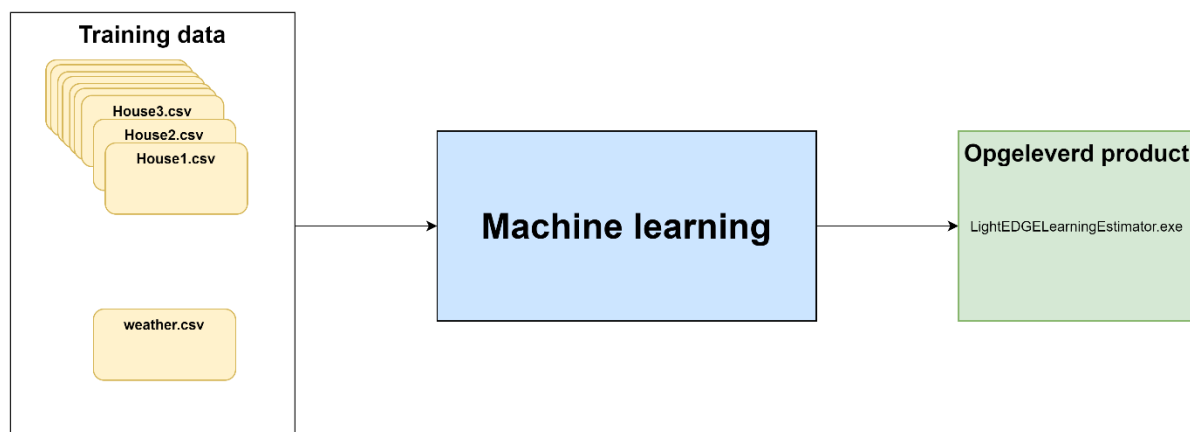
Alliander voert een verificatie uit met een aantal test-scenario's in de vorm van test-data, waarin in sommige gevallen veranderingen zijn opgenomen, namelijk de toevoeging van zonnepanelen (**PV**: photovoltaic)

Indien test-data tot fouten leidt wordt contact gezocht om deze binnen enkele dagen te repareren. Dit wordt maximaal 2x gedaan.

Na doorlopen van alle testscenario's van alle opgeleverde oplossingen worden deze geëvalueerd en de uitslagen gepubliceerd bij alle deelnemers en binnen Alliander.

## Geleverde data

Nadat de Non-Disclosure Agreement (NDA, Zie Bijlage 1) en verwerkersovereenkomst zijn getekend, krijg je toegang tot de data. De data bestaat uit twee sets: Energiedata van huishoudens en weerdata. Beide datasets gaan over dezelfde tijdsperiode: 01-08-2018 t/m 31-08-2020. Dit zijn dus 762 dagen (365+366+31).



## Gebruiksdata

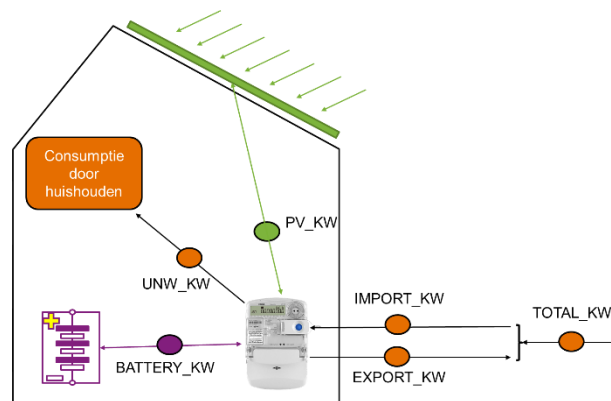
Gebruiksdata bestaan uit minutenwaarden. Het zijn datasets van meerdere huizen, en zijn op te delen in drie situaties (met daaronder de columns behorend bij elke dataset):

- Huizen met Slimme Meters (SM)  
[timestamp, house, IMPORT\_KW, EXPORT\_KW, TOTAL\_KW, UNC\_KW]
- Huizen met Slimme Meters en zonnepanelen (SM en PV)  
[timestamp, house, IMPORT\_KW, EXPORT\_KW, TOTAL\_KW, UNC\_KW, PV\_KW]

# Lite Learning Estimator Challenge

- Huizen met Slimme Meters, zonnepanelen en een batterij (SM, PV en Battery)  
[timestamp, house, IMPORT\_KW, EXPORT\_KW, TOTAL\_KW, UNC\_KW, PV\_KW, BATTERY\_KW]

De columns van de datasets zijn dus afhankelijk van de situatie. In het plaatje hieronder is het nog eens weergegeven:



## Weerdata

Weerdata bestaan uit een grote dataset in uurwaarden, met hierin onder andere de temperatuur, globale straling(J/cm<sup>2</sup>) en zonneschijnduur (in 0.1 uren, dus 10=heel uur zonneshijn). In Bijlage 3 zijn de definities van alle columns te vinden.

