



NVE

Reguleringsmyndigheten
för energi – RME

RME EKSTERN RAPPORT

Nr. 4/2022

.....

Establishing nodes in the distribution grid

.....

THEMA Consulting Group og Expert Analytics



RME Ekstern rapport nr. 4/2022

Establishing nodes in the distribution grid

Utgitt av: Reguleringsmyndigheten for energi
Forfatter: THEMA Consulting Group og Expert Analytics
Forsidefoto: iStock

ISBN: 978-82-410-2193-0
ISSN: 2535-8243
Saksnummer: 202205739

Sammendrag: I denne rapporten presenterer THEMA og Expert Analytics sitt arbeid med å identifisere og modifisere algoritmer for å samle forbruk i klynger eller til fiktive noder. Dette er relevant i utviklingen av nye oppgavevariabler, herunder effekt- og energidistanse. Konsulentene ser på tre hovedtyper av klyngealgoritmer: K-means, Gaussian Mixture Models og DBSCAN. Algoritmene kan i større eller mindre grad tilpasses og modifiseres. Algoritmene scorer ulikt langs ulike kriterier som eksogenitet, kompleksitet og fleksibilitet. Algoritmene er testet på faktiske data fra Elhub og fra et lite utvalg nettselskaper. Dette innebærer at konsulentene har beregnet effektdistanse basert på noder. Når det gjelder hvilken algoritme som er mest egnet for vårt bruk så vil dette være noe RME må vurdere nærmere.

Emneord: Effektivitetsanalyse, oppgavevariabler, noder, målepunkter, elhub

Reguleringsmyndigheten for energi
Middelthuns gate 29
Postboks 5091 Majorstuen
0301 Oslo

Telefon: 22 95 95 95
E-post: rme@nve.no
Internett: www.reguleringsmyndigheten.no

Mars, 2022

Forord

Reguleringsmyndigheten for energi (RME) regulerer nettselskapenes inntekter. Formålet er å bidra til effektiv drift, utnyttelse og utvikling av nettet. RME gjennomfører hvert år en effektivitetsanalyse som måler selskapene mot hverandre, og rangerer dem ut fra hvor mye ressurser de bruker på å bygge, drifte og vedlikeholde nettinfrastrukturen. Nettselskapenes avkastning bestemmes deretter av hvor kostnadseffektivt de løser sine oppgaver.

RME har i de senere årene utforsket nye oppgavevariabler i effektivitetsanalysen for distribusjonsnettet. Mulige nye oppgaver inkluderer effekt- og energiavstand, som er ment å skulle være et mer eksogent og representativt mål på oppgavene som nettselskapene utfører. Effekt- og energidistanse skal beskrive avstanden som kraften må transporteres for å nå den enkelte kunden.

Effekt- og energiavstand kan beregnes fra hvert innmatingspunkt og til hvert enkelt målepunkt hos sluttkunde. En slik tilnærming kan imidlertid være krevende når vi tar i betraktning at det er 3,2 millioner individuelle målepunkter i nettet. Et alternativ til å bruke individuelle målepunkter er å aggregere disse i klynger eller til virtuelle målepunkter.

RME har bedt THEMA og Expert Analytics om å analysere ulike algoritmer for å etablere klynger av målepunkter og deretter vurdere hvordan disse kan brukes i beregningen av en effektdistanse. Algoritmene som er analysert er K-means, Gaussian Mixture Models og DBSCAN. Algoritmene kan i større eller mindre grad tilpasses og modifiseres ut ifra brukerbehov. Algoritmene er testet ut på et datasett fra Elhub supplert med ytterligere nettdata fra enkelte nettselskaper.

Studien konkluderer med at det er gode argumenter for å bruke en klyngetilnærming i beregning av effekt- og energidistanse. Når det gjelder hvilken algoritme som er mest egnet må dette vurderes ut ifra kriteriene eksogenitet, kompleksitet og fleksibilitet. Dette må RME vurdere nærmere.

Alle vurderingene og konklusjonene i rapporten er konsulentenes egne.

Vi inviterer alle til å komme med innspill til arbeidet. Tilbakemeldinger merkes med referansenummer 202205739 og sendes til rme@nve.no innen 1. juni.

Oslo, mars 2022

Tore Langset
Direktør
Reguleringsmyndigheten for energi

Roar Amundsveen
Fung. seksjonssjef
Økonomisk regulering

Dokumentet sendes uten underskrift. Det er godkjent i henhold til interne rutiner.

Establishing nodes in the distribution grid





Project info

Client

Reguleringsmyndigheten for en-
ergi (RME)

Project number

RME-21-03

Project title

Metode for etablering av noder i
distribusjonsnettet

Report info

Report availability

Public

THEMA report number

2021-16

ISBN number

978-82-8368-098-0

Date of publication

8th December 2021

About the project

RME is considering whether to include a power distance variable as an output in their DEA benchmarking model for Norwegian electricity distribution grids. To reduce the computational complexity and improve exogeneity with respect to LVD/HVD lines, one option could be to aggregate individual metering points into virtual nodes. In this project we investigate three main types of algorithms for clustering metering points into nodes with emphasis on properties such as exogeneity, tuneability (flexibility), complexity and computational costs. We also discuss their possible applications for the computation of a power distance variable.

Project team

Project manager

Åsmund Jenssen

Contributors (alphabetically)

Trygve Bærland

Sigmund S. Kielland

Jonathan Feinberg

Lisa Zafoschnig

Haavard Holta

Executive Summary

The Norwegian Energy Regulatory Authority (RME) is responsible for the economic regulation of Norwegian electricity grid companies. A key element in the regulation is a cost norm determined by a DEA benchmarking model (Data Envelopment Analysis). In the distribution grid, the DEA model is designed to compare the performance of grid companies by benchmarking the total costs against a set of output parameters that serve as a proxy for the task of supplying electricity in the respective grid areas.

In recent years RME has investigated possible new output parameters in the DEA Model for the distribution grid. The current model uses the number of customers, the number of substations and length of lines in the high voltage distribution grid (>1 kV). Possible new outputs include the power and energy distance that are intended to serve as a more exogenous and representative measure of the tasks of grid companies by accounting for the distance over which power needs to be transferred to reach each customer. In these analyses several approaches have been used, including using the real grid and an artificial grid method based on metering data and geographical information on the grid. A drawback of these methods is that the power distance calculation needs to be applied to all metering points and substations to capture the total power distance in both the low voltage and high voltage distribution grid.

An alternative approach to using individual metering points is aggregation of metering points into clusters or virtual nodes in the grid. These clusters can contain information on e.g. distance to metering points from substations, installed capacity, annual and hourly demand and the number of metering points included in the clusters. Such an aggregation can enable a more efficient computation of the power distance and also improve the incentives by removing the bias towards the existing grid solutions (230V vs. 400V in the low voltage grid). On this background, RME has commissioned a study by THEMA and Expert Analytics

to investigate different clustering algorithms and their possible role in the computation of a power distance parameter.

A clustering analysis essentially involves the task of labeling objects in such a way that nodes with the same label (i.e. belong to the same cluster) are considered to be physically close to each other. Clustering algorithms can be distinguished by a number of secondary criteria. A key feature is whether the clusters are anchored. Cluster anchoring involves the allocation of a root node to each cluster. In our analysis, one possibility is to use the actual substations in the grid as the root node. Other distinguishing features can be whether the number of clusters is determined by the algorithm or not and the size of each cluster. For instance, cluster size can be linked to energy consumption or capacity. Finally, algorithms can differ with respect to the stability of the results, model complexity and computational cost.

We have considered a set of algorithms that differ along the dimensions described above. Specifically, we have used three basic types of algorithms with some variations: K-means, Gaussian Mixture Models and DBSCAN.

K-means is a simple method that is built on two principles: Each cluster is defined by its centroid, and each node belongs to the cluster which centroid is the closest. The clustering is then carried out through an iterative procedure.

