

Light Learning Estimator Challenge

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



Title	Lite EDGE Learning Estimator 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	15 dec 2021		
Contact	Leon de Jong	Leon.de.jong@alllander.com	
	Hans Fugers	hans.fugers@alllander.com	

Lite EDGE Learning Estimator Challenge



Achtergrond van de challenge

Het stroomnet loopt in snel tempo vol, door spanningsproblemen kunnen bijvoorbeeld steeds meer zonnepanelen, laadpalen, et cetera niet meer worden aangesloten en mensen moeten steeds langer wachten op hun verzwaarde aansluitingen. De energie-transitie zorgt voor drukke jaren voor de netwerkbedrijven als Alliander.

Digitalisering brengt ook steeds meer mogelijkheden, zoals netsturing op afstand en beter inzicht in waar congestie aanwezig is en op gaat treden.

Alliander werkt hard aan verschillende innovaties die in de toekomst veel van onze huidige en toekomstige issues gaan oplossen en nieuwe mogelijkheden creëert voor klanten en partners.

Zo wordt bij Research Center for Digital Technologies [RCDT] gekeken naar de mogelijkheden van de Digital Mesh [DM]. Binnen DM doen we onderzoek naar een decentrale opzet van Smart Assets die voorzien zijn van 'intelligentie' in de vorm van computer, storage en communicatie. Wanneer hardware goedkoper, krachtiger en kleiner wordt, zullen we deze 'intelligentie' in elke asset gaan aantreffen. Elke asset kan dan life-time recording van zichzelf realiseren en autonoom acties uitvoeren.

Als mogelijk onderdeel van de 'intelligentie' zien wij een Light EDGE Learning Estimator, die periodiek een voorspelling maakt en daarmee gezamenlijk op wijkniveau anticiperen op problemen mogelijk maakt. Deze software moet dan wel werken op beperkte hardware, zoals de Raspberry Pi, die vervolgens aan de slimme meter bij jou thuis wordt bevestigd. De software geeft vervolgens twee keer per dag een voorspelling van jouw energiepatronen, zowel verbruik als opwek. Mocht er iets in jouw gebruik veranderen, door bijvoorbeeld de aanschaf zonnepanelen of een elektrische auto, dan heeft de Estimator zich binnen enkele dagen aangepast en volgen nieuwe goede voorspellingen.

De Light EDGE Learning Estimator is dus 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 op een rijtje:

Voordelen voor gebruikers:

- Inzicht in verbruik en de drukke uren die voor problemen kunnen zorgen
- Privacy-gevoelige data blijft veilig bij jou
- Door drukke periodes te vermijden : minder stroomstoringen

Voordelen voor Alliander:

- Minder storingen als mensen gaan anticiperen op drukte
- Minder datastromen, de data blijft lokaal
- Real-time inzicht in het net, waardoor netbeheer betrouwbaarder en betaalbaarder wordt

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!

Lite EDGE Learning Estimator Challenge

alliander

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.

Winnaar

De Winnaar(s) ontvangt naast eeuwige roem ook 1000 euro.

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 [\[zie hier\]](#).

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.

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

Met opmerkingen [HF1]: Nader in te vullen, overleg met Gerben en Tom

Met opmerkingen [HF2]: Is er geen bewerkersovereenkomst nodig????

Met opmerkingen [LJ3R2]: <https://intranet.alliander.com/ondersteunend-materiaal>

Met opmerkingen [GH4R2]: Je kan ook nog naar deze refereren: <https://www.alliander.nl/privacystatement>

Met opmerkingen [SJ5]: als je helemaal architectuur ook wil hebben dan is wel meer nodig dan 1 docker file. Een docker-compose setup zou ik ook accepteren.

Met opmerkingen [HF6R5]: prima

Met opmerkingen [SJ7]: wat wordt hier mee bedoeld? Dat de data binnen de correcte marges valt of dat het voldoet aan een bepaalde schema?

Met opmerkingen [HF8R7]: Controleer de input zodat onze test-opzet niet tot problemen gaat leiden, zoals andere map-namen/locaties, formatdingetjes, etc..

Lite EDGE Learning Estimator Challenge



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.