## Format

Vermogensdata in kilowatt (kW):

- 60 CSV-files, voor elk huis een aparte file
- In Bijlage 4 is te zien welk huis welke situatie heeft (SM, SM+PV, SM+PV+BATTERY)

Weerdata

- CSV-files

## Opgeleverde product

Het opgeleverde product moet aan de volgende voorwaarden voldoen:

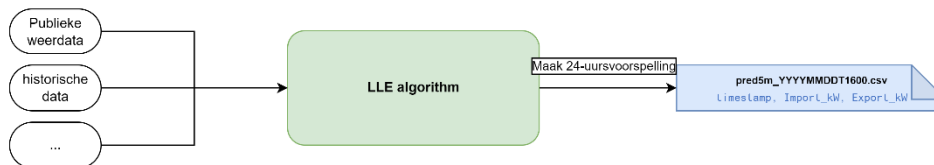
1. Elke dag om 04:00 en 16:00 wordt de csv-file voor voorspellingen gemaakt met de 24-uursvoorspelling voor Import en Export, inclusief timestamps in blokken van 5 minuten. Deze voorspelling wordt gedaan in 288 5-minutenwaarden, waarin wordt voorspeld hoeveel kW gemiddeld wordt gebruikt in die periode. Deze voorspelling wordt opgeslagen als CSV-file. Hierbij kunnen natuurlijk weersvoorspellingen, historische data, en wat dan ook nodig is voor een goede voorspelling in worden meegenomen. Verdere details staan onder Details: voorspelling.

Format CSV met headers:

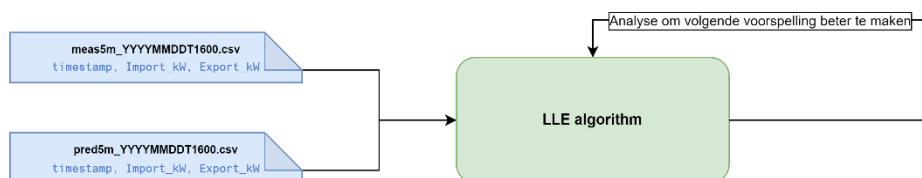
Timestamp, Import\_kW, Export\_kW

2021-11-22T14:00:00+02:00, 0.123, 1.123

2021-11-22T14:05:00+02:00, 0.234, 0.231



2. De LLE blijft leren (*continuous learning*), zodat bij een verandering in het energiepatroon, het na een periode weer accurate voorspellingen kan doen. Als bijvoorbeeld zonnepanelen worden aangeschaft, gaan zowel de patronen van Import\_kW als Export\_kW veranderen, LLE zou hier snel mee om moeten kunnen gaan. Hierbij kan het LLE bijvoorbeeld meetdata, dus de echte energiepatronen, vergelijken met de voorspelling van die dag(en). De daadwerkelijke energiepatronen worden in hetzelfde format opgeslagen als de voorspellingen.



Verdere details over in welke mappen e.d. de bestanden moeten worden opgeslagen, staan in Bijlage 7.

## Continuous learning

Gebaseerd op de kwaliteit van de vorige voorspellingen kan het zelflerend algoritme vervolgens nieuwe voorspellingen aanscherpen, zodat de voorspellingen (gemiddeld genomen) steeds beter worden. Ook als een verandering in Import- en Export-patronen optreedt, is het de bedoeling dat het algoritme hiermee om kan gaan, en binnen 7 dagen weer een redelijk voorspelling doet. De implementatie hiervan is aan de deelnemer.

