



Radboud Universiteit Nijmegen

Modellenpracticum Stroomrichtingen

Auteurs:

Antonie BLOM
Victor BOSMAN
Kees VAN KEMPEN
Luc KOENEN
William KUIJLTJES

Begeleiders:

Wieb BOSMA
Jacco HERES
Martijn VAN BRAAK-KASKENS

stroomrichting2020@keesvankempen.nl

01-08-2020

Inhoudsopgave

1	Inleiding en theorie	2
1.1	Topologie van het stroomnet	2
1.2	Rekenen aan het net	3
1.3	Beschikbare gegevens	4
1.3.1	EMS-data	4
1.3.2	Profielendata	4
1.3.3	Routemapdata	5
2	Voorverwerking van data	6
3	Modellen	7
3.1	Machine learning	7
3.2	Absoluut lineair verband	8
3.3	Rising/falling edges	13
4	Conclusie en discussie	15
4.1	Windenergie	15
4.2	Warmte-krachtkoppelingsinstallaties	16
4.3	Verbeteren zonne-energiemodel	16
4.4	Machine Learning	18
A	--help for various commands	20
A.1	model.py	20
A.2	rds2feather.py	20
A.3	emplume.py	20
A.4	defluff.py	21
A.5	tapeduck.py	21

1 Inleiding en theorie

Het huidige stroomnet is het resultaat van meer dan honderd jaar ontwikkeling. Het is gebouwd op het principe van centrale opwekking door energiecentrales en verbruik door klanten. De laatste jaren is dit tweede echter veranderd door de komst van zonnepanelen, windmolens en andere wijzen van teruglevering door klanten: het is geen eenrichtingsverkeer meer. Voor een goed inzicht in de belasting van het netwerk, is het voor de netbeheerder belangrijk te weten wat de richting van deze stromen is. Met die kennis kan de netbeheerder bepalen of er nog capaciteit over is en er nog klanten op het netwerk kunnen worden aangesloten, of dat dat niet meer past.

In dit project hebben wij gekeken naar mogelijke modellen voor het bepalen van de stroomrichting binnen delen van het netwerk op basis van de door Alliander beschikbaar gestelde gegevens. We zullen eerst het netwerk en deze gegevens beschrijven. Daarna gaan we in op de door ons ontwikkelde modellen en bespreken we daarvan de resultaten.

1.1 Topologie van het stroomnet

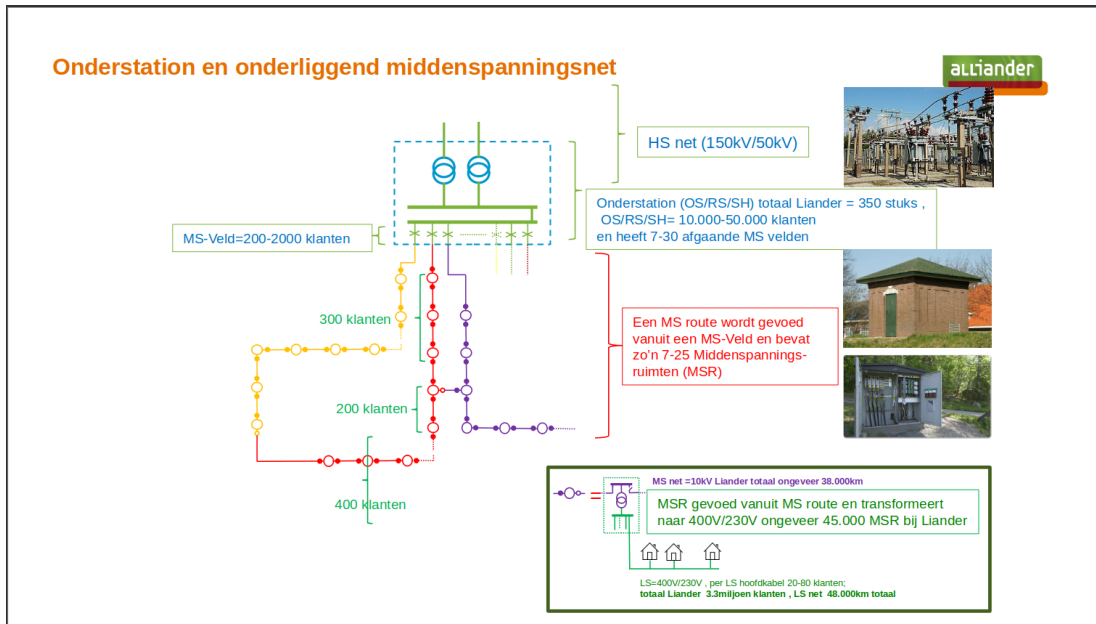
Het spanningsnet is op te delen in drieën: het hoogspannings-, middenspannings- en laagspanningsnet. Voor een begrip van de gegevens kijken we naar de koppelingen tussen deze netwerken.

Het hoogspanningsnet wordt beheerd door Tennet. Grote vermogens worden hierin vanaf energiecentrales over grote afstanden vervoerd. Bij onderstations wordt deze hoogspanning omgevormd naar middenspanning. Alliander is belast met het beheer van het middenspanningsnet in een groot deel van Nederland. De gegevens die in sectie 1.3 worden besproken komen voor een groot deel uit metingen bij de onderstations.

In het netwerk hebben we te maken met drie typen stations die de schakel tussen het hoog- en middenspanningsnet vormen. Deze typen worden hieronder omschreven.

- **Onderstations** koppelen het middenspanningsnet aan het hoogspanningsnet (Tennet). Bij onderstations wordt de hoogspanning getransformeerd naar middenspanning. Hiervoor zijn dure transformatoren nodig. Deze koppeling wordt geïllustreerd in figuur 1.
- **Schakelstations** zijn kleinere stations die gekoppeld zijn aan één of meer onderstations (met kabels van rond de 10 km tot de onderstations). Door deze vaak redundante aansluiting wordt bescherming geboden tegen uitval.
- **Regelstations** zijn een onderklasse van schakelstations die de spanning enigszins opkrikken. Ze worden vooral gebruikt in dunbevolkte gebieden en zitten vaak op een afstand van meer dan 10 km van onderstations. Door deze grote afstand is er een significant spanningsverval en is het omhoog transformeren van de spanning nodig voordat de stroom verdeeld wordt naar de verbruikers.

Vanaf deze stations lopen aftakkingen het middenspanningsnetwerk in. Deze aftakkingen heten afgaande velden. De afgaande velden vormen routes. Een route kan gevoed worden door meerdere afgaande velden. Op die manier wordt de belasting over de kabels verdeeld en is de route beschermd tegen uitval van een afgaand veld. Deze schakelingen worden geïllustreerd in figuur 1.



Figuur 1: Een onderstation koppelt het hoogspanningsnet (50 kV–150 kV) aan het middenspanningsnet (7 kV–25 kV).[1, p.10]

Tot slot wordt in woonwijken de middenspanning getransformeerd naar laagspanning (LS, 230 V–400 V) voor gebruik in huishoudens. Dit gebeurt in middenspanningsruimtes (MSR; zie de foto's rechts in figuur 1). Hieraan zullen we verder geen aandacht besteden.

We richten ons verder op de routes in het middenspanningsnet. Routes kunnen naar woonwijken of naar grotere klanten gaan. Binnen woonwijken wordt verbruikt, maar ook opgewekt door bijvoorbeeld zonnepanelen. Onder grotere klanten bevinden zich bedrijven en kunnen ook zonneweides en windmolenparken zitten. Het kan voorkomen dat er binnen een route tegelijkertijd verbruik en opwekking plaatsvindt. Uit metingen bij een middenspanningsruimte en bij onderstations valt dit niet op te maken. Juist daar ligt de uitdaging.

1.2 Rekenen aan het net

Binnen het midden- en laagspanningsnet zijn er momenteel twee plaatsen met voldoende metingen: bij de onderstations en bij de klanten. Deze twee meetplaatsen geven twee manieren voor het berekenen van de totale stroom door een route, respectievelijk top-down en bottom-up.