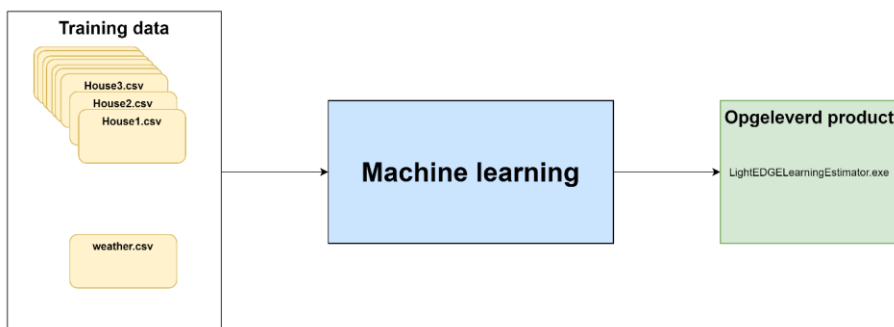
Met opmerkingen [SJ9]: wat voor fouten verwachten we hier? Verschillen in dataformat ofzo?

Met opmerkingen [HF10R9]: De oplossing kan in onze tests "spak" lopen hoe vaak gaan we dit dan oplossen?

Lite EDGE Learning Estimator Challenge

Trainingsdata

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).

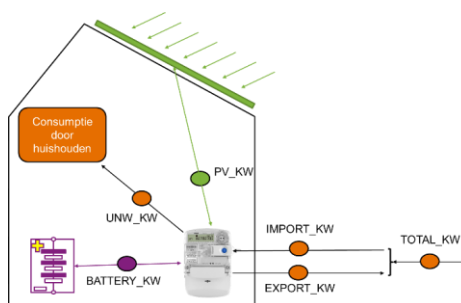


Gebruiksdata

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]
- 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:



Lite EDGE Learning Estimator Challenge

alliander

Weerdata

Weerdata bestaan uit een grote dataset in uurwaarden, met hierin onder andere de temperatuur(°C), globale straling(J/cm²) en zonneschijnduur (in 0.1 uren, dus 10=heel uur zonneschijn). 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 een voorspelling geven van het energie-opwek en -verbruik van een huishouden. 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.
 - a. Optioneel kan het product hiervoor externe weersvoorspellingen raadplegen.
2. Daadwerkelijke gemiddelde **Import- en Export-waarden** kunnen verkrijgen uit de Slimme meter en opslaan in 5-minutenwaarden, ook als deze waarden op een hogere frequentie (bijvoorbeeld in 1 of 10-secondenwaarden) binnenkomen. Deze worden ook opgeslagen als CSV-file.
3. Elke dag om 04:00 en 16:00 een analyse maken van de voorspelling die 24 uur daarvoor gedaan is. De toetsing wordt gedaan over 5 min-intervallen, dus bij een hogere frequentie van input wordt deze gemiddeld (of opgeteld) over de 5 minuten.
4. Het algoritme blijft leren (continuous learning), zodat bij een verandering in het energiepatteroon, het na een periode weer accurate voorspellingen kan doen. Voorbeelden zijn het aanschaffen of wegdoen van een elektrische auto, nemen van zonnepanelen, of een veranderde gezinssamenstelling.
 - a. Ons eerste idee is dat het algoritme zich na 15 dagen weer heeft aangepast

Details: voorspelling

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

Dit resulteert dus 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.

Deze reeks dient dan als voorspelde waarden, die met de realiteit van de metingen (na afloop) leiden tot bijstelling van het algoritme dat de voorspelling maakt.

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

Met opmerkingen [LN11]: Je hebt nog nergens verteld wat import en export is. (Zit in het plaatje, maar was mij niet meteen duidelijke). Als ik als normale sterveling op ctrl+f "import" doe dan zal ik het niet vinden.

Met opmerkingen [LJ12R11]: Als het goed is kan dat nu wel, het staat beneden in de bijlage gedefiniëerd

Lite EDGE Learning Estimator Challenge



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

Bijvoorbeeld filenaam : [pred5m_20211122T1600.csv](#)