## Toetsing

Als je oplossing van de Light Learning Estimator is ingeleverd, dan gaan wij hem zo snel mogelijk toetsen. Dit vindt op de volgende manier plaats:

De toetsing wordt gedaan aan de hand van slimme meterdata van huizen die vooraf zijn achtergehouden. Sommige hiervan bevatten zonnepanelen, andere niet of later. Het aangeleverde product krijgt steeds de data van een enkel huishouden gevoed om te kijken hoe goed het energiepatroon van dat specifieke huishouden voorspeld kan worden.

Nadat het is aangesloten wordt per 12 uur (om 04:00 en 16:00) de meetdata van die dag ingevoerd, waarna het algoritme steeds een voorspelling doet en kan leren aan de hand van de daadwerkelijke meetwaarden. Historische weerdata van die dag is ook als 'voorspelling' beschikbaar in de format waarin het ook is geleverd. Mocht je nog slimme andere externe bronnen gebruiken voor een goede voorspeller, dan is het handig dat je hiervan historische waarden aanlevert van de periode 01-08-2018 tot 31-08-2020, zodat we deze ook in kunnen voeren.

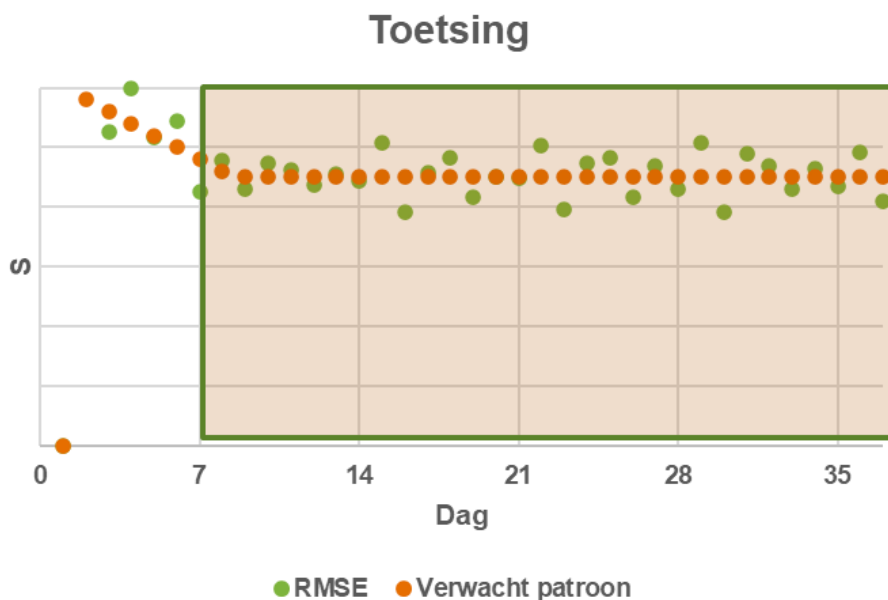
Vervolgens wordt voor elke voorspelling die gemaakt wordt door de LLE, getoetst aan de hand van de echte waarden. Dit gebeurt aan de hand van de Root Mean Square Error-methode. Hier komt voor elke dag een waarde uit.

N.B. De volgorde van gebeurtenissen om 04:00 en 16:00 is als volgt:

1. CSV met voorspelling wordt opgehaald
2. CSV met meetwaarden wordt gevoed en RMSE wordt bepaald
3. Nieuwe voorspelling wordt gemaakt door ingeleverd LLE algoritme

## Relevante toetsdagen

Eerst krijgt het product 7 dagen de tijd om aan het gebruikspatroon te wennen, om vervolgens in een 'toetsperiode' te belanden van 30 dagen, waarin wordt gekeken naar de RMSE-waarden van die dagen. Dit leidt dus tot 60 RMSE-waarden (2 per dag\*30 dagen).



Vervolgens volgt er na een x aantal dagen nog een keer hetzelfde gedaan. Deze waarde x is van tevoren bepaald, en zorgt ervoor dat de tweede toetsing gebeurt vlak nadat bij sommige huizen zonnepanelen worden geïnstalleerd, om te kijken of de LLE om kan gaan met deze verandering. Deze waarde x is van tevoren dus wel bekend bij de jury.

De gemiddelde RMSE-waarde wordt bepaald nadat de LLE op data van meerdere huizen is getoetst, en degene die gemiddeld genomen over de scenario's de laagste RMSE-waarde heeft, wint.