Bij bottom-uprekenen worden de verbruiksmetingen van de klanten gebruikt. Bij veel klanten gaat dit om metingen van de levering en teruglevering in een jaar (in kWh). Met informatie over wat voor een type klant het betreft wordt een profiel gekozen dat daarbij past. Het verbruikspatroon van het profiel wordt met het totaalverbruik geschaald om een inschatting te maken van het verbruik over de tijd. Voor hogere niveaus van meetpunten, zoals bij onder-, regel- en schakelstations, worden deze inschattingen opgeteld. Deze gegevens zitten in de profielendataset die in sectie 1.3.2 wordt behandeld. [2]

Bij top-downrekenen kijken we naar de metingen bij de onderstations bij Alliander zelf. Het energiemanagementsysteem (EMS) verricht stroom- en vermogensmetingen aan de afgaande

velden. Elke route kan meerdere afgaande velden hebben. Door de metingen van elk afgaand veld bij elkaar op te tellen, komen we op een totaalverbruik uit.

Deze twee totalen zouden overeen moeten komen. Helaas lopen we hier tegen problemen in beide metingen aan. Niet alle klanten leveren metingen aan, waardoor de data verrijkt moet worden met inschattingen voor het gehele verbruik in de route. Daarnaast zijn de metingen die aangeleverd worden niet altijd te vertrouwen, bijvoorbeeld door fouten in het documenteren van waar in het stroomnet de meting zich precies bevindt, of door het niet doorgeven van verschakelingen. Bij de EMS-metingen zijn de groottes van de metingen op zich wel betrouwbaar, al kan het zijn dat er een meter verkeerd om zit aangesloten waardoor het teken verkeerd is, of verschakelingen niet correct worden gedocumenteerd. Een nadeel van deze EMS-gegevens is dat informatie over de stromen lager in het netwerk niet zichtbaar is.

1.3 Beschikbare gegevens

Voor dit onderzoek hebben we meerdere datasets tot onze beschikking gekregen. Een deel gaat over de topologie van het net en de rest bevat metingen bij onderstations en profielendata. Later hebben we deze gegevens ook verrijkt met weergegevens van het KNMI.

1.3.1 EMS-data

De EMS-dataset beschrijft metingen per afgaand veld. Deze gegevens zijn afkomstig uit het energiemanagementsysteem (EMS) van Alliander. Hier en daar noemen we deze gegevens ook wel PQI-data, vernoemd naar de gemeten waarden: actief vermogen P , reactief vermogen Q en stroom I . Een route kan meerdere afgaande velden hebben. De metingen in deze dataset zijn gedaan door Alliander en zijn vrij precies. Wel kunnen er systematische fouten in de richting zitten door meetapparatuur die verkeerd om aangesloten zit. Als een route wordt gevoed door meerdere kabels dienen de metingen op die kabels opgeteld te worden. De dataset wordt omgeschreven in tabel 1.

Bij I gaat het om de fasestroom I_f , de stroom door een enkele fase. Het totaal heeft drie fasen op afstand 120° . Het schijnbare vermogen is wat er uiteindelijk bij de onderstations gemeten wordt. Dit vermogen bestaat uit een actief (P) en reactief (Q) gedeelte. Het actieve vermogen is wat er daadwerkelijk verbruikt wordt. Het reactieve vermogen is energie die niet wordt omgezet en later weer terugstroomt. Dit wordt ook wel blindvermogen genoemd.¹ Om vanuit de fasestroom het schijnbare vermogen S uit te rekenen, geldt het verband in vergelijking (1).

$$S = 3U_{f0}I_f = U_{ff}I_f\sqrt{3} \quad (1)$$

`ems_pqi_${YYYYMM}.rds` bevat per 5 minuten de stroom I, actief vermogen P en reactief vermogen Q voor meetpunt `TA_NAME` op tijdstip `M_TIMESTAMP`.

1.3.2 Profielendata

De route_profielen-dataset bevat voor elke route `ASSETID` per kwartier een schatting van het vermogen in kW. Deze schatting is gebaseerd op metingen bij de klanten. Vaak gaat dit

¹Voor de kenners: het verband tussen de typen vermogens wordt gegeven door $S = UI = \sqrt{P^2 + Q^2}$.

²De `M_TIMESTAMP` is een SQL `DATETIME` geconverteerd naar een `string`.

Kolom	Datatype Python 3	Datatype R	Eenheid	Voorbeeldelement
ROUTE_NAAM	string	chr	-	ALK 10-1V151
TA_NAME	string	chr	-	Alk 10 V151
M_TIMESTAMP	string ²	chr	-	2018-05-01 00:05:00.000000000
I	float64	num	A	19.2
P	float64	num	MW	0.36
Q	float64	num	MVAr	0.19

Tabel 1: De kolommen in de EMS-dataset hebben een datatype meegekregen. Waar aanwezig is ook de eenheid meegegeven. Het voorbeeldelement geeft de opmaak van de kolom weer.

om jaarverbruiken en is extra informatie nodig om hiervan vermogens over de tijd te krijgen. Er wordt een inschatting gemaakt van het verbruik of teruglevering gebaseerd op het type klant. Grootte van huishoudens en aanwezigheid van zonnepanelen kunnen hierin meespelen. De dataset wordt omschreven in tabel 2.

Kolom	Datatype Python 3	Datatype R	Eenheid	Voorbeeldelement
ASSETTYPE	string	chr	-	ROUTE
ASSETID	string	chr	-	SCH 10-2V180
DATUM	float64	Date	-	2018-05-01
0000, 0015, ..., 2345	float64	num	kW	610
UPLOADDATUM	category	-	-	-

Tabel 2: De kolommen in de profielendataset hebben een datatype meegekregen. Waar aanwezig is ook de eenheid meegegeven. Het voorbeeldelement geeft de opmaak van de kolom weer.

1.3.3 Routemapdata

De routemapdataset is een koppeltabel tussen de afgaande velden zoals omschreven in de EMS-data en de complete routes zoals omschreven in de profielendata. De koppeltabel bevat verder ook de bedrijfsspanning van elke route, welke kan afwijken van de werkelijke spanning door spanningsverval over de kabels en belasting op de route. De bedrijfsspanning is de spanning tussen twee fasen (U_{ff}). Ook het type station waarop de route is aangesloten wordt vermeld. Hieronder zijn de drie typen stations die in het netwerk voorkomen omschreven. De dataset wordt omschreven in tabel 3.

Kolom	Datatype Python 3	Datatype R	Eenheid	Voorbeeldelement
ROUTE_NAAM	string	chr	-	AAL 10-1V11
M_POINT	string	chr	-	01AOF
TA_ELEMENT	string	chr	-	I, P of Q
TA_NAME	string	chr	-	Aa1-RS 10-G V11
BEDRIJFSSPANNING	float64	chr	kV	10.5
STATION_OMSCHRIJVING	string	chr	-	Regelstation, Onderstation of Schakelstation

Tabel 3: De routemapdataset koppelt afgaande velden aan routes. Ook de bedrijfsspanning van de afgaande velden staat hierin, die weer gekoppeld kunnen worden aan gemeten stromen om vermogens uit te rekenen. In de kolom **TA_ELEMENT** staat welke meter er beschikbaar is. Voor elke meter is een aparte rij in deze tabel aanwezig.

2 Voorverwerking van data

Om een model effectief te doen werken, zonder dat het tegen al te veel eigenaardigheden in de dataset aanloopt, is enige voorverwerking nodig. Zo blijkt bijvoorbeeld dat data uit Amsterdam vrij onbetrouwbaar is, niet alle datasets dezelfde tijdzone gebruiken (UTC dan wel CET/CEST) en sommige routes leeg zijn, of zelfs lege namen hebben.

Om dit op te lossen is een script geschreven genaamd `defluff.py`. Aanvankelijk nam het script de aangeleverde databestanden in RDS-formaat, en voerde het een aantal voorverwerkingsprocedures uit. Vervolgens schreef het de gefilterde data naar bestanden in *Feather File Format*; Python/Pandas werkt sneller en gemakkelijker met die bestandsindeling dan met RDS.

Het `defluff.py` script bleek nogal veranderlijk, en aangezien het na elke verandering opnieuw een RDS-bestand moest inlezen (hetgeen wel twee minuten kon duren), werd voor het converteren van RDS naar feather, een nieuw script opgesteld: `rds2feather.py`. Dit werd een dataverwerkingsstap die nodig was vóór het uitvoeren van `defluff.py`.

Nog een voorverwerkingsstap was het toevoegen van laatste dagen van de maand. Deze laatste dagen staan verzameld in een apart bestand.³ Het script `emplume.py` is geschreven om aan feather-bestanden de laatste dag van de maand toe te voegen, voordat deze worden doorgestuurd naar `defluff.py`.

Omdat deze scripts los van elkaar zijn gemaakt, was het makkelijker om de datafiltering in `defluff.py` stapsgewijs op te bouwen. Scripts konden zo getest worden zonder keer op keer de volledige dataverwerking uit te voeren.