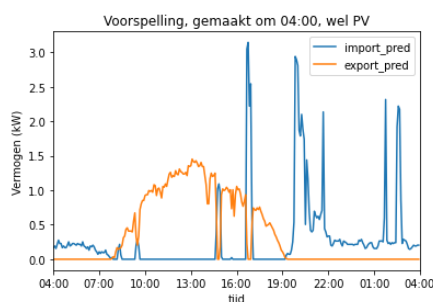
Te relateren aan trainingsdata

Format CSV met headers:

```
Timestamp, Import, Export
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: werkelijke waarden meten

Het opgeleverde product bepaalt elke 5 minuten de gemiddelde Import- en Exportwaarde in kW, zodat het vergeleken kan worden met de voorspelling. Ook als deze waarden op een hogere frequentie (bijvoorbeeld in 10-secondenwaarden) binnenkomen. Welke data door de Slimmer Meter verstrekt kan worden, is te vinden in Bijlage 5.

Dit kan bijvoorbeeld worden gedaan door elke 5 minuten te kijken naar de waarden van:

- `electricityReceivedLowTariff`
- `electricityReceivedNormalTariff`
- `electricityReturnedLowTariff`
- `electricityReturnedNormalTariff`

Door deze waarden elke 5 minuten te registreren, kunnen de gemiddelde Import- en Export-waarden in kW worden bepaald. Voor het toetsen van de voorspellingen dienen deze reeksen in CSV-files te worden opgeslagen.

Met opmerkingen [HF13]: We gaan toch geen tarief-verschil maken???

Met opmerkingen [LJ14R13]: Nee inderdaad, zeker niet. Maar deze dingen worden momenteel wel gemeten door een slimme meter.

Lite EDGE Learning Estimator Challenge



Om makkelijk te kunnen vergelijken met het voorspellingsbestand dient elke dag om 04:00 en 16:00 een nieuw CSV-file aangemaakt te worden aangemaakt, op hetzelfde moment als [pred5m_20211122T1600.csv](#).

De file heeft als naam `meas_<timestamp>.csv` op de locatie: `/home/measurements`

Bijvoorbeeld filenaam : [meas5m_20211122T1600.csv](#)

Format van deze CSV met headers:

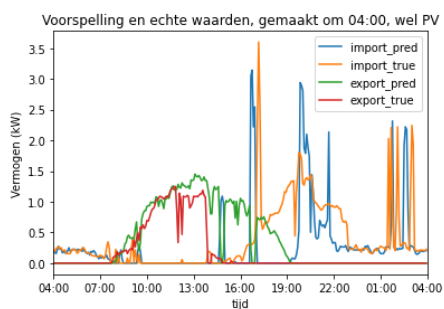
Timestamp, Import, Export

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:



Details: [Analyse kwaliteit voorspelling](#)

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

$$S_t = \sqrt{\frac{1}{n} \sum_{i=1}^n (Actual_i - Prediction_i)^2} \quad n = 1, 2, \dots, 288$$

$$E = \frac{1}{n} \sum_{t=1}^{2d} S_t \quad t = \# \text{ 24-hour predictions} = 2 * \# \text{ days} = 1, 2, \dots, 14$$

Dit wordt zowel voor de voorspelling van Import als voor Export gedaan door [meas5m_20211122T1600.csv](#) en [pred5m_20211122T1600.csv](#) op te roepen en het verschil te bepalen.

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

Lite EDGE Learning Estimator Challenge

alliander

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

		S
		0.28

Bij een perfecte voorspelling heeft S een waarde van 1, en deze waarde daalt naarmate het verschil tussen voorspelling en realiteit groter wordt. Voor Import en Export wordt apart een S-waarde bijgehouden. Deze wordt weer naar een aparte CSV-file geschreven. Elke maand wordt een nieuwe CSV-file hiervoor aangemaakt, met daarin de S-waarden van die maand.

De file heeft als naam `quality_<TTTTMM>.csv` op de locatie: `/home/quality`

Bijvoorbeeld filenaam : `quality_202111.csv`

Format van deze CSV met headers:

```
Timestamp, S-Import, S-Export
20211122T0400, 0.804, 0.304
20211122T01600, 0.764, 0.673
```

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-waarden 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.

Met opmerkingen [HF15]: Is dit niet veel te lang/kort : checken bij experts

Toetsing

De Toetsing wordt gedaan aan de hand van 5 scenario's waarin het opgeleverde product wordt getoetst. Hierin worden meerdere maanden aan slimme meter data van een enkel huishouden gevoed aan het opgeleverde product. Eerst krijgt het product tijd om aan het gebruikspatroon te wennen, om vervolgens in een 'toetsperiode' te belanden, waarin wordt gekeken naar de S-waarden. De gemiddelde S-waarde wordt bepaald, en degene die gemiddeld genomen over de scenario's de beste S-waarden heeft, wint.

Met opmerkingen [SJ16]: Hier moet nog heel scherp uitgelegd worden hoe precies die data 'gevoed' wordt. Is er een endpoint of functie die in een loop 1 voor 1 nieuwe datapunten krijgt? Is het een grote dump van 12 uur aan data?

$$S_t = \frac{1}{n} \sum_{i=1}^n (Actual_i - Prediction_i)^2 \quad n = 1, 2, \dots, 288$$

$$E = \frac{1}{n} \sum_{t=1}^{24} S_t \quad t = \# 24 - \text{hour predictions} = 2 * \# \text{days} = 1, 2, \dots, 14$$

Met opmerkingen [SJ17]: misschien is het het waard om een repo te maken waarin mensen verder kunnen werken door bijv. een branch te maken/te forken. Zodat de manier de ML algo aanroepen hetzelfde is. Misschien de code waarmee je gaat testen openbaar maken, en een neppe testcase uploaden.

Lite EDGE Learning Estimator Challenge

alltander

Testscenario's

De scenario's bevatten vergelijkbare patronen van de trainingsdata en zouden na 7 dagen accuraat moeten zijn. In zijn algemeenheid wordt in alle scenario's de eerste week met data genegeerd om aan te passen aan de test-data.. Na deze week volgt een toetsperiode, waarin de voorspellingen van het algoritme worden beoordeeld op accuratesse.

In scenario 2 t/m 5 worden veranderingen toegepast aan het energiepatroon, om te kijken of het algoritme om kan gaan met de verandering. Eerst volgt weer een week om het algoritme te laten leren, om vervolgens weer een maand getoetst te worden.

De scenario's zijn hieronder visueel weergegeven.



Met opmerkingen [LJ18]: Plaatje aanpassen -> aanpassingsmaand -> week

Met opmerkingen [LJ19R18]: S-waarde vervangen voor S-waarden (kwaliteit)

Met opmerkingen [GH20]: Plaatje aanpassen nav 7 dagen adaptief zijn

Scenario 1 - gelijke omstandigheden

Een benchmarktest met een huis zonder zonnepanelen. Eerst heeft het algoritme een week (7 dagen) de tijd om te wennen aan de energiepatronen, vervolgens wordt het een maand getoetst.

Resultaat: Een lijst van 60 scores voor de kwaliteit van de voorspellingen.

Scenario 2 - na 2 maand, introductie van PV

De eerste twee maanden zijn identiek aan scenario 1, maar vervolgens wordt op dag 61 een elektrische auto toegevoegd aan het huishouden, waardoor de energiepatronen veranderen.

Resultaat: Twee lijsten:

- 60 kwaliteitsscores voor maand 2
- 60 kwaliteitsscores voor maand 4

De winnaar

De winnaar is degene met de laagste gemiddelde score S , hieronder weergegeven als E . Mocht het voorkomen dat een deelnemer niet de hoogste gemiddelde S -waarde heeft, maar wel bij meer dan 1 scenario de laagste E -waarde haalt, dan wordt de prijs gedeeld: De eerste plaats krijgt 1000 euro, de tweede 500 euro.