## Analyse kwaliteit voorspelling

24 uur nadat de voorspelling is gedaan, vindt een evaluatie plaats van hoe goed de voorspellingen van Import\_kW en Export\_kW overeenkwamen met de werkelijkheid. Voor elke 5-minutenwaarde wordt gekeken naar het absolute verschil tussen de voorspelling en de werkelijkheid. De RMSE- formule wordt toegepast:



$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Predicted_i - Actual_i)^2}{N}}$$

Dit wordt zowel voor de voorspelling van Import\_kW als voor Export\_kW gedaan door [meas5m\\_20211122T1600.csv](#) en [pred5m\\_20211122T1600.csv](#) op te roepen en het verschil te bepalen.

Een klein werkvoorbeeld met 7 willekeurige Import\_kW-waarden (bij de daadwerkelijke challenge worden dit er natuurlijk 288 per dag voor opwek en 288 per dag voor verbruik):

timestamp	IMPORT_KW	PRED_IMPORT_KW	delta
2021-11-22T14:00:00+02:00	1.08	1.12	0.04
2021-11-22T14:05:00+02:00	1.34	1.42	0.08
2021-11-22T14:10:00+02:00	1.45	1.23	0.22
2021-11-22T14:15:00+02:00	1.21	1.32	0.11
2021-11-22T14:20:00+02:00	1.32	1.33	0.01
2021-11-22T14:25:00+02:00	1.29	1.22	0.07
2021-11-22T14:30:00+02:00	1.28	1.25	0.03

		RMSE
		0.28

Voor Import\_kW en Export\_kW wordt apart een RSME-waarde bijgehouden.

## De winnaar

De winnaar is degene met de laagste gemiddelde score *RMSE*.

Natuurlijk testen we het algoritme ook op een Raspberry Pi 4, en zou hij hierop moeten kunnen *runnen*.

## Special entry

Het kan zijn dat iemand een heel andere manier vindt om een meer accurate voorspelling te maken, bijvoorbeeld door inzet van andere hardware of andere publiek toegankelijk (open) data.

Een entry in deze categorie moet voldoende uitgebreid beschrijven, waar deze afwijkt en wat de verbetering is die daarmee wordt bereikt. Met voldoende bedoelen we dat we de opzet onafhankelijk kunnen toetsen op basis van de beschrijving en aangeleverde code.

De eindscore moet dan tenminste 50% beter scoren dan de concurrentie.

# Lite Learning Estimator Challenge



## Uitkomsten en terugkoppeling

Twee weken na de deadline, op 21 februari, volgt de bekendmaking van de winnaars.

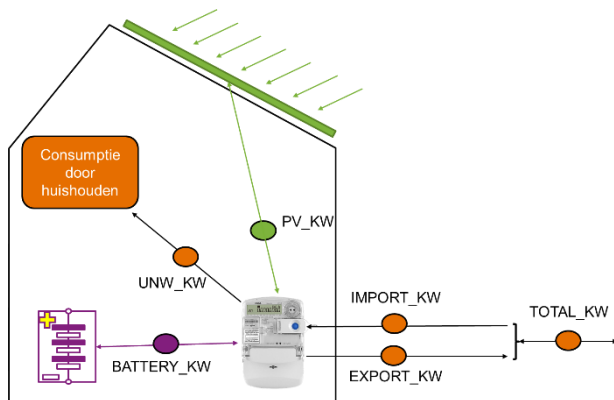
## Bijlages

### Bijlage 2 : Format gebruikersdata

Column naam	betekenis	Eenheid/format
Timestamp	tijd	yyyy-mm-ddThh:mm:ss+02:00
House	Geeft aan om welk huis het gaat	-
IMPORT_KW	Vermogen van grid naar SM	kW
EXPORT_KW	Vermogen van SM naar grid	kW
TOTAL_KW	IMPORT_KW – EXPORT_KW	kW
PV_KW*	Vermogen opgewekt door PV	kW
BATTERY_KW**	Vermogen van batterij naar SM	kW
UNC_KW	Vermogen gebruikt door huishouden	KW

\* Alleen bij huizen die na verloop van tijd zonnepanelen krijgen

\*\*Alleen bij huizen die na verloop van tijd een batterij hebben



### Bijlage 3 : Tabel applicaties huizen

1 = 'huis bezit deze applicatie'