Om de drie voorverwerkingsstappen (converteren van RDS naar feather met `rds2feather.py`, toevoegen van laatste dagen van de maand met `emplume.py` en het filteren van de data met `defluff.py`) automatisch aan elkaar te lassen, en het mogelijk te maken dit proces parallel uit te voeren, werd het script `tapeduck.py` geschreven. Dit is als het ware een shell-script verkleed als Python-script, dat wil zeggen: in plaats van dat het Python-functies uit de andere scripts aanroept om verschillende elementen van het proces samen te voegen, roept het de commando's aan in de shell. Op deze manier kan *concurrency* worden geïmplementeerd middels het Unix-commando `xargs`. Er wordt dan geen gebruik gemaakt van Python-threads, die niet parallel aan elkaar kunnen lopen.

Alle voorverwerkingsscripts zijn zo geïmplementeerd dat zij geen data verwerken als wordt gedetecteerd dat er reeds een resultaat-bestand bestaat dat nieuwer is dan de invoerbestanden.⁴ Hierom is er een `-f` optie voor elk van deze scripts om de dataverwerking te forceren.

Alle scripts zijn nogal *verbose*. Ze schrijven berichten naar `stderr` onder andere opdat inzicht wordt verkregen over de looptijd van de verschillende fases van de voorverwerking. Typische uitvoer van `defluff.py` ziet eruit als volgt:

```
$ scripts/defluff.py -f -i ./data/input -o ./data/output 201801
Reading ./data/input/ems_pqi_201801.feather ...
Done reading 104972586 rows! (took 2.1400s)
Reading ./data/input/route_profielen_201801.feather ...
Done reading 12511600 rows! (took 0.0268s)
Converting M_TIMESTAMP to datetime...
```

³Het ophalen van de gegevens uit de database bleek gedaan te zijn door een zoekopdracht naar maanddagen 1 tot en met 30, waardoor voor sommige maanden de laatste dag ontbrak.

⁴Dit is analoog aan de werking van het Unix-commando `make`

```

Done! (took 2.9992s)
Converting M_TIMESTAMP to UTC...
Done! (took 1.9807s)
Removing NaT M_TIMESTAMPS (DST-transition hour)...
Done removing 0.0% of rows! (took 1.4932s)
Removing empty ROUTE_NAAM...
Done removing 26.7% of rows! (took 1.4676s)
Removing Amsterdam...
Done removing 8.4% of rows! (took 136.1522s)
Removing lonely routes without profiles...
Done removing 23.9% of PQI and 77.5% of profile rows! (took 3.7073s)
Writing filtered feather files...
Done! (took 3.0184s)

```

Vanaf het begin is er aandacht geweest voor het inzichtelijk maken en houden van de werking van deze diverse scripts, in het bijzonder door middel van `--help` opties. Een overzicht van de `--help` berichten voor elk van de vier hiervoor benoemde scripts is te vinden in bijlage A.

3 Modellen

Er is gekeken naar twee mogelijke invalshoeken: via machine learning of via een ouderwets, ambachtelijk model.

We hebben ingezet op het bepalen van de stroomrichting met de stroom I uit de EMS-dataset, omdat deze altijd voor handen ligt en deze het meeste zegt over de belasting van een station. De profielgegevens worden gebruikt om het model te verifiëren. Door te kijken naar daggemiddelden, gemiddelden op bepaalde momenten op een dag, maxima, minima, hellingen en spreiding in al deze variabelen, is geprobeerd een model te bouwen. Hiervoor is ook gekeken naar weergegevens zoals voor zonnepanelen en windmolens.

In het vervolg lichten we drie verschillende invalshoeken toe en beschrijven daarvan de resultaten. Ter illustratie zullen we vaak routes als voorbeeld gebruiken. In deze context komen vier grootheden regelmatig terug.

- Q_{zon}^n is het stralingsvermogen van de zon bij weerstationnummer n .
- P_{prof} is het vermogen uit de profielendataset.
- $\pm P_{ems}$ is het vermogen uit de EMS-dataset. De \pm geeft aan of het teken van deze data gedraaid is ($-$) of niet ($+$), naar aanleiding van de uitgevoerde tests.
- $\sqrt{3}VI_{ems}$ is het vermogen dat bepaald is uit de bedrijfsspanning uit de routemapdataset en de spanning uit de EMS-dataset volgens vergelijking (1).

3.1 Machine learning

Voor het opzetten van een model op basis van machine learning wilden we gebruikmaken van de Tensorflow-module. In eerste instantie lijkt toepassing van een *Hidden Markov Model* (HMM) toepasselijk. Standaardmodellen nemen echter de *Markov Assumption* aan die stelt dat de staat van het systeem op t_{i+1} alleen afhangt van van de staat op t_i en de waarnemingen

op dat moment[3]. Dit lijkt niet passend bij het probleem. Mogelijk zijn andere modellen wel geschikt (bijvoorbeeld *Interactive Higher Order Hidden Markov Models*).

Een andere mogelijkheid is het gebruikmaken van een neuraal netwerk. In het bijzonder zijn *recurrent neural networks* met LSTM (*long short term memory*) dan interessant. Een open vraagstuk is hoe de gegevens het beste gepresenteerd kunnen worden aan het netwerk en wat een goede keuze is voor de outputs. De data heeft in principe veel gecorreleerde gegevens. Zo kunnen de datapunten uit een route op verschillende manieren in de tijd gerelateerd zijn. Een mogelijkheid is om de data aan te bieden per route voor een hele maand. Dit maakt echter dat de dimensie van de input hoog is en er dan relatief weinig data beschikbaar is. (Er is wel data van meerdere maanden. De maand zou eventueel ook als variabele kunnen worden toegevoegd). Een andere optie is om het per week te presenteren (om de 15 minuten) met een extra variabele voor de dag en de dag van de week. Bij deze keuze verlies je wel informatie over welke routes bij elkaar horen voor verschillende weken. Dan is nog de vraag wat te kiezen is als output. Idealiter zou je voor een route alle data willen invoeren en dan op elk tijdstip het teken bepalen. Het lijkt mij echter realistischer om het netwerk te trainen om voor een klein aantal tijdstippen (bijvoorbeeld de pieken van de route waarvan je het teken wilt bepalen).

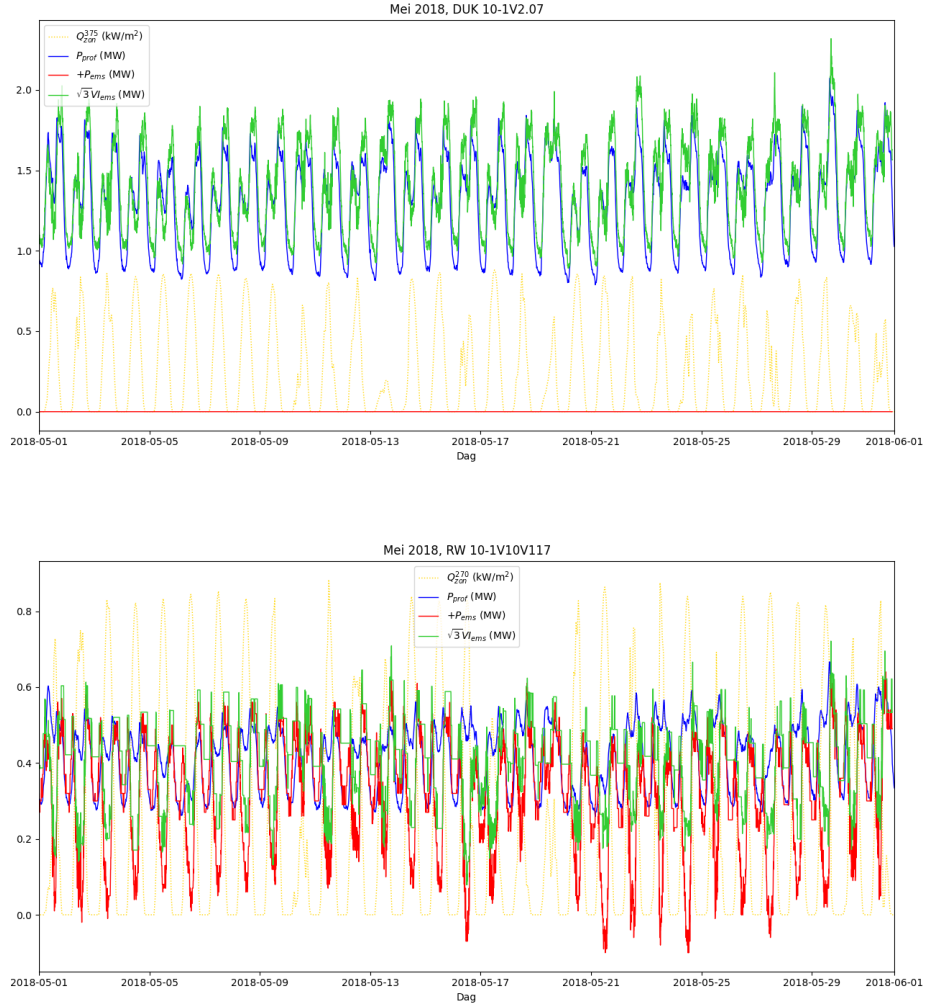
Afhankelijk van het type klanten dat aangesloten is aan een route zal die een bepaald gedrag vertonen. Het is daarom goed mogelijk dat er een (vooralsnog) onbekende set met features is in het gedrag van een netwerk die wijst op het type gebruikers in het netwerk. Daarom kan het interessant zijn om de verschillende routes te clusteren. Vervolgens kunnen de routes gelabeld worden per cluster en dit label kan als extra variabele aan een neuraal netwerk worden toegevoegd. Het probleem met deze aanpak is dat clusteren slecht tot niet mogelijk is in data met hoge dimensie. Daarom zal de data eerst moeten worden voorverwerkt om de dimensie te reduceren. Of en hoe dit mogelijk is, is nog niet duidelijk.