Lite EDGE Learning Estimator Challenge



$$S_t = \sqrt{\frac{1}{n} \sum_{i=1}^n (Actual_i - Prediction_i)^2} \quad n = 1, 2, \dots, 288$$

$$E = \frac{1}{n} \sum_{t=1}^{2d} S_t \quad t = \# \text{ 24-hour predictions} = 2 * \# \text{ days} = 1, 2, \dots, 14$$

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 op tenminste 50% van de scenario's beter scoren dan de concurrentie.

Uitkomsten en terugkoppeling

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

Lite EDGE Learning Estimator Challenge

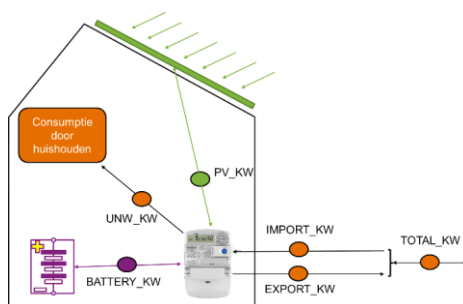


Bijlages

Bijlage 1 : Non Disclosure Agreement

Bijlage 2 : Format gebruikersdata

Column naam	betekenis	Eenheid/format
Timestamp		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



Bijlage 3 : Tabel applicaties huizen

1 = 'huis bezit deze applicatie'

House	Smart meter?	PV panels?	Battery?
1	1	1	1
2	1		
3	1	1	
4	1	1	
5	1	1	1
6	1	1	1
7	1	1	1
8	1		
9	1	1	
10	1	1	
11	1	1	
12	1		
13	1	1	
14	1	1	
15	1	1	1
16	1	1	1

Lite EDGE Learning Estimator Challenge



17	1	1	
18	1	1	
19	1	1	
20	1	1	
21	1	1	
22	1		
23	1	1	
24	1	1	
25	1	1	
26	1		
27	1		
28	1	1	
29	1	1	1
30	1		
31	1	1	
32	1	1	
33	1	1	
34	1		
35	1	1	
36	1		
37	1		
38	1	1	
39	1		
40	1	1	
41	1	1	
42	1		
43	1	1	
44	1	1	
45	1	1	
46	1	1	
47	1		
48	1	1	
49	1	1	
50	1	1	
51	1	1	
52	1	1	
53	1		
54	1		
55	1		
56	1	1	
57	1	1	1
58	1		
59	1	1	
60	1	1	
61	1		
62	1		

Lite EDGE Learning Estimator Challenge



63	1		
64	1	1	
65	1	1	
66	1		
68	1		
69	1	1	
70	1	1	
71	1		
72	1		
73	1		
74	1		
75	1		
76	1		
77	1		
-			
Total	76	47	8

Bijlage 4 : Format weerdata

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

Bijlage 5 : Format SM data

Meting	Betekenis
ident	ID van de gebruiker
p1Version	Versie van slimme meter
timestamp	Tijddata van de elektriciteitsmeter
equipmentId	ID van de slimme meter
electricityTariffIndicator	Geeft aan welk tarief op dat moment actief is
electricityReceivedLowTariff	Houdt bij hoeveel energie op laag tarief is ontvangen (kWh)
electricityReceivedNormalTariff	Houdt bij hoeveel energie op normaal tarief is ontvangen (kWh)
electricityReturnedLowTariff	Houdt bij hoeveel energie op laag tarief is teruggeleverd (kWh)
electricityReturnedNormalTariff	Houdt bij hoeveel energie op normaal tarief is teruggeleverd (kWh)
electricityPowerReceived	
electricityPowerReturned	
powerFailures	Hoeveelheid stroomonderbrekingen
longPowerFailures	Hoeveelheid langdurige stroomonderbrekingen
powerFailureEventLogSize	
voltageSagsPhaseL1	Aantal spanningsdalingen op fase L1
voltageSwellsPhaseL1	Aantal spanningsophogingen op fase L1
voltageL1	Spanning van fase L1 in volt (V)
currentL1	Stroom van fase L1 in ampère (A)
powerReceivedL1	Huidig verbruik van fase L1 in Watt (W)
powerReturnedL1	Huidige teruglevering van fase L1 in Watt (W)

Lite EDGE Learning Estimator Challenge



message	
gasEquipmentId	Serienummer van de aardgasmeter
gasTimestamp	tijddata van de aardgasmeter
gasM3	Stand van de gasmeter in gebruikte kubieke meters gas (m3)

Lite EDGE Learning Estimator Challenge

alltander

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'

Met opmerkingen [LN21]: Vanaf versie noemen (welke versie er nu en alles wat daarna komt).

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.

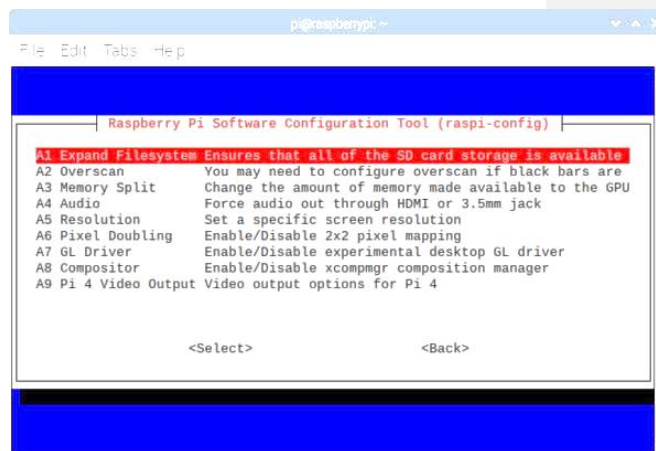
Voor nog meer ruimte :

```
$ 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.

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