House	Smart meter?	PV panels?	Battery ?
1	1	1	1
3	1	1	
4	1	1	
5	1	1	1
7	1	1	1
8	1		
9	1	1	
10	1	1	
11	1	1	
13	1	1	
15	1	1	1
16	1	1	1
17	1	1	
19	1	1	
20	1	1	
21	1	1	

# Lite Learning Estimator Challenge

24	1	1	
25	1	1	
26	1		
27	1		
28	1	1	
29	1	1	1
30	1		
32	1	1	
33	1	1	
34	1		
35	1	1	
36	1		
37	1		
38	1	1	
39	1		
40	1	1	
42	1		
43	1	1	
44	1	1	
45	1	1	
46	1	1	
48	1	1	
50	1	1	
52	1	1	
53	1		
56	1	1	
57	1	1	1
58	1		
59	1	1	
60	1	1	
61	1		
62	1		
63	1		
64	1	1	
66	1		
68	1		
69	1	1	
71	1		
72	1		
73	1		
75	1		
Total	57	37	7

## Bijlage 4 : Format weerdata

Bron: <https://www.knmi.nl/nederland-nu/klimatologie/uurgegevens>, weerstation 278 – Heino, ligt op 15 km van de huizen

# Lite Learning Estimator Challenge

## Bijlage 6: Hardware en OS Raspberry Pi

We hanteren een Raspberry PI4+ met 4 Gb geheugen en als OS de laatste versie van 'Raspberry Pi OS'

Na installatie wordt een update en upgrade uitgevoerd:

Dus in de terminal : `<ctrl><alt>-T`

```
$ sudo apt-get update
```

```
$ sudo apt-get full-upgrade
```

Expand file-system to get more space

```
$ sudo raspi-config
```

Selecteer optie 7 Advanced Options

Selecteer daarna A1 Expand filesystem

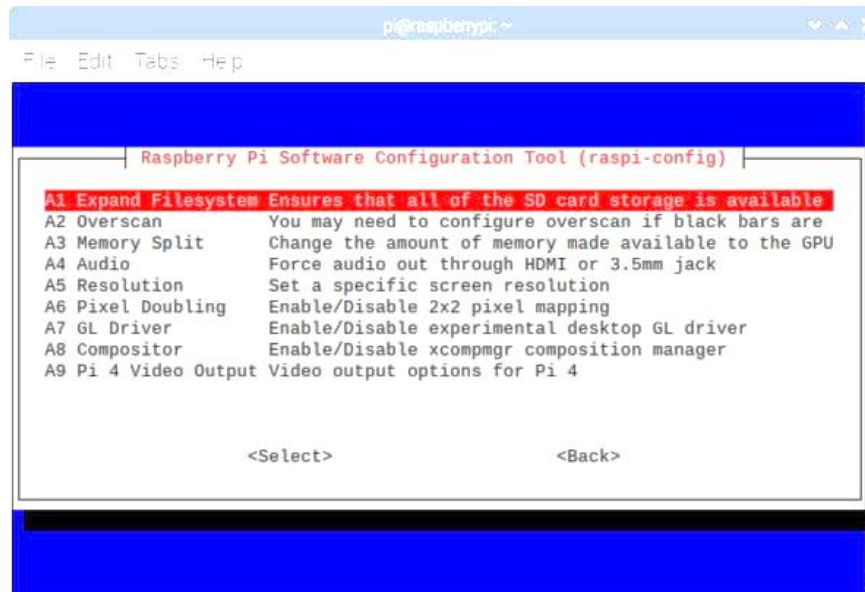
Herstart met

```
$ sudo reboot
```

Met het commando

```
$ df -h
```

ziet men het resultaat van de verkregen ruimte.



Voor nog meer ruimte :

```
$ sudo apt-get purge wolfram-engine
```

```
$ sudo apt-get purge libreoffice*
```

```
$ sudo apt-get clean
```

```
$ sudo apt-get autoremove
```

## Docker

Zie

ook

link

[URL](#)

Voor de installatie van docker download de installatie-script met:

```
$ curl -fsSL https://get.docker.com -o get-docker.sh
```

Execute het script met:

```
$ sudo sh get-docker.sh
```

Om de huidige gebruiker het leven eenvoudig te maken en sudo overbodig te maken voor docker-taken:

```
$ sudo usermod -aG docker &USER$
```

De default RPI-gebruiker - \$USER\$ is ook als 'pi' te schrijven.