Daarnaast is er op een groot deel van de routes uitsluitend verbruik. Een model dat altijd gokt dat het teken positief is zal dus alsnog een hoge succesrate krijgen. Bij het trainen van welk model dan ook zal hier rekening mee moeten worden gehouden (bijvoorbeeld door *false positives* extra zwaar te wegen).

3.2 Absoluut lineair verband

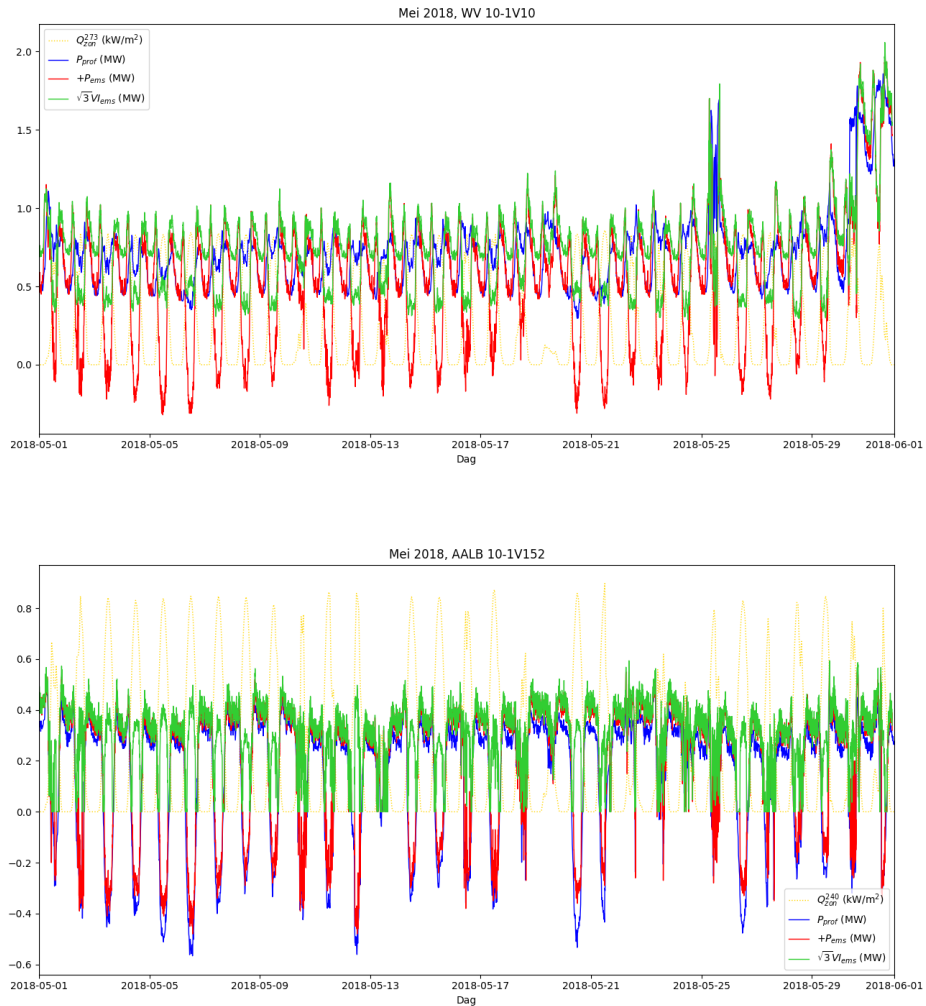
Na voor verschillende routes te hebben gekeken naar het gedrag van de stroommetingen gedurende de dag, werd duidelijk dat er bijna altijd overdag een daling is in de gemeten stroom. Onder de aanname dat dit veroorzaakt wordt door de opwek van zonne-energie gaan we kijken naar hoe deze verandering in de stroom overeenkomt met de eventuele zonneshijn. Hiervoor bekijken we als voornaamste parameter om de zonneshijn te kwantificeren naar de gemiddelde intensiteit van de zonnestraling over het afgelopen uur, Q , in kW/m^2 . Deze waarde wordt op (bijna) elk KNMI-station gemeten en wordt in die hoedanigheid vrijgegeven door het KNMI als open source data. Omdat de data van $I(t)$ echter een resolutie van vijf minuten hebben in plaats van een uur, gaat hier wel enige nauwkeurigheid mee verloren. Om de data van de stroommetingen compatibel te maken met de weergegevens van het KNMI wordt van de stroomdata een uurgemiddelde genomen, genoteerd met $I_h(t)$ waarbij $t \in 1, \dots, 24$. Hierbij staat bijvoorbeeld I_{16} voor het gemiddelde van de stroom van de tijdstippen 15:05 uur t/m 16:00 uur. Vervolgens is er voor gekozen om de afwijking van deze uurgemiddeldes ten opzichte van het ‘nachtgemiddelde’⁵ of ‘baseloa’ I_{night} te bekijken: $\Delta I(t) = I_h(t) - I_{\text{night}}$.

⁵De gemiddelde stroom tussen 01:00 uur en 05:00 uur, dus het gemiddelde van alle waardes I_2, \dots, I_5 over het gekozen tijdsinterval (bijvoorbeeld gedurende een maand).



Figuur 2: In de bovenste plot zijn er geen metingen van P_{ems} bekend. Wel is goed te zien dat zonne-energie een rol speelt in deze route. Op de plekken waarbij er een piek in het zonnevermogen te zien is, is een dal in zowel P_{prof} als $\sqrt{3}V_{I_{ems}}$ te zien. Dit maakt de aanwezigheid van zonnepanelen in de route aannemelijk. De daling van het verbruik bij stijging van het zonnevermogen impliceert dat we kijken naar verbruik en het teken van het verbruik goed staat.

De tweede grafiek laat ook een sterke rol van zonne-energie zien, op dezelfde manier. Wat opvallend is, is de lage resolutie van I_{ems} . Daarnaast is opmerkelijk dat P_{prof} een tekenverandering van $\sqrt{3}V_{I_{ems}}$ suggereert, terwijl P_{ems} daar geen aanleiding toe geeft. Aannemend dat het teken van P_{prof} accurater is, geeft dit aanwijzing tot een systematische meetfout in het teken van P_{ems} , dus een verkeerd aangesloten meter.



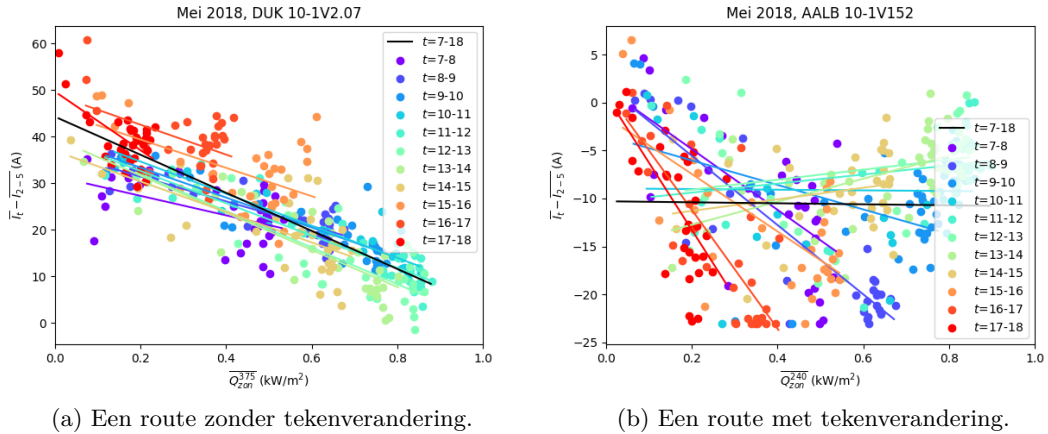
Figuur 3: De bovenste grafiek is vergelijkbaar met die van route **RW 10-1V10V117** in figuur 2. Volgens de EMS-data vindt hier teruglevering plaats, maar de profielendata suggereert verbruik.

Wat in de tweede grafiek opvalt, is dat er aanwijzingen zijn voor een tijdsafhankelijke tekenverandering van het vermogen. De groene lijn ($\sqrt{3}VI_{ems}$) en de rode lijn (P_{ems}) zijn elkaars gespiegelden in de tijdsas als het zonnevermogen erg hoog is. De route gaat hier over van netto verbruik naar netto teruglevering.

Voor een aantal verschillende routes zijn er scatterplots gemaakt waarin Q en ΔI tegen elkaar werden uitgezet. Al snel werd duidelijk dat voor routes waarvoor aan de hand van de dag-/maandprofielen te zien was dat er effecten van opwek van zonne-energie waren, dit verband evenredig te zijn met $-Q$. Daarom is er voor gekozen het volgende lineaire verband in Q te fitten:

$$\Delta I = aQ + b, \quad (2)$$

waarbij a dus negatief bleek te zijn. Er werd dus besloten om een test te introduceren die bekijkt hoe sterk dit effect is, en een route kan classificeren als een route waar danwel zonne-energie aanwezig is of niet sterk genoeg aanwezig is. Als testparameter werd de lineaire fitparameter a_h bekeken voor de waarden van gelijke uurgemiddeldes I_h , bijvoorbeeld alle waarden I_1 gedurende één maand en de overeenkomende waarden van Q op deze tijdstippen. Dit onder de aanname dat gedurende het bekeken tijdsinterval zowel zonnige als bewolktere dagen langskomen, en op deze manier het gedrag van ΔI als functie van Q bekeken kan worden doordat een reeks aan verschillende waarden van Q aan bod komen. Als de fitparameter a_h bijvoorbeeld voor alle bekeken tijdstippen $h = 7, \dots, 18$ negatief is, kun je inderdaad aannemen dat onder invloed van een toenemende intensiteit van de zonnestraling er meer zonne-energie wordt opgewekt en dus het stroomverbruik gedempt wordt. Mochten niet alle waarden van a_h negatief zijn, dan is er een extra test om te bekijken of het gemiddelde van deze waarden ‘negatief genoeg’ is. Dit is het geval als het gemiddelde van a_h kleiner is dan $-0.1I_{\text{night}}$. Dat betekent dat gedurende de dag de gemiddelde afname van het stroomverbruik bij een intensiteit van de zonnestraling van 1 kW/m² meer dan 10% van de baseload is ten opzichte van het stroomverbruik bij een intensiteit van de zonnestraling van 0 kW/m².



Figuur 4: Plots van het gefitte lineaire verband voor twee type routes waar zonne-opwek een rol speelt: een route zonder en met tekenverandering.

Als we echter wat verder kijken naar routes waarin zonne-energie een rol speelt, zien we dat er ook routes zijn waarvoor de hoeveelheid opgewekte elektriciteit door zonne-energie zo hoog is dat het vermogen daadwerkelijk van teken veranderd moet worden. Deze zonneprofielen zullen vervolgens niet meer goed benaderd worden door het lineaire verband doordat er een ‘knik’ in de scatter plot zit doordat de stroom altijd positief blijft. Omdat de toename van de stroom na de knik echter precies hetzelfde gedrag heeft als de afname van de stroom voor de knik, op een minteken na, kan dit waarschijnlijk goed benaderd worden door een zogenaamd ‘absoluut

lineair verband¹. Hiervoor gebruiken we de volgende fitfunctie:

$$\Delta I = c|Q - m| + d, \quad (3)$$

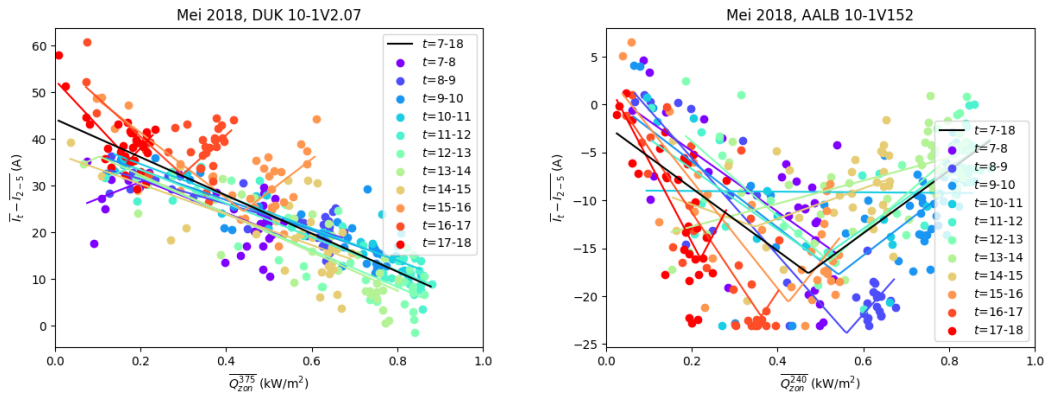
waarbij m een drempelwaarde voorstelt. Indien $Q > m$ neemt ΔI toe met een toenemende Q , als $Q < m$ neemt ΔI af met een toenemende Q . De waarde van de drempelwaarde m wordt bepaald door het minimum te vinden van een kwadratisch verband. Het kwadratische verband dat gefit wordt is

$$\Delta I^* = z_2(\Delta I - \min(\Delta I))^2 + z_1(\Delta I - \min(\Delta I)) + z_0, \quad (4)$$

omdat op deze manier een inschatting van m gevonden kan worden onafhankelijk van de fitparameters c en d . Het minimum van dit kwadratisch verband bevindt zich op

$$m^* = -\frac{z_1}{2z_2}. \quad (5)$$

Vervolgens maken we een lineaire fit in $|Q - m^*|$ om inschattingen c^* en d^* te vinden, welke we vervolgens gebruiken als een ‘educated guess’ om het absoluut lineaire verband met betere precisie te fitten. Op deze manier komen we uiteindelijk aan de parameters c , m en d .



(a) Een route zonder tekenverandering.

(b) Een route met tekenverandering.

Figuur 5: Plots van het gefitte absoluut lineaire verband voor twee type routes waar zonne-opwek een rol speelt: een route zonder en met tekenverandering.

Voor routes met een tekenverandering van het vermogen zou dit absoluut lineaire verband beter moeten werken dan het lineaire verband. Om dit te kwantificeren kijken we naar de gemiddelde afwijking van de punten in de scatterplot tot de fit. Voor vier voorbeeldroutes staan de fitparameters en deze gevonden ‘errors’ gegeven in Tabel 4. Hierin zien we dat voor route AALB 10-1V152 en WV 10-1V10 de gevonden fout kleiner is voor het absoluut lineaire verband dan voor het lineaire verband, dat deze voor RW 10-1V10V117 net iets hoger is en dat deze voor DUK 10-1V2.07 een stuk hoger is. Er is een kleine marge ingebouwd die de fout van het absoluut lineaire verband groter mag zijn dan die van het lineaire verband, zonder dat geconcludeerd wordt dat het lineaire verband beter werkt voor de gekozen route.

Als we naar deze resultaten kijken, zien we bovendien dat voor de route DUK 10-1V2.07 (figuren 2 en 5) de drempelwaarde m van het absoluut lineaire verband negatief is. De fit

Routenaam	Lineaire fit (LF), a, b	LF error	Absoluut lineaire fit (ALF), c, m, d	ALF error	Switch?
AALB 10-1V152	-0.519 , -10.288	1.618	32.676 , 0.472 , -17.613	0.767	True
RW 10-1V10V117	-22.689 , 9.282	0.747	24.148 , 0.789 , -9.295	0.882	True
WV 10-1V10	-44.188 , 18.447	2.923	52.522 , 0.724 , -17.125	1.414	True
DUK 10-1V2.07	-40.869 , 44.292	2.218	-40.869 , -1.721 , 114.628	5.287	False

Tabel 4: Enkele voorbeeldroutes met de fitparameters voor de gehele dataset voor de maand mei uit 2018 en de gemiddelde afwijkingen van de data t.o.v. deze fit ('error'). De kolom 'Switch?' geeft de waarde aan van de modelparameter die bepaalt of vervolgens een tekenverandering nodig is. Voor alle routes werd geconcludeerd dat er zonne-opwek aanwezig was.

'werkt' desalniettemin goed omdat ook c een tekenverandering heeft doorgaan. Daarom is er ook een test ingebouwd die kijkt of m een realistische waarde heeft (namelijk tussen 0 en 1, anders ligt de threshold voor tekenverandering namelijk buiten het spectrum waarin Q zijn waardes aanneemt) vooraleer wordt geconcludeerd dat het absoluut lineaire verband een betere fit is dan het lineaire verband en er dus een tekenverandering plaats moet vinden.

Let wel dat er uit dit model geen definitieve waarde van m gevonden gaat worden voor de omslag van verbruik naar opbrengst. De drempelwaarde is slechts een hulpmiddel om te bepalen of er een knik in de lineaire fit zit. Een model dat hiervoor een betere inschatting kan maken, kan wel veel waard zijn.

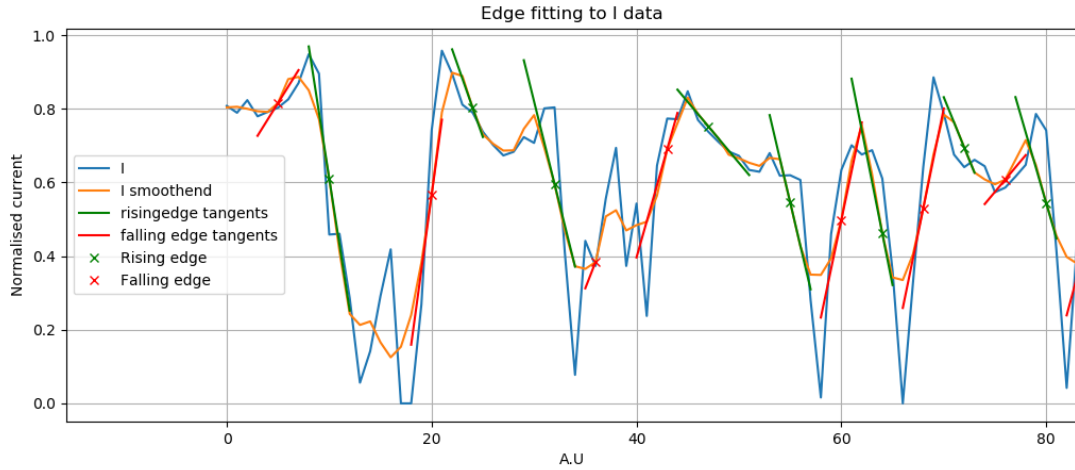
3.3 Rising/falling edges

Het leeuwendeel van de (niet direct bemeterde) opwek door klanten is zonne-opwek. Daarom zal er op de meeste routes een verband bestaan tussen de zonnekracht en het verbruik. De *rising/falling edge*-methode poogt dit verband te achterhalen en het teken te bepalen. Het achterliggende idee is als volgt.

De klanten hebben een (zonnekracht ongerelateerd) verbruik. Daarnaast is er ook een zonne-opwek die bij benadering recht evenredig is met de zonnekracht. Wanneer er een netto verbruik is op een route en de zonnekracht toeneemt zal door de toegenomen opwek het verbruik en daarmee de stroomsterkte afnemen. Als daarentegen er een netto teruglevering is, zal een toename in zonnekracht resulteren in meer teruglevering en zo een toename in de stroomsterkte. Om te bepalen of een dergelijk verband tussen de stroomsterkte en de zonnekracht bestaat, wordt de data gereduceerd tot *rising* en *falling edges* en vastgesteld of er *rising/falling edges* gelijktijdig optreden in beide datasets. Deze methode hanteert de volgende aannames.

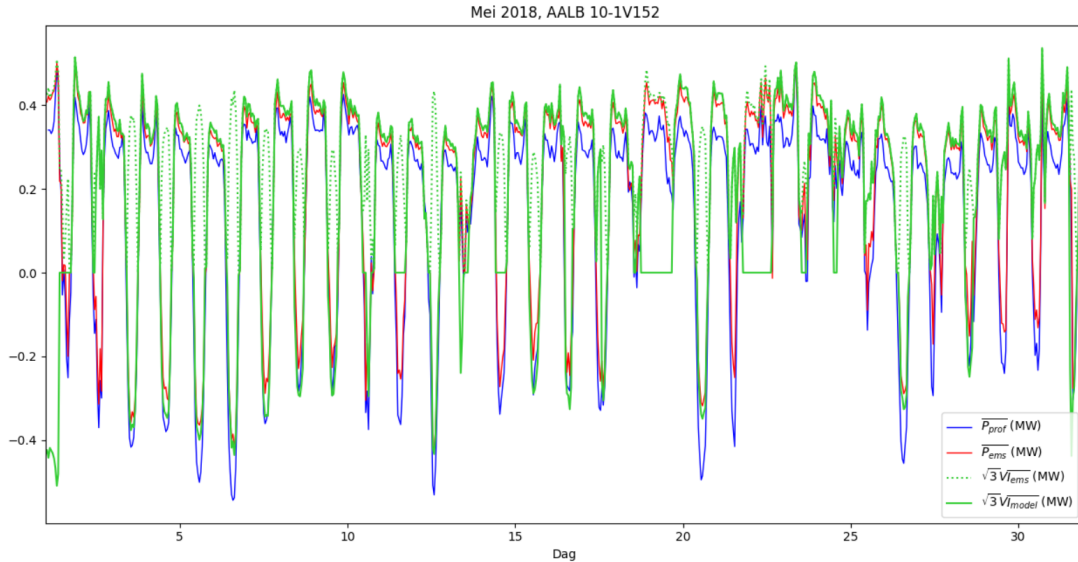
- alle opwek is zonne-opwek
- de opwek is recht evenredig aan de zonnekracht
- het verbruik is onafhankelijk van de zonnekracht

Voor onze implementatie van het model wordt de zonnekracht zoals gemeten door het dichtst bijzijnde KNMI-station genomen. Dit zijn uurgemiddelden. Vervolgens wordt ook van de stroomsterkte het uurgemiddelde genomen en een Savitsky-Golay-filter van orde 3 toegepast. Dit resulteert in een gladde curve die meer geschikt is om edges aan te fitten en verwijdert irrelevante hoogfrequente informatie. Vervolgens worden aan beide datasets *rising* en *falling edges* gefit. Het fitten gebeurt door op de data een convolutie uit te voeren met een genormaliseerd lijnstuk



Figuur 6: Voor het bepalen van de *falling/rising edges* wordt de grafiek van de ruwe gegevens I glad gemaakt. Hierdoor verdwijnt een hoop ruis. De raaklijnen op gedetecteerde *edges* zijn ingetekend en worden gekenmerkt als *rising* of *falling*.

(normalisatie betekent hier dat de minimale/maximale van de convolutie ± 1 is). Een voorbeeld van de convolutie en het plaatsen van de raaklijnen is te zien in figuur 6. Van deze convolutie worden de pieken bepaald met de *scipy*-methode `signal.find_peaks`. Deze pieken corresponderen met de middens van de *edges*. Vervolgens worden op deze plekken raaklijnen gefit. De raaklijnen worden omgezet tot *rising* en *falling edges* door ze in te korten tot de maximale lengte waarbij ze binnen een zekere afstand van de originele data blijven. Vervolgens wordt bepaald waar de stroomsterkte mogelijk door nul kan gaan. Deze punten worden bepaald door met `signal.find_peaks` de minima in de stroomsterkte te zoeken en daaruit punten te selecteren die voldoende dicht bij nul komen. Nu wordt de data opgebroken in stukken ingesloten tussen twee nulpunten en aangenomen dat het teken daar overal gelijk is. Het teken van een dergelijk stuk wordt bepaald door te kijken naar de overeenkomst met de tussen de nulpunten gelegen *edges* van de datasets. Er wordt uitgerekend hoeveel procent van de *rising edges* in de zonnedata overeenkomt met de *falling edges* van de stroomsterkte en andersom. Een hoog percentage impliceert dat er netto verbruik is. Er wordt ook uitgerekend hoeveel procent van de *rising edges* in de zonnedata overeenkomt met *rising edges* in de stroomsterkte en hetzelfde voor de *falling edges*. Een hoog percentage hier impliceert netto teruglevering. Het teken met het hoogste percentage overeenkomstige *edges* wordt toegekend aan het gebied. Bij gelijke percentages wordt geen conclusie getrokken. Een voorbeeld van de toepassing van dit model is te zien in figuur 7.



Figuur 7: In de grafiek is het resultaat van het *rising/falling edges*-model te zien, toegepast op route **AALB 10-1V152**. Deze route kent tijdsafhankelijke tekenwisseling door over te gaan van verbruik naar teruglevering bij een hoog zonnevermogen. Op een groot deel van de grafiek werkt de tekenbepaling van I_{model} goed en komt het gemodelleerde vermogen overeen met P_{prof} . Op enkele plaatsen was er geen conclusie en ligt de groene lijn op nul.

4 Conclusie en discussie

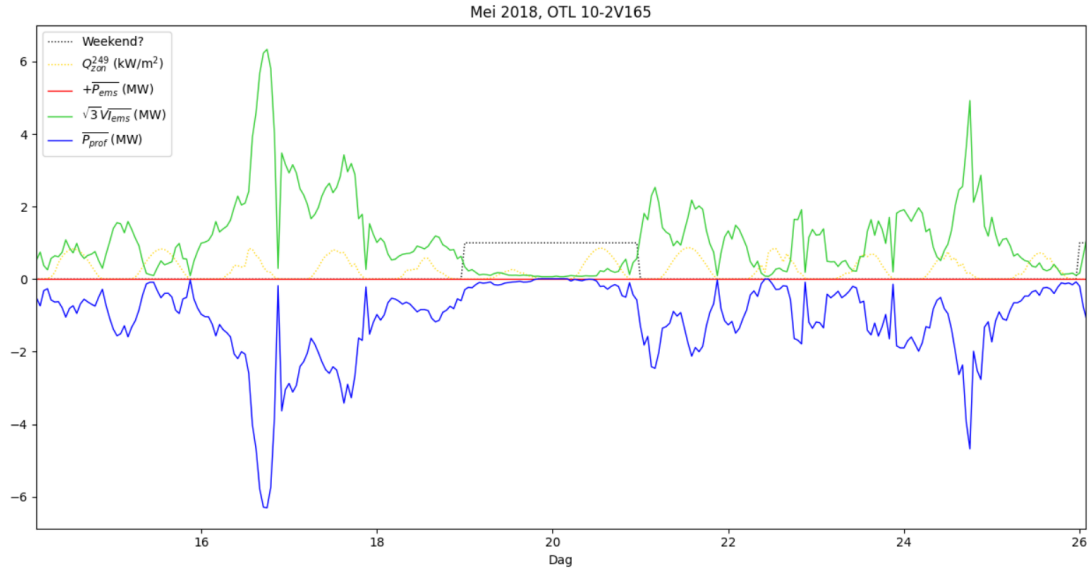
Al de hier ontwikkelde modellen zijn vooral gebaseerd op effecten door zonne-energie. Echter zijn er een hoop typen routes waarover we niet veel kunnen zeggen met deze modellen. De modellen kunnen daarbij wel iets zeggen over de significantie van de resultaten en zo een onderscheid maken tussen routes waarop de modellen wel werken en routes waar op een andere manier naar gekeken moet worden. Zelfs zonder profieleninformatie kan een inschatting gemaakt worden van de stroomrichting door variatie in de zonneshijn.

Daarnaast is het goed om te realiseren dat er mogelijk ook andere effecten zijn die verbruik veranderen bij grote zonneshijn, zoals hoger verbruik door airconditioning en mogelijk lager verbruik door mensen die naar buiten gaan.

Enkele soorten routes die nog interessant zijn om te onderzoeken, zijn routes met enkel teruglevering en routes met een warmte-krachtkoppelinginstallatie.

4.1 Windenergie

Bij een willekeurige route ligt verbruik meer voor de hand dan opbrengst. Als een route een stroomverbruik heeft dat strikt boven nul ligt, zal al snel gedacht worden dat dit verbruik betreft. Echter kan het voorkomen dat dit helemaal niet zo is. Zo bestaan er zonneweides en windmolenparken die strikt opwekken en niet verbruiken. Een voorbeeld hiervan is route OTL 10-2V165 zoals te zien is in figuur 8. Met de modellen van dit project worden dit soort routes niet herkend. Informatie over het profiel van de route is hier onmisbaar voor een juiste inschatting van het teken.



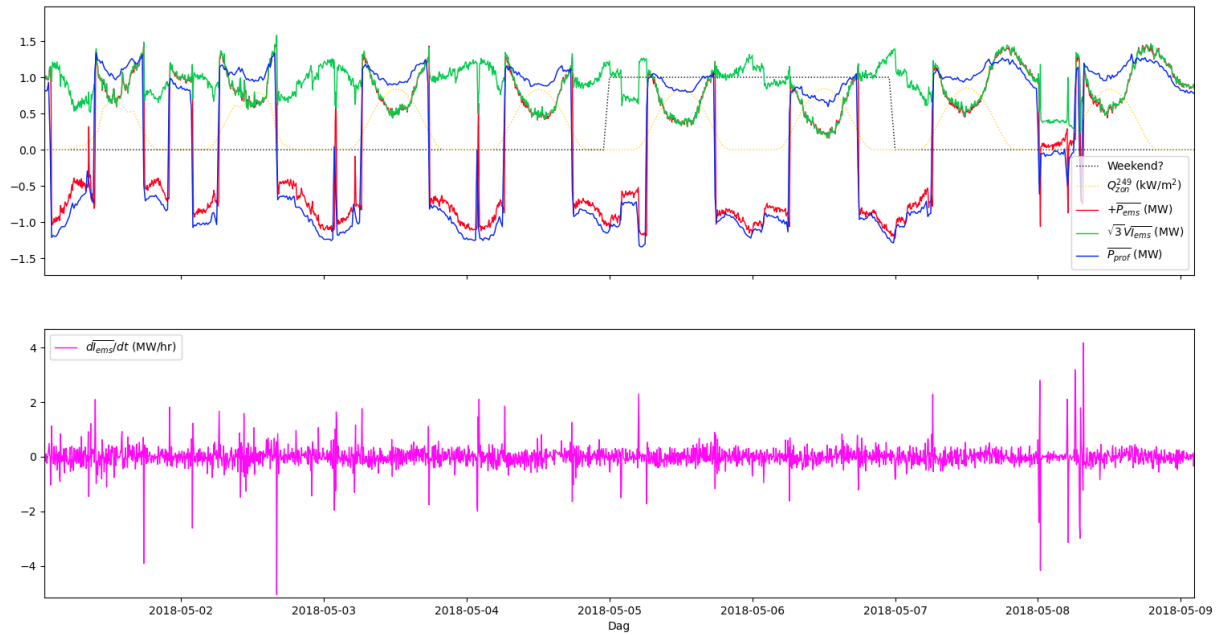
Figuur 8: Route **OTL 10-2V165** is een aansluiting op een windmolenpark. Naast de vermogensmetingen en -berekeningen, wordt ook het zonnevermogen per oppervlakte-eenheid voor het meest dichtbijgelegen meetstation afgebeeld. Zonder informatie over het profiel van deze route, ligt het voor de hand ervan uit te gaan dat deze route alleen verbruikt, waar het tegendeel waar is.

4.2 Warmte-krachtkoppelingsinstallaties

In figuur 9 worden de data van route OTL 10-2V160 in mei 2018 weergegeven. Het onderste deel van de figuur geeft de sterke hellingen van de stroom weer. Het gaat hier vermoedelijk om een warmte-krachtkoppelingsinstallatie (WKK-installatie). Deze installaties zetten warmte van verbranding van brandstof om naar elektrische energie. Bij lage brandstofprijzen of juist hoge elektriciteitsprijzen kunnen klanten deze plots aanzetten, wat leidt tot een sterke helling in de stroomgrafiek. Dit kan niet alleen worden gebruikt voor het verminderen van het stroomgebruik, maar ook voor het terugleveren naar het net. Het vergelijken van gegevens over de energieprijzen met de stroomsterktes van zulke routes kan informatie over de richting van de stroom leveren. Dit is een mogelijk aanknopingspunt voor een later model.

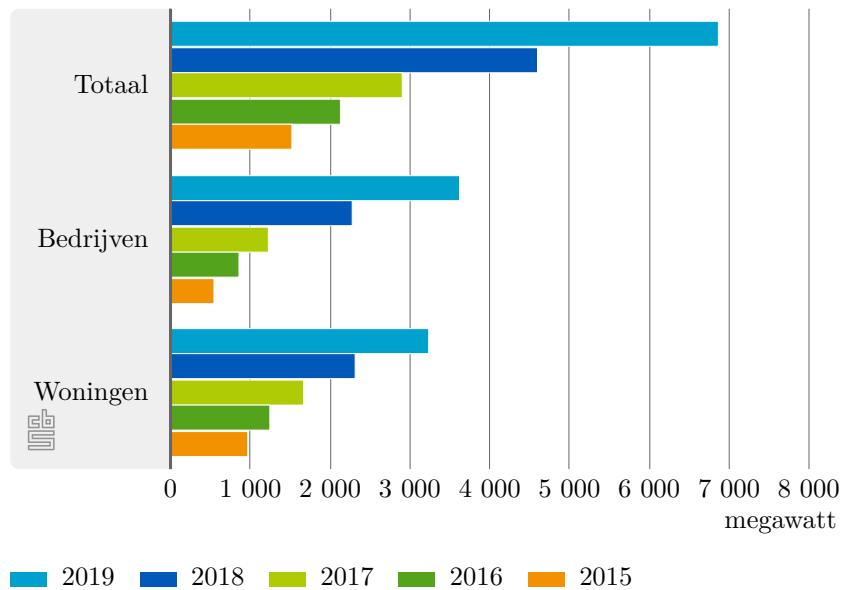
4.3 Verbeteren zonne-energiemodel

Naast het kijken naar andere soorten routes, zijn verdere ontwikkelingen in modellen voor routes met zonne-energieopwek veel waard. In figuur 10 is een stijging van het zonnevermogen in Nederland te zien. Voor de relevantie van de hier besproken modellen is niet alleen een stijging in deze opwek nodig, maar is het meer waard als deze opwek verdeeld (over routes) plaatsvindt. Het Algemeen Dagblad meldde in 2019 dat er 600 duizend huishoudens zijn met zonnepanelen op hun dak.[4] Met de ongeveer 8 miljoen huishoudens die Nederland telt, gaat dit om 7.5% van alle huishoudens.[5]



Figuur 9: Route **OTL 10-2V160** bevat waarschijnlijk een warmte-kranchkoppelingsinstallatie. De sterke hellingen geven aanwijzingen voor plotselinge schakelingen van zo een installatie.

Opgesteld vermogen zonnepanelen



Figuur 10: Het Centraal Bureau voor de Statistiek (CBS) meldt een gestage stijging van het opgestelde vermogen van zonne-energie in Nederland.[6]

Mogelijkheden voor het verbeteren van de modellen zijn het categoriseren van de gegevens op basis van of het een weekday of een weekenddag betreft en het in acht nemen van het bij Alliander bekende vermogen aan zonnepanelen in een route.

4.4 Machine Learning

Er is gekeken naar het opzetten van een model voor de stroomrichting op basis van machine learning. Dit model heeft geen resultaten opgeleverd. We vermoeden dat een model op basis van machine learning verder niet-triviaal is om op te zetten. Reden hiervoor is dat het gevaar bestaat dat een model altijd op verbruik zal gokken. Veel routes blijven een groot van de tijd verbruiken. Zo zal een model dat altijd gokt op verbruik toch in een groot deel van de tijd gelijk krijgen. Hierom zal zeker bij een machinaal getraind model een hoog succespercentage geëist moeten worden en moet er kritisch naar het getrainde netwerk gekeken worden.

Referenties

- [1] van Braak M, Toonen J, Heres J. Modellenpracticum Bepalen stroomrichting (intakeslides); 2020.
- [2] Valckx T. Stochastic effects of customers on peak loads in a power distribution grid [Ph.D. thesis]; 2019.
- [3] Marti DJJH. Hidden Markov Models. In: Speech and Language Processing; 2019. Available from: <https://web.stanford.edu/~jurafsky/slp3/A.pdf>.
- [4] Algemeen Dagblad. ‘Zonnepanelen worden nu echt zichtbaar in de samenleving’; 2019. [online; geraadpleegd op 31-07-2020]. <https://www.ad.nl/wonen/zonnepanelen-worden-nu-echt-zichtbaar-in-de-samenleving~aab95be3/>.
- [5] Centraal Bureau voor de Statistiek. Huishoudens; samenstelling, grootte, regio, 1 januari; 2019. [online; geraadpleegd op 01-08-2020]. <https://opendata.cbs.nl/statline/#/CBS/nl/dataset/71486NED/table?fromstatweb>.
- [6] Centraal Bureau voor de Statistiek. Verbruik hernieuwbare energie met 16 procent gegroeid; 2020. [online; geraadpleegd op 01-08-2020]. <https://www.cbs.nl/nl-nl/nieuws/2020/22/verbruik-hernieuwbare-energie-met-16-procent-gegroeid>.

A --help for various commands

A.1 model.py

```
usage: model.py [-h] [-d DATA_DIR] [--plot-lin] [--plot-abs] [--skip-plots]
               [--hour-average]
               ROUTE MONTH
```

Analyzes the direction of currents of a medium voltage network route using a rising/falling edges analysis and generates plots to get insight into the route.

positional arguments:

ROUTE	name of route to analyze, 'random' for a random route
MONTH	month for which to apply the analysis, e.g. 04 for April

optional arguments:

-h, --help	show this help message and exit
-d DATA_DIR	directory with the feather data files
--plot-lin	plot the fit of the linear correlation
--plot-abs	plot the fit of the absolute linear correlation
--skip-plots	skip plotting the models
--hour-average	plot hour averages instead of 15 min values

example:

```
For analysis of route 'AALB 10-1V152' in May, use:
./model.py 'AALB 10-1V152' 05
```

N.B.: This script currently expects data from 2018 with certain file names.

A.2 rds2feather.py

```
usage: rds2feather.py [-h] [-o OUT_DIR] [-f] FILES [FILES ...]
```

Convert RDS file(s) to feather

positional arguments:

FILES	RDS files to convert
-------	----------------------

optional arguments:

-h, --help	show this help message and exit
-o OUT_DIR	feather file output directory
-f	convert even if destination file is newer than source

A.3 emplume.py

```
usage: emplume.py [-h] [-o OUT_DIR] [-i IN_DIR] [-f] YEARS [YEARS ...]
```

Augment monthly data for given year with additional data

positional arguments:

YEARS	YYYY input feather dates
-------	--------------------------

optional arguments:

-h, --help show this help message and exit
-o OUT_DIR feather file output directory
-i IN_DIR feather file input directory
-f augment even if destination file is newer than source

A.4 defluff.py

usage: defluff.py [-h] [-o OUT_DIR] [-i IN_DIR] [-f] [--keep-ams]
 [--keep-timezone]
 DATES [DATES ...]

Take feather dataset, convert timestamps to UTC, remove Amsterdam and remove other fluff

positional arguments:

DATES YYYYMM input feather dates

optional arguments:

-h, --help show this help message and exit
-o OUT_DIR output directory
-i IN_DIR feather file input directory
-f defluff even if destination file is newer than source
--keep-ams keep Amsterdam in dataset
--keep-timezone don't convert timestamps to UTC

A.5 tapeduck.py

usage: tapeduck.py [-h] -i IN_DIR -o OUT_DIR [-P MAX_PROCS] [-f]
 [--skip PROCS]
 DATE_RANGE

Perform 'rds2feather | emplume | defluff' for input data within given date range

positional arguments:

DATE_RANGE YYYYMM-YYYYMM data range

optional arguments:

-h, --help show this help message and exit
-i IN_DIR RDS file input directory
-o OUT_DIR feather file output directory
-P MAX_PROCS, --max-procs MAX_PROCS
run at most MAX_PROCS processes at a time
-f pass -f flag to subprocesses
--skip PROCS skip given process(es) (i.e.
rds2feather, emplume, defluff)