The Gaussian Mixture Model (GMM) involves selecting a set of parameters defining an underlying probability distribution from the data available so that the joint probability density of the data is maximized, again utilizing an iterative procedure to arrive at the most feasible distribution where data to the largest extent possible are produced in high probability areas. GMM can be customized to include e.g. anchoring or weights to determine cluster size that are set outside the model.

Finally, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm builds on a principle where each node is characterized according to its surroundings and a set of user-defined parameters. For instance, the user can

define a threshold value for the Euclidian distance to identify whether a node is in the neighborhood of another node. This can then be used to identify core nodes that form a cluster together with nodes that are reachable from the core node, as well as outlier nodes.

To investigate the properties of the different algorithms we have used a dataset from Elhub with supplementary grid data from companies that have participated in the previous analysis of the power distance parameters. The Elhub data exhibited low quality with respect to geographical location, however, including missing or incorrect coordinates for the metering points. Several measures were taken to correct the data, in addition to pre-processing.

We have carried out two case studies to illustrate the properties of the algorithms: Klepp, using Elhub and actual grid data, and Mørenett, using Elhub data only. We have looked at parameters such as the number of clusters, line lengths and a power distance calculation using the artificial grid method developed in previous work. For both cases DBSCAN differs with respect to the number of clusters as it is the only method considered where this number is not a direct input parameter. DBSCAN tends to group data with a high density of nodes (metering points) into large clusters. DBSCAN also tends to give longer line lengths, in part due to the treatment of outlier nodes as separate clusters. This also results in the power distance being significantly longer with DBSCAN in both case studies. In the Klepp case, we find that the methods using anchoring yield power distances fairly close to the baseline in the actual grid (as we have data available to benchmark against the baseline). In the Mørenett case, which is more geographically complex, we find examples of grid lines crossing geographical obstacles such as water. A final observation is that the computational time of the power distance calculation favors the K-means method, as this method yields more evenly distributed cluster sizes.

Overall, we consider that there are strong arguments in favor of using clustering methods in the power distance calculations. Clustering yields a

higher degree of exogeneity compared to alternatives such as using the real grid or the simple hybrid approach investigated in a previous study where each metering point is allocated to the nearest existing substation. With clustering, the separation between the low voltage and high voltage distribution grid are decided by an algorithm rather than any decisions by the grid companies. Anchoring can be used as an intermediate solution between the real grid and the non-anchored methods (with some loss of exogeneity). On the question of which clustering algorithm to use, this should be done according to a closer consideration of the criteria of exogeneity, complexity and tuneability (or flexibility). Tuneability is needed to bridge the gap to the real grid using available information, but may on the other hand lead to overly complex models. The weighing of different criteria should be considered further by RME.

Sammendrag

Reguleringsmyndigheten for energi i NVE (RME) er ansvarlig for den økonomiske reguleringen av norske kraftnettselskaper. Et sentralt element i reguleringen er en kostnadsnorm som bestemmes ved hjelp av benchmarking i en DEA-modell (Data Envelopment Analysis). I distribusjonsnettet er DEA-modellen utformet for å sammenligne nettselskapenes prestasjoner ved å benchmarke totalkostnadene mot et sett av oppgavevariabler som til sammen skal utgjøre et mål på oppgaven med å forsyne ulike nettområder med elektrisitet.

I de senere årene har RME utforsket mulige nye oppgavevariabler i DEA-modellen for distribusjonsnettet. Oppgavene i den nåværende modellen er antall kunder, antall nettstasjoner og lengden på linjer og kabler i høyspent distribusjonsnett (>1 kV). Mulige nye oppgaver inkluderer effekt- og energigjavnstand, som er ment å skulle være et mer eksogent og representativt mål på oppgavene som nettselskapene utfører. Effekt- og energidistanse skal reflektere avstanden som kraften må transporteres for å nå den enkelte kunden. I disse analysene er flere metoder blitt benyttet, herunder å ta utgangspunkt i det faktiske nettet og metoder med syntetiske nett basert på måleverdier og geografisk informasjon om nettet. En ulempe med disse metodene er at beregningene av effektdistanse må gjøres for alle målepunkter og nettstasjoner for å fange opp den samlede effektdistansen i både lavspent og høyspent distribusjonsnett.

Et alternativ til å bruke individuelle målepunkter er å aggregere målepunktene i klynger eller virtuelle noder i distribusjonsnettet. Disse klyngene kan inneholde informasjon om eksempelvis avstand fra målepunkt til nettstasjon, installert kapasitet, forbruk på års- og timebasis og antall målepunkter som inngår i klyngene. Slik aggregering kan gjøre det mulig å beregne effektdistanse på en mer effektiv måte og også styrke incentivene ved å fjerne favoriseringen av eksisterende nettløsninger (230V vs. 400V i lavspentnettet). På denne bakgrunnen har RME bedt THEMA og Expert Analytics om å

analysere ulike algoritmer for å etablere klynger og vurdere hvordan de kan benyttes i beregningen av en effektdistansevariabel.

Analyse av klynger (clustering analysis) innebærer å merke objekter på en måte som gjør at punkter med samme etikett (det vil si at de tilhører den samme klyngen) vurderes å være fysisk nær hverandre. Klyngealgoritmer kan skilles fra hverandre gjennom flere sekundære kriterier. Et nøkkelspørsmål er forankring av klyngene. Forankring av klynger handler om å allokere rotnoder (root nodes) til hver klynge. I vår analyse er det mulig å bruke de faktiske nettstasjonene som rotnoder. Andre særtrekk ved algoritmene kan være hvorvidt antall klynger bestemmes av algoritmene eller ikke, og størrelsen på hver klynge. For eksempel kan klyngestørrelsen relateres til energiforbruk eller kapasitet. Endelig kan algoritmene skille seg fra hverandre med hensyn til stabiliteten i resultatene, modellkompleksitet og beregningskostnader.

Vi har vurdert et sett av algoritmer som er forskjellige langs dimensjonene vi beskrev ovenfor. Konkret har vi brukt tre hovedtyper av algoritmer med noen varianter: K-means, Gaussian Mixture Models and DBSCAN.

K-means er en enkel metode som bygger på to prinsipper: Hver klynge er definert ved sin sentroide (det geometriske senteret i klyngen), og hvert punkt tilhører klyngen som inneholder den nærmeste sentroiden. Konstruksjonen av klynger skjer deretter gjennom en iterativ prosedyre.

Gaussian Mixture Model (GMM) innebærer at man velger et sett av parametere som definerer en underliggende sannsynlighetsfordeling for de tilgjengelige dataene på en måte som maksimerer den samlede sannsynlighetstettheten til dataene. Dette skjer gjennom en iterativ prosedyre for å komme fram til den mest passende fordelingen der data i størst mulig grad produseres i områder med høy sannsynlighet. GMM kan skreddersys for å inkludere eksempelvis forankring eller vektorer for å bestemme klyngestørrelse som fastsettes utenfor modellen.

Endelig bygger DBSCAN (Density-Based Spa-

tial Clustering of Applications with Noise) på et prinsipp der hvert punkt karakteriseres ut fra omgivelsene og et sett av brukerdefinerte parametere. For eksempel kan brukeren definere en terskelverdi for den euklidske avstanden for å avgjøre hvorvidt et punkt er i nærheten av et annet punkt. Dette kan i sin tur brukes til å identifisere både kjernepunkter som utgjør en klynge sammen med punkter som kan nås fra kjernepunktet og ekstrepunkter (outliers).

For å utforske egenskapene til de forskjellige algoritmene har vi brukt et datasett fra Elhub og supplert med nettdata fra selskaper som har deltatt i tidligere analyser av effektdistansevariabelen. Elhubdataene har vist seg å ha lav kvalitet med hensyn til geografisk informasjon, herunder manglende eller feil koordinater for målepunktene. Flere grep ble tatt for å øke datakvaliteten, i tillegg til pre-prosessering av dataene.

Vi har gjennomført to casestudier for å illustrere virkemåten til algoritmene: Klepp Energi (Elhubdata og faktiske nettdata) og Mørenett (bare Elhubdata). Vi har sett på parameter som antall klynger og linjelengder, og vi har gjort en beregning av effektavstand ved hjelp av en metode basert på syntetiske nett som er utviklet i tidligere prosjekter. I begge tilfellene skiller DBSCAN seg ut med hensyn til antall klynger. Det skyldes at denne metoden er den eneste hvor antall klynger ikke er en direkte inputparameter. DBSCAN tenderer også til å gi lengre linjer, delvis fordi metoden behandler ekstrempunkter som separate klynger. Dette resulterer også i at effektavstanden blir vesentlig lengre med DBSCAN i begge casestudiene. I Klepp-caset finner vi at metodene med forankring gir effektavstander som er relativt nær referansen i det faktiske nettet (ettersom vi i dette caset har faktiske nettdata). I Mørenett-caset, som er mer komplisert geografisk, finner vi eksempler på at linjer krysser geografiske hindre som vann. En siste observasjon er at beregningstiden for effektavstand er kortest for K-means ettersom denne metoden gir en mer jevn fordeling av størrelsen på klyngene.

Samlet sett vurderer vi at det er sterke argumenter for å bruke klyngealgoritmer i beregningen

av effektavstand. Ved å bruke klyngealgoritmer får vi en høyere grad av eksogenitet sammenlignet med alternativer som å bruke det faktiske nettet eller den enkle hybride tilnærmingen i et tidligere prosjekt der hvert målepunkt ble tilordnet den nærmeste nettstasjonen. Med klyngealgoritmer bestemmes skillet mellom lavspent og høyspent distribusjonsnettet gjennom en algoritme i stedet for nettselskapenes beslutninger. Forankring kan brukes som en mellomløsning mellom det faktiske nettet og metodene uten forankring (dog med noe tap av eksogenitet). Når det gjelder spørsmålet om hvilken algoritme som bør brukes, bør det avgjøres ved en nærmere vurdering av kriteriene eksogenitet, kompleksitet og fleksibilitet. Fleksibilitet er nødvendig for å redusere avstanden til det faktiske nettet ved å bruke tilgjengelig informasjon, men kan på den andre siden føre til at modellene blir for komplekse. Vektingen av ulike kriterier bør vurderes nærmere av RME.

Contents

1. Introduction	1
1.1. Background and previous work	1
1.2. Contributions and report structure	2
1.3. The Norwegian power grid	3
2. Clustering algorithms	6
2.1. Preliminaries	6
2.2. Desired features and selection criteria	6
2.3. Overview of algorithms and recommendations	8
3. Data sources and data handling	15
3.1. Data needs for power distance computation	15
3.2. Data sources	15
3.3. Pre-processing of data	16
4. Results	18
4.1. Case 1: Klepp	18
4.2. Case 2: Mørenett	23
5. Discussions and recommendations	25
5.1. Interpretation of output results	25
5.2. Stability	25
5.3. Implications for the power distance variable	27
5.4. Other considerations	27
5.5. Recommendations	28
A. Acronyms	30
B. References	31



1. Introduction

In this chapter, we give a brief overview of previous work related to developing the new *power distance* output variable for benchmarking of grid companies, and the motivation for applying clustering algorithms to the low-voltage distribution (LVD) grid.

1.1. Background and previous work

The Norwegian Energy Regulatory Authority (Reguleringsmyndigheten for Energi (Regulatory authority for Energy, RME) within Norges Vassdrags- og Energidirektorat (Norwegian Water Resources and Energy Directorate, NVE), is responsible for regulating the Norwegian grid operators [1]. As power systems are changing with new technologies, more available data and different consumption patterns, RME aims to design a fair and future-proof regulation for stakeholders on all grid levels. In recent years, RME has put large efforts into updating and improving the income regulation for network companies in the distribution grid. Since 2018, RME has commissioned several studies to design new output parameters in the benchmarking of grid companies. Previous work has investigated possible ways of computing a so-called *power distance* parameter that accounts for the distance over which power needs to be transferred to customers. Several methods that relied on different input data were analyzed, ranging from the minimal power distance using the full grid system and hourly metering data to more simplified methods such as artificial grid methods that relied merely on metering data aggregated to yearly level.

In the first study on the minimal power distance in 2018 [2], computational methods were tested on exemplary test grids. It was concluded that

the computational complexity was too high for the method to be applied on real grid cases. In the subsequent study [3], alternative approaches to computing the power distance were proposed and tested on sub-grids of a selection of Norwegian grid companies. As a result of insufficient data quality on grid infrastructure the study recommended to move forward with an artificial grid method. Instead of using actual grid data from the high-voltage distribution (HVD) grid, the developed algorithm builds a synthetic radial grid based on the location and demand of substations. As a first step, the demand in the low-voltage distribution grid was aggregated to substation level by summing the hourly metering data at all associated metering points. Further work [4] on *grid-free* methods investigated alternative methods to construct a synthetic grid and proposed a method to account for demand distribution in the low-voltage distribution grid. In [4], it was highlighted that the low-voltage distribution grid should be considered in calculations of the power distance to avoid any bias or skewed incentives that favor the choice of 230 V lines. For combining all grid levels in the distribution network, several methods were proposed of which a cost-weighting of low-voltage and high-voltage distribution grid was considered the most promising. In parallel, work on streamlining and standardizing data sets needed for the power distance computation was performed by Multiconsult [5]. The project worked with hourly metering data and geographical data of four grid companies and included tasks such as linking addresses to geographical coordinates, combining metering data to asset metadata and handling of erroneous data entries.

As work on alternative output parameters progresses, it has become clear that data plays a



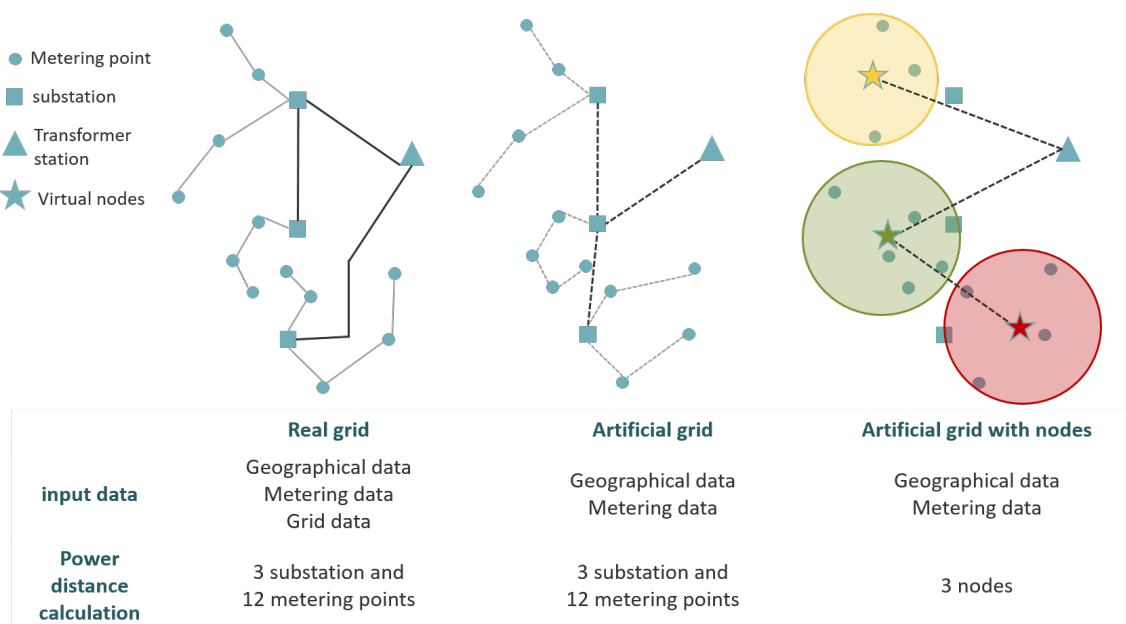


Figure 1.1.: Simplified illustration of different approaches to calculate the power distance on a test case.

crucial role in how the task of grid companies can be reflected. Several points remain to be investigated before any conclusion can be made about the applicability of a power and/or energy distance parameter in the Data Envelopment Analysis (DEA) model and the most efficient ways of handling data input. As the proposed methods have moved away from using real grid data, the accuracy of reflecting the distribution of demand through metering data becomes even more important and merits special attention.

1.2. Contributions and report structure

In RME's overarching ambition of designing a fair and future-proof income regulation for distribution system operators progress has been made along two main axes - defining computational methods and streamlining data handling. To compute the power power distance of a grid, three main data sets have been used:

- Metering data per metering point and/or sub-

station with an hourly resolution.

- Geographical information on the location of metering points, substations and transformer stations.
- The physical topology of the distribution grid, i.e. the links between metering points, substations and transformer stations.

Figure 1.1 is an illustration of past and expected progress towards a more efficient computational method for new power distance output variable. The initial work on defining new output variables focused on calculating the power distance based on available grid data and metering data, including geographical information and information about the grid real grid topology. This approach, illustrated on the left of Figure 1.1, required all three input data sets and, if applied to the entire distribution grid, would use metering points and substations in the calculation. By removing grid data as an input, the data requirements can be reduced. An artificial grid method as suggested in [3], shown in Figure 1.1 (middle), only requires metering data and geographical information. The drawback of using

both the real grid (left) and an artificial grid (right) is that to cover the full grid system a power distance calculation would need to be applied to all metering points and substations. Note that for simplicity the same artificial grid method is illustrated in both LVD and HVD grids in Figure 1.1. Previous studies also investigated the possibility of using different methods in the LVD and HVD grid.

In this report, commissioned by RME, THEMA and Expert Analytics investigate the possibility of aggregating metering points in clusters as shown on the right of Figure 1.1. The resulting virtual nodes defined by the clusters contain relevant information on associated metering points so that an artificial grid method applied between transformer stations and the virtual nodes accounts for the entire distribution grid. This has two main advantages. Firstly, the computation of power distance can potentially be simplified since the computational method only relies on data on substation level. The fact that the virtual nodes contain relevant information on the low-voltage distribution grid makes it possible to capture the entire value chain without additional weighting of grid levels. Secondly, the use of virtual nodes is more exogenous than the use of existing substations. The position of existing substations is determined by the choice of 230 V or 400 V lines in the low-voltage distribution grid and may benefit those grid companies with longer lines in the high-voltage distribution grid. By using virtual nodes that are based on underlying meter data, bias of existing infrastructure can be eliminated. The aggregated virtual nodes should contain information on

- number of metering points associated to each cluster by customer group
- statistical measures describing the distance to metering points
- sum of annual demand for metering points associated to each cluster
- sum of power limit for metering points associated to each cluster

It is important to note that any simplification

should not come at the expense of any bias in the final output. All proposed methods for data aggregation need to be analyzed with respect to their implications on the benchmarking process. We thus identify the motivation of this study to be developing methods to establish virtual nodes in the distribution grid that improve the computation of new output parameters without negatively affecting the fairness in the DEA model.

In the following sections, each of the bullet points will be discussed in the context of input data needs, algorithm features, and simulation results. In Chapter 2 we present the proposed clustering algorithms, including both technical descriptions and practical description of the features relevant for metering point clustering. Chapter 3 will give an overview of the input data sources, data quality, data requirements, and data pre-processing steps. In Chapter 4, the proposed algorithms are applied to real grid test cases provided by the data sources. Finally, Chapter 5 includes some final remarks on implications for the benchmarking process.

1.3. The Norwegian power grid

To describe the scope of the work presented in this report, it is necessary to provide a basic understanding of the power grid in Norway. Figure 1.2 shows a schematic of the grid levels in the electricity network and their exchange points. The grid levels marked in green are considered in this study. The power grid in Norway is structured in the following four levels [1]:

Transmission grid: The highest grid level, also referred to as grid level 1, typically operates at a voltage of 300 kV or 420 kV and connects producers and trade capacity with connection points to lower grid levels across the country.

Regional grid: The regional grid, grid level 2, operates at a voltage of 33 kV-132 kV and serves as an intermediate grid level between the transmission grid and the distribution grid.



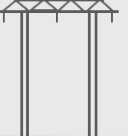
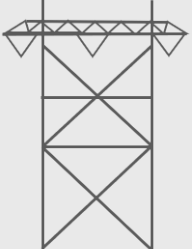
				
	Low-voltage distribution	High-voltage distribution	Regional	Transmission
Grid level	4	3	2	1
Voltage level	230, 400 V	1 - 22 kV	33 - 132 kV	(132), 300, 420 kV
Transformer	● Metering Point	■ Substation	▲ Transformer station	◆

Figure 1.2.: Schematic representation of the grid levels in Norway.

High-voltage distribution grid: Grid level 3 operates at a voltage of 1 to 22 kV. Some industrial customers and small producers are connected to the high-voltage distribution grid.

Low-voltage distribution grid: The low-voltage distribution grid, grid level 4, supplies final consumers at a voltage of 230 V or 400 V.

Note that each transformer between two voltage levels is associated to the grid level with the higher operating voltage. As an example, a transformer between the regional grid and the high-voltage distribution grid will be classified as grid level 2, the same level as all other assets in the regional grid. Metering points, which represent final consumers, operate in grid level 4.

1.3.1. Terminology

For clarification we define common terms and expressions that will be used throughout this report to refer to grid assets and involved stakeholders. We will refer to grid levels according to the definitions of NVE, listed above and illustrated in Figure 1.2.

When speaking of the distribution grid, we refer to assets in both the high-voltage and low-voltage distribution grid, spanning voltage levels from 230 V to 22 kV. In this context, we also want to introduce the abbreviations HVD and LVD for the grid levels in the distribution grid.

In practice all connection points between different voltage levels are transformers. To differentiate between grid levels we will use the different terms for assets in level 2 and those in grid level 3. Transformers between the high-voltage distribution grid and the low-voltage distribution grid will be referred to as *substation*, from the Norwegian term *nettstasjon*. For transformers between the regional grid and the high-voltage distribution grid the term *transformer station* will be used. The term metering point refer to the point where final consumers are connected to the low-voltage distribution grid on grid level 4.

The term *point* is used to describe all line end-point objects. I.e. commonly a metering point, a substation or a transformer station in the LVD or HVD grid. The clustering algorithms presented in Chapter 2 can be used to label any type of points. However, for the results presented in Chapter 4

we will only apply the clustering algorithms to the LVD grid for clustering of metering points, and we will thus use the terms *metering points* and *points* interchangeably. The term *node* is reserved for connection points in the LVD grid, i.e. substations. A virtual node, as introduced in Section 1.2, is the single alternative connection point for all metering points belonging to a given cluster of metering points in the LVD grid.

The Norwegian term *nettselskap* will be translated as *grid company* or Distribution System Operator (DSO) in this report. As the name implies, a Distribution System Operator (DSO) operates the distribution grid which includes the low- and high-voltage distribution grids, or grid levels 3 and 4.

The area in which one grid company operates the grid is called *konsesjonsområde* or *nettområde* in Norwegian and will be referred to as *grid area* in this report.

2. Clustering algorithms

This chapter is devoted to describing and discussing a suite of clustering algorithms. We will start by introducing the naming conventions used and some desired properties we want from our clustering methods, before getting into the description of the various algorithms explored in this study.

2.1. Preliminaries

Clustering is the task of labeling objects in such a way that objects that have the same label are considered to be close to each other, in some sense. The number of distinct labels should then be lower than the number of objects under considerations, and we say that the objects with the same label belong to the same *cluster*.

Though theoretically the word *close* can be interpreted in many ways, we will in the application of clustering metering points together interpret the word as being close in geographical distance.

However, we will here point out the potential utility of clustering algorithms on other grid levels than the LVD grid, and will therefore refer to the objects that we want to label as *points*.

The clustering of a set of points is not in and of itself enough to aid in the power distance calculation. As outlined in Section 1.2, the points belonging to the same cluster should be represented by a single virtual node. We call it virtual in this context since it is created artificially and does not represent a physical location. The power distance calculation on these virtual nodes will then be computationally a lot less expensive than a power distance calculation on the full set of points. In the following, we will call these virtual nodes *cluster nodes* to emphasize that each one represents a cluster. Moreover, to ascribe to each cluster node a geographical position we have used a cluster's *centroid*, which is the

the average geographical position of all points in the cluster.

2.2. Desired features and selection criteria

As already stated, all clustering algorithms will cluster together points that are physically close to each other when using physical distance as a way of determining how close two points are to each other. This is the primary criterion that should be upheld by any algorithm one chooses to use. However, there are usually other secondary properties that can also be achieved without noticeably affecting the *closeness*-criteria. These secondary criteria are what distinguishes the various algorithms from each other, as no single algorithm can achieve everything. The subsections below will discuss all secondary criteria considered. Later, when discussing the specific algorithms, the support for the various secondary properties will be addressed.

2.2.1. Anchored clusters

As mentioned in Section 2.1, we need to be able to form a cluster node from a cluster. That is, construct a point that is to represent the whole cluster, which will be used in the ensuing power distance calculations.

However, we can consider the opposite perspective, and say that from a set of nodes we want to cluster a set of points. In the context of metering points and substations this is the problem of assigning each metering point to one substation. Similar to the usual clustering problem, it is reasonable that the points that are assigned to the same cluster node should in some sense be close

to each other, but also close to the provided cluster node. For instance, when calculating the power distance, this approach is suitable when trying to consider how efficiently DSOs have connected their metering points to substations.

Achieving this task requires us to move outside the canonical scope of clustering algorithms that are typically presented in the literature, as this approach of starting with a set of cluster nodes and assigning points to each one is usually not addressed.

As discussed in Chapter 3, we study two types of data sets; sets that contain information about the real grid topology, i.e. metering point – substation connections and substation – transformer station connections; and sets that do not contain such information. For the LVD grid, which is mostly a radial grid, each metering point has an associated substation. We will refer to this substation as the *root node* for all metering points connected to it.

Following the idea that all clusters have a root node that they connect to, it stands to reason that each cluster should have one and only one root node. If one tries to naively run a classical clustering algorithm without addressing the allocation problem, many clusters will either end up with multiple or no root. In the context of clustering metering points, this would mean that a straightforward application of some clustering algorithm would almost surely lead to clusters where points included in the same cluster are connected to different substations.

The reason that classical clustering algorithms fail to assign points to a given set of cluster nodes correctly is that they are designed to work with a minimum number of assumptions. Adding the restriction that points in a cluster should not only be close to each other, but also close to some given point is simply outside the scope of most clustering algorithms. That does not mean that there are no such algorithms, just that not all of them do so canonically. With some algorithm adjustments, many algorithms can be made to support cluster node assignment. These will be discussed below.

For simplicity of notation we refer to the idea of

allocating each point to exactly one cluster node as *cluster anchoring* or just *anchoring*.

From a practical point of view, the need for anchoring only arises when the cluster node is taken from a given data set. If a cluster node is not provided, but instead created virtually, anchoring is not necessary. Assignment can simply be done after the clustering is completed by creating virtual nodes, as described in Section 2.1, one for each cluster. Note though, that creating a new virtual cluster node should be done with care, as naive approaches like using the cluster center as substation location can end up in impossible locations like in the middle of water. This observation is discussed in more detail in Section 5.1.

2.2.2. Algorithm determined cluster count

When substations and their locations are not available from data, the idea of adding anchoring points becomes moot, as assignment of cluster nodes can be done after allocation of the clusters. Furthermore, in such a context one might not even want to fix the number of clusters, as there likely is no single correct answer to this question. Instead, it might be better to have algorithms that themselves determine the optimal number of clusters according to some criteria.

2.2.3. Balancing the size of each cluster

By default, there are no limitations on how many metering points should be in each cluster. One rule that could be imposed is that each cluster should have some specific size. In the context where anchoring is in place, this size imposition means that data about the clusters, like cluster energy consumption or cluster energy capacity can be included as a factor to converge towards. In Chapter 4, the maximum daily consumption over a year in each cluster node is used for this purpose. Note that because this is a secondary requirement below the primary one of low distance between



samples inside a cluster, this rule can only be enforced where it is not in contradiction to the primary requirement.

It is also worth noting that assuming that anchoring is not in place, we can no longer impose restrictions on cluster location, individual cluster sizes, or total demand in each cluster. The only size restrictions that makes sense to impose is that the clusters are of equal size or, in the case where point weights are available, equal aggregated point weights for each cluster. This is because clustering is an exploratory tool and the label of a cluster is not determined before the clustering is performed. There are also few tools available for doing this in practice, so only an anchored variant is considered here.

2.2.4. Node stability

One important question to ask when looking at the output of a clustering algorithm is how stable the results are. If moving the location of a point a small amount ends up in a different cluster association the classification must be considered unstable. Having a qualitative measure for each node allows us to assess if this is the case without doing manual and costly exploration of the data. Some observations related to node stability for the test cases studied in this report are discussed in Section 5.2.

2.2.5. Model simplicity and computational cost

All the features described in the previous subsections will necessarily add both model complexity and, in many cases, computational cost. Arguments exist both for and against including each of these features. What to include is a question of how well they work, available data, and stakeholder preferences. To that end we have implemented a small suite of different algorithms with somewhat disjointed set of properties. These will be discussed below.

2.3. Overview of algorithms and recommendations

The properties described in the previous section are all the features that would be nice to incorporate into a working clustering algorithm. However, adding all properties at the same time is not possible. We therefore explore a few different algorithms that will incorporate different subset of the features. This section will discuss these algorithms. In addition in Table 2.1 all algorithms and their features are summarized.

2.3.1. K-means clustering

The simplest of the popular clustering algorithms around is *K-means*. It is built on two principles: Each cluster is uniquely defined by its *centroid*, and each point belong to the cluster whose centroid is the closest. The centroid is simply the average geographical position of all points in the cluster. This creates a natural iterative algorithm: Start by allocating K cluster centroids at random and apply the following two steps:

- Assign all points to cluster with the closest centroid.
- Update centroid based on the current points in cluster.

Iterating between these two steps, eventually the algorithm will converge to a stable setup where metering point allocation and centroid do not change between iterations. There are some added complexities of more carefully selected start locations and stopping criteria. But in essence the underlying algorithm remains the same. In the implemented application the initial start location is defined by using a method called *k-means++* [6], and early stopping is set to 300, which is the default value in the Scikit-Learn implementation.

This algorithm is so basic that it does not include any of the features described in Section 2.2. However it is possible to include the anchoring by not using the iterative part of the algorithm. To

Table 2.1.: Overview over the various clustering algorithms and their features.

	K-means	anchored K-means	GMM	custom GMM	DBSCAN
Anchoring	no	yes	no	yes	no
Fixed cluster count	yes	yes	yes	yes	no
Fixed cluster size	no	no	no	yes	no
Stability score	no	no	yes	yes	no
Model complexity	low	low	high	high	low
Computational cost	low	low	high	high	low
Custom point weights	no	no	no	yes	yes
Cluster size based on demand	no	no	no	yes	no

do so we initialize the clusters by the provided substations, and just assign nodes to the closest centroid (k-means with max iteration parameter set to zero), the substations are guaranteed to be approximately in the middle of each cluster. In other words, we simply just use the substation as is, and assign all metering points to the closest substation.

Similar to anchoring, it is possible to add point weights to the K-means model as follows: This can be achieved by replacing the calculation of centroid in the second step of the above iteration by weighted averages, where the weights are user defined. A context-relevant example would be to use the demand of each metering point as weights. This would result in the cluster nodes being drawn towards points with high demand, and would possibly lead to a lower power distance evaluation than that of plain K-means. However, scikit-learn's implementation of K-means does not allow for the specification of point weights, and it is therefore not considered further in this study.

2.3.2. Gaussian mixture model

The vast majority of probability distributions that exist are parameterized, meaning they represent a class of different distributions that can be actualized in the case where the distribution parameters are selected. As seen in Figure 2.1, it is possible to select these parameters in such a way that they

are more or less superimposed over some data of interest.

Maximum likelihood theory is a formal framework for setting probability distribution parameters to fit to data. In essence the theory just state that the parameters are to be selected in such a way that the joint probability density of the data is maximized. This way you get the *most feasible* distribution where data are produced in high probability areas to the largest extent possible.

Gaussian Mixture Model (GMM) is a specific probability distribution defined as a weighted sum:

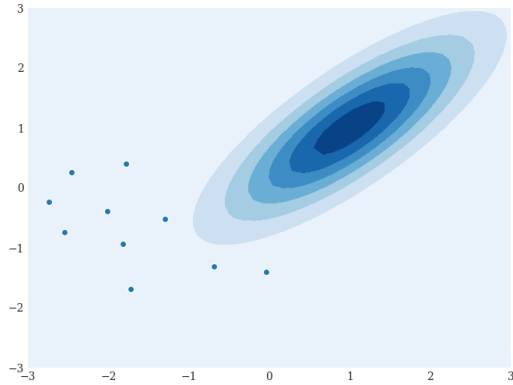
$$\begin{aligned}
 p(X \mid \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K) \\
 = \sum_{k=1}^K w_k p_k(X \mid \mu_k, \Sigma_k)
 \end{aligned} \quad (2.1)$$

where p_k are Gaussian probability density functions with mean μ_k and covariance Σ_k :

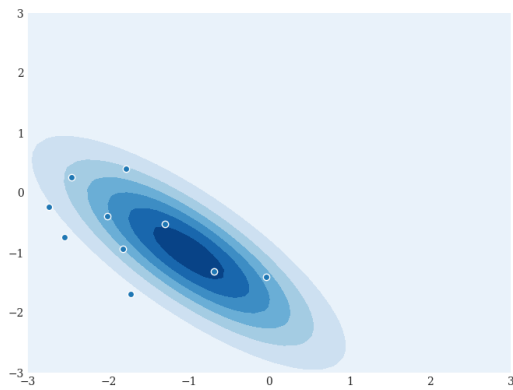
$$\begin{aligned}
 p_k(X \mid \mu_k, \Sigma_k) \\
 = \frac{1}{\sqrt{\pi^D |\Sigma_k|}} e^{-\frac{1}{2}(X - \mu_k)^T \Sigma_k^{-1} (X - \mu_k)},
 \end{aligned} \quad (2.2)$$

and w_k are weighting functions constrained to have $w_k > 0$ and $\sum w_k = 1$. If not otherwise stated, the weights are constant across clusters: $w_k = 1/K$.

The distribution shape of GMM allows for multiple modes and allows for spreading of the distribution across space. In particular the mean μ_k defines the center of the cluster, and the covariance



(a) Away from data.



(b) Superimposed on data.

Figure 2.1.: Probability distribution either superimposed over data or not. x and y axes represents an exemplary unitless Cartesian 2D grid.

Σ_k defines the *width* of the cluster. This can be observed in Figure 2.2.

Since the Gaussian mixture model consists of a probability density function, we can readily apply maximum likelihood to it and create a model where the data are covered by the density. To do this, we first make the assumption that our metering point data are drawn from the Gaussian mixture model, but also we assume that each sample is drawn independently. This results in the following

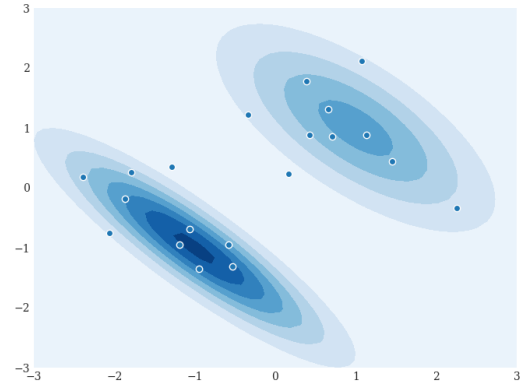


Figure 2.2.: An illustration of a simple Gaussian mixture model with two modes. x - and y -axes represents an exemplary unitless Cartesian 2D grid.

mathematical formula,

$$p(X_1, \dots, X_N \mid \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K) = \prod_{n=1}^N p(X_n \mid \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K), \quad (2.3)$$

where latter p is the probability defined in equation (2.2).

As a practical detail, finding the maximum likelihood is an iterative method, and requires a starting position. This can be done in multiple ways, but in practice initialization is done by first doing K-means as described in Section 2.3.1 and then applying the maximum likelihood optimization on the K-means results.

Gaussian mixture models have many different application, where clustering is only one. To put mixture models into the context of clustering, one needs to use the available probability density function to assign a metering point to each cluster. Formally speaking, we assign a metering point to a cluster if the associated Gaussian density (its likelihood) is the largest compared to the other clusters. As an example, consider Figure 2.2 which has two modes. In the context of clustering we then say that it has two clusters, one centered at the top right and one at the bottom left. Each

metering point is assigned to the density which has the highest probability. Note that this will result in a categorization similar to assigning by proximity when densities are far spread apart, but will likely yield different results when the densities are closer to each other.

As introduced in Section 2.2.4, it is useful to have a qualitative measure for each node to determine how stable an association between a point and a cluster really is. Since the Gaussian mixture model is a probability model, we can define such a measure in the context of probabilities. In particular, we are interested in the probability that a sample is generated from its assigned cluster l and not from any other. This probability can be calculated using the likelihood functions

$$P(X_n \text{ belongs to cluster } l) = \frac{w_l p_l(X_n | \mu_l, \Sigma_l, w_l)}{\sum_{k=1}^K w_k p_k(X_n | \mu_k, \Sigma_k, w_k)} \quad (2.4)$$

for each point X_l and cluster l .

We note that the distribution described in (2.1) could be replaced by a weighted sum of other distributions than the Gaussian normal distribution. An advantage in using Gaussian distributions is that the resulting optimization problem is relatively easy, compared to using other distributions. Moreover, the model (2.1) is highly expressive in what kinds of point distributions it is able to capture. For these reasons, the vast majority of the literature on the subject is focused on the use of Gaussian distribution in mixture models.

2.3.3. Customized Gaussian mixture model

Almost all clustering algorithms described in this chapter are standardized solutions gathered from the `scikit-learn` software library. This was chosen to avoid reinventing the wheel, and to ensure that the implementation follows best practice. However, to try to adopt the extra features described in Section 2.2, a custom application had

to be created. To this end, a Gaussian mixture model as defined in Subsection 2.3.2 was chosen. The alternative implementation will be discussed in this Subsection. For convenience, this new implementation will be referred to as *custom Gaussian mixture model*.

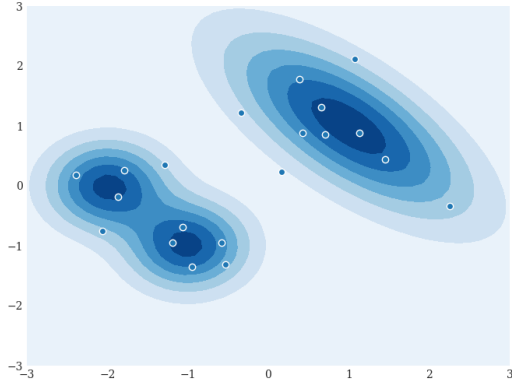
As introduced in Section 2.2.1, traditional clustering algorithms do not support anchoring. With a custom implementation available, we can extend the theory to introduce new features. With that in mind, we extend the Gaussian mixture model to include anchoring by using the following new likelihood function p' :

$$\begin{aligned} p'(X_1, \dots, X_N | \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K) \\ = \alpha p(X_1, \dots, X_N | \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K) \\ + (1 - \alpha) p(Y_1, \dots, Y_K | \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K) \end{aligned} \quad (2.5)$$

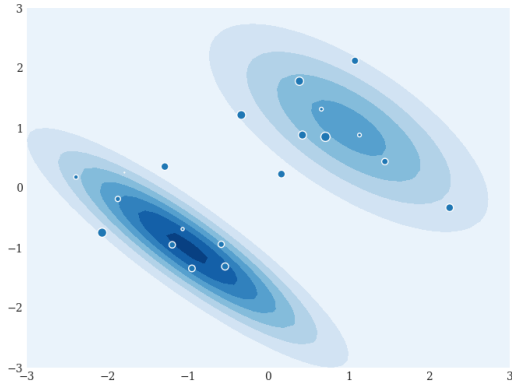
where Y_1, \dots, Y_K are the anchor points, and α is a configurable parameter on the unit interval, which we default to be 0.5. The value selected is somewhat arbitrary, but indicates that the importance of keeping the cluster close to a substation is equal to maximizing the likelihood over the samples.

This extended likelihood function can be looked upon as a weighted average between the likelihood of the metering point data, and the likelihood of the cluster nodes. The first part handles the clustering itself, the second part ensures that the likelihood stays around its anchor point. It does not formally guarantee one node per cluster, but in practice the solution will always include only a single node. A way to ensure cluster convergence to something sensible faster is to replace the K-means initialization described in Section 2.3.2, with just using the cluster nodes as means, combined with infeasible, large uncorrelated covariance matrices.

The size of a cluster in Gaussian mixture model is determined by the weight sizes w_k . So far these have all been chosen to be fixed and all equal, but there is no limitation requiring this. In figure 2.3, we illustrate how the distribution shape differs when the weights are not all equal. One mode is



(a) GMM with equal weighting



(b) GMM with uneven weighting

Figure 2.3.: GMM where cluster weights are either equal in size or not. x - and y -axes represents an exemplary unitless Cartesian 2D grid.

much larger than the other, indicating that it is likely that more samples will be a part of one cluster compared to the other.

As introduced in Section 2.2.3 cluster weights can be determined outside the scope of the optimization. The only requirement is that the values chosen are non-negative, and the values are normalized such that $\sum w_k = 1$. The latter is always done when applied.

Adding weighting to each cluster is possible within the scope of current literature. However, if one wants to weight each node in the same way,

the literature on such an approach is missing. To this end we have implemented an novel approach to include node weights. We start by reformulating the maximum likelihood formulation in (2.3) as a maximum log-likelihood:

$$\begin{aligned} & \operatorname{argmax}_{\mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K} \\ & \quad \log(P(X_1, \dots, X_N \mid \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K)) \\ = & \operatorname{argmax}_{\mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K} \\ & \quad \log(p(X_1 \mid \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K)) \\ & \quad + \dots \\ & \quad + \log(p(X_N \mid \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K)). \end{aligned} \quad (2.6)$$

This new log-likelihood will have the same optimal parameters because of the fact that the log-operator is injective, meaning that the largest point in a function will remain the largest point after a log-transformation. Doing this transformation is quite common in probability theory, as doing maximization of a log-likelihood is often both more numerically stable and more mathematically tractable. For our purpose however, we only observe that our optimization problem now is formulated as a sum where each term is dependent on a single node. Our novel approach is then to just add weights v_1, \dots, v_N to each term:

$$\begin{aligned} & v_1 \log(p(X_1 \mid \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K)) \\ & \quad + \dots \\ & \quad + v_N \log(p(X_N \mid \mu_1, \Sigma_1, w_1, \dots, \mu_K, \Sigma_K, w_K)). \end{aligned} \quad (2.7)$$

Using a formulation on this form allows us to get the effect of increasing the weight of a node, increases how much it counts towards the total. It also has the intuitive property that setting all weights equal reverts to the default implementation.

2.3.4. Density-based spatial clustering of applications with noise

The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm contrasts the

K-means and GMM algorithms, described in Sections 2.3.1 and 2.3.2, in that it does not depend on the convergence of an iterative method to determine a clustering of the input nodes. Instead, each node is categorized according to its surroundings and some user defined parameters, which will be elaborated on in the following description. The computational cost of DBSCAN compared to these other methods is as a consequence low.

In general terms, each node x is categorized according to the following:

- x is a **core point** if there are *sufficiently many* points, counting itself, in its *neighborhood*.
- x is **reachable** from a core point y if there is a sequence of core points y_1, \dots, y_N , with $y_1 = y$, y_{n+1} in the neighborhood of y_n for $n = 1, \dots, N-1$ and x in the neighborhood of y_N .
- x is an **outlier point** if it is not reachable by any core point.

See figure 2.4 for an example of the classification of a simple set of points.

We can then partially cluster the points by saying that each core point must belong to a cluster and all points reachable from the same core point belong to the same cluster. Partially clustering here means that we are not guaranteed that all points belong to a cluster – i.e. the outlier points, which are not themselves core points, and not reachable from one either.

Some terms are left intentionally vague in the above description to indicate that the user has some control over how to define them, and as such control the resulting clustering.

In the definition of core points it is up to the user to describe how close a point has to be to another in order for it to be in the other's neighborhood. The natural way of determining this is to use the Euclidean distance together with a threshold value ϵ , and say that y is in the neighborhood of x if

$$\|x - y\|_2 < \epsilon,$$

where $\|\cdot\|_2$ denotes the standard the l^2 norm of the geographical position of a node. The easiest way

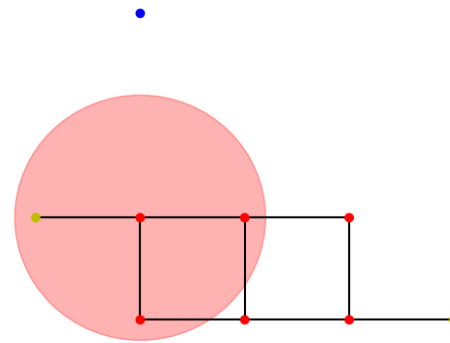


Figure 2.4.: DBSCAN's classification of a set of points. The core points are drawn in red, reachable points that are not themselves core points are yellow, while outlier points are drawn in blue. Edges between points are here meant to signify that they are in each other's neighborhood. The bigger red circle represents the neighborhood of the specific point in its center. Here, core points are those with at least 3 points in its neighborhood.

to control the influence the output of the DBSCAN algorithm is by the selecting an appropriate value for ϵ , where higher values will yield a higher number of core points and fewer clusters. However, there is nothing in the way of stopping the user from using a different metric than the euclidean. For instance, one can use a different l^p metric or incorporate topographical data when measuring the distance between points.

In a similar vein, the user must also provide how many points have to be in a point's neighborhood for it to be described as a core point. Requiring a core point to have many neighboring points will yield fewer core points, resulting in more clusters, but also more outliers. There is also some flexibility here in the way we choose to count points. In the current application, the points are metering points and core points are metering points that are geographically close to a high number of other metering points. However, we can give some metering points more weight than others based on their relative demand. In the following results chapter,

metering points with higher maximum daily demand across a year will count more towards core points designation than metering points with lower demand. This will result in metering points with historically high demand having a high probability of being included in a cluster. Outliers will then be metering points with historically low demand and thus have a relatively low impact on the power distance.

The fact remains, though, that with DBSCAN the user only has indirect control over some of the features described in Section 2.2. One can attempt to anchor clusters by prescribing a heavy weight to certain nodes, but it does not guarantee that any additional nodes will be included in the same cluster or that every node will be reachable by an anchor node.

The same goes for the balancing of cluster sizes. For instance, having one cluster much larger than the others is usually a result of non-uniformly distributed input data, something that is not easily remedied or controlled by the user specified parameters.

3. Data sources and data handling

In this chapter we discuss data sources and the necessary pre-processing steps.

3.1. Data needs for power distance computation

In the process of designing computational methods for the power distance, the importance of high-quality input data has been highlighted. Alongside efforts to refine the methods to compute new output parameters RME has also emphasized the need to improve the required input data and reduce data needs. In the three previous studies on power distance [2, 3, 4], three main data sets have been used:

- Metering data per metering point and/or substation with an hourly resolution.
- Geographical information on the location of metering points, substations and transformer stations.
- The physical topology of the distribution grid, i.e. the links between metering points, substations and transformer stations.

In [3], real grid data from selected test cases in the Norwegian distribution grid [3] was coupled with metering data aggregated to substation level. As part of the study, methods to simplify and error-check grid data were developed, however, the study led to the conclusion that the quality of available grid data was not sufficient to ensure a fair benchmarking process among all network companies. Instead, methods that create an idealized grid based on the location and demand per substation were further investigated in the HVD grid and alternative methods were defined to account for the LVD grid [7]. For methods that rely solely on metering

data and geographical coordinates, the quality of these data sets becomes increasingly important. In parallel, data processing routines to standardize data for meters and substations was developed in [5]. So far, all methods have been tested on the grid system of the DSOs KE Nett AS (Klepp), Mørenett AS (Mørenett), Jæren Everk AS (Jæren) and Glitre Energi Nett AS (Glitre). On the one hand, this allowed for direct communication in case of issues thus offering more direct insight to the data background. On the other hand the datasets lacked standardization and pre-processing was needed which increased the risk of erroneous data impacting the power distance calculations. By extending the scope of the output parameters to the LVD grid and including all Norwegian grid companies, the amount of metering data increases drastically to 130 000 substations and over three million metering points.

3.2. Data sources

This study uses data from Elhub and grid data provided by Klepp.

Developing methods based on Elhub data allows for more standardized data handling moving forward. In addition, the dataset from Elhub cover the entire Norwegian distribution grid. Consequently the analysis and use of metering data can be extended from the previously small scope of four grid companies. This will allow for valuable insights on differences and similarities between different grid companies.

Unfortunately the Elhub data do not include topological information about the grid. For the clustering algorithms presented in Chapter 2 employing anchoring, such as customized GMM and anchored K-means, information about the substa-



tion - metering point connections are needed to define root nodes for each cluster. Furthermore, for benchmarking we are interested in comparing the true grid to the artificial grid created by the clustering algorithms. To that end, grid data provided by Klepp must be used.

3.3. Pre-processing of data

The geographical location of each metering point is central to any of the above-mentioned clustering methods. The Elhub data in particular suffered from very low data quality with respect to geographical location. The problems encountered in this data set include:

Missing data : Only about 26 % of the data points had defined coordinates, and would have to be collected by other means.

Incorrect coordinate reporting : Where the Elhub data provided coordinates, several samples were placed somewhere in Nordland, even if the address data stated that the point should be in a different county. In the ensuing we have opted to trust the address data over the provided coordinates.

As a remedy to these problems, we used the API provided by Geonorge [8] to search for coordinates based on addresses. The API had an approximately 83 % success rate, resulting in a substantial increase in data quality, and yielded coordinates of the form longitude-latitude pairs.

As a general approach, we always want to convert available data to a format easily recognisable by the clustering methods under consideration. To this end we have had to perform a series of pre-processing steps before getting to the actual data analysis part.

Collect All data was stored as CSV files, and our scripts and programs used Pandas to read them into a programmatically workable format. This format is fine for smaller case studies, like this, but it does not scale well, and other formats should be considered when scaling to

larger computations. The main problem with reading the data from a CSV is the amount of data having to be stored in-memory when performing the clustering. A solution, as we have done in this study, is to filter the data set into smaller CSV files, for example on owning DSO for each metering point. The main drawback with this approach is the large number of files the user has to keep track of when scaling up the clustering to a production setting, and the maintenance cost this entails. A cleaner, more easily maintainable solution would be to store all metering points, and their relevant metadata, in a database to which the user can make appropriate queries when performing a clustering of the metering points.

Clean Some cleaning of the data was also needed. This included removal of data points still missing coordinates, and casting the data fields into an appropriate value type – e.g. metering point ID and municipality code to integers, or metering point consumption to floating point values.

Combine We also had to combine the provided consumption data with the metering points metadata. This was done by matching metering point IDs. In training our clustering algorithms, we used consumption data as an additional column in the metering point data set to represent sample weights (where applicable). We should also note here that what consumption, or what aggregate of consumption, to put in as sample weights can have an effect on output clusters.

Convert Since an integral part of the power distance calculation is the distance between metering points, the longitude-latitude pairs were transformed into UTM zone 33N (EPSG:25833) coordinates, which have unit meter. Thus making appropriately dimensioned distance calculations easier.

Supplement Many of the methods we are considering in this study require anchoring, which pre-

supposes that we have a set of cluster centers with well-defined locations. Both K-means and GMM can initialize these locations in more or less sensible way, but it is valuable to compare these clusterings with the corresponding clustering using real grid data. Therefore, on a subset of the metering points we have supplemented the data with what substation a metering point is connected to, and metadata related to some substations.

Concentrate Many large pre-processed data sets end up being "sparse" in the sense that there is information in the data set, but it is spread thin across too many columns. In this case, concentration and dimension reduction has been of little issue. To get dense datasets we have simply left out data fields not required in the model analysis, and removed samples with missing data.



4. Results

In this chapter we will consider some numerical experiments using the methods described in Chapter 2.

The first case study consists of metering points in Klepp municipality in Rogaland. Here we also have substation data available, which makes it possible to use methods that require anchoring, like customized GMM and anchored K-means. Having substation data, together with information about which metering points are connected to it, also provides us with a baseline to compare the power distance calculations with.

The second case study consists of metering points in the grid area of Mørenett, provided by Elhub. In this case we do not have available data on what substation each metering point is connected to, which makes only K-means++, vanilla GMM and DBSCAN relevant. There is also no baseline to make a meaningful comparison with. However, the geographical distribution of the nodes in this case exhibits some noteworthy features not present in the Klepp data.

In the methods where weighting of each metering point is applicable (customized GMM and DBSCAN), the weights are taken as the maximum daily demand for each metering point across the year 2020.

For the power distance calculations, we have used the Artificial grid method, as described in [3], on each cluster, together with a power distance calculation from a root of each cluster to their centroid. This last step is to make sure that all methods within the case study are comparable. Otherwise, a method that puts each metering point in its own separate cluster would yield the best power distance.

For methods with anchoring, the anchor nodes are used as cluster roots, while for methods

without anchoring we have used the centroid of each cluster. In the case of DBSCAN we have also treated the outlier points as their own cluster. This is not the only way to handle this case. For instance, one could consider adding each outlier point to its nearest cluster. However, in this study we have opted to treat them as a separate cluster to emphasize one of the main problems with DBSCAN.

Moreover, the user defined parameters in DBSCAN were chosen so as to strike a balance between having few outliers and many evenly sized clusters. In the following studies we used only the size of a metering point's neighborhood and how many points were reachable by a metering point to be considered a core point as parameters to tune the DBSCAN models. Also, the counting of each metering point in DBSCAN was weighted by their maximum demand across 2020. Thus, metering points with a historically high demand will have a higher probability of being assigned to cluster.

The above mentioned maximum daily demands on each node are also used in the ensuing power distance calculations. Lastly, throughout the examples we have used $\alpha = 0.4$ for the power distance scaling parameter.

4.1. Case 1: Klepp

Our first case is the data provided by KE Nett. For the different methods, these were the values for the user defined parameters that we chose:

- **GMM and K-means++:** For these methods we used 300 components, to approximately mimic the number of clusters in the baseline and the anchored methods.
- **custom GMM:** In addition to using the provided

substations as anchors, the anchor weighting was chosen to be 0.5. That means that the mixture model gives equal weight to maximizing the likelihood of the point data and keeping the clusters centered around the substations.

- **Anchored K-means:** This method has no other parameters to tune after setting the substations as cluster anchors.
- **DBSCAN:** For this method we used $\epsilon = 300\text{m}$ as the limit for when two points are considered close, and a point has to have at least 4 neighbors, including