## Bijlage 7: Details voorspelling en measurement

### Details: voorspelling

Elke dag om 04:00 en 16:00 wordt een voorspelling gedaan over de import- en exportwaarden van de komende 24 uur. De voorspelling maakt een voorspelling van elk tijdsblok van 5 minuten. Dit leidt dus tot  $24 \times 12 = 288$  waarden voor energiegebruik (Import\_kW) en teruglevering (Export\_kW).

Dit resulteert in een reeks waarden met time-stamp (start van de 5 min-interval) en 2 float waarden die de gemiddelde Import en Export kW aangeven gedurende die 5 min. De som van deze twee waarden is dus het netto verbruik door het huishouden wat normaliter de geplande belasting van het elektriciteitsnet vormt.

Voor het toetsen van deze voorspellingen dienen deze reeksen in CSV-files te worden opgeslagen.

De file heeft als naam `pred_<timestamp>.csv` op de locatie: [/home/predictions](#)

Bijvoorbeeld filenaam : `pred5m_20211122T1600.csv`

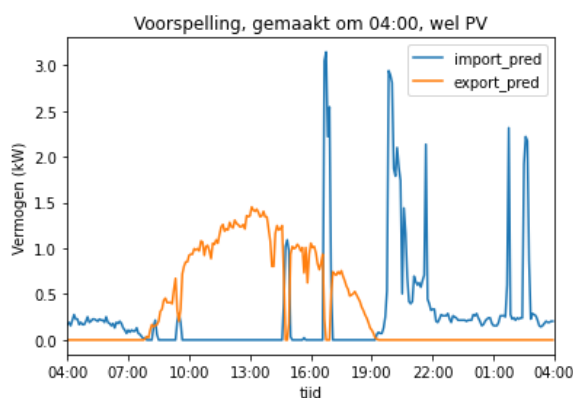
Te relateren aan trainingsdata

Format CSV met headers:

```
Timestamp, Import_kW, Export_kW
2021-11-22T14:00:00+02:00, 0.123, 1.123
2021-11-22T14:05:00+02:00, 0.234, 0.231
...
```

N.B. De timestamp het begin van het interval aan, dus `2021-11-22T14:00:00+02:00` bevat gemiddelde Import en export (kW) tussen 14:00:00 – 14:04:59 op 22-11-2021.

Geplot ziet de .csv er bijvoorbeeld als volgt uit:



### Details: Historische meetwaarden oproepen

Om het algoritme *continuous learning* te maken, kan het van pas komen om de voorspelling na 24 uur te toetsen aan de daadwerkelijke meetdata.

Voor elke dag kan de meetdata gevonden worden onder de naam `meas5m_<timestamp>.csv` op de locatie: [/home/measurements](#)

Bijvoorbeeld filenaam : `meas5m_20211122T1600.csv`

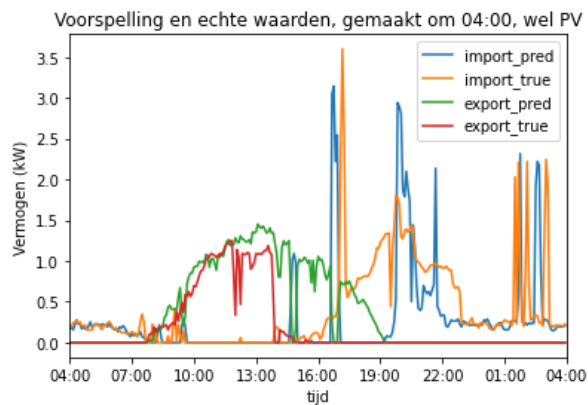
# Lite Learning Estimator Challenge

Format van deze CSV met headers:

```
Timestamp, Import_kW, Export_kW  
2021-11-22T14:00:00+02:00, 0.123, 1.123.  
2021-11-22T14:05:00+02:00, 0.234, 0.231.
```

Terwijl de prediction-file om 04:00 en 16:00 gelijk wordt gevuld met 288 waarden, wordt aan de measurement-file gedurende 24 uur elke 5 minuten een line toegevoegd.

Hieronder een voorbeeld van wanneer `meas5m_20211122T1600.csv` en `pred5m_20211122T1600` na 24 uur geplot zouden worden:



===== the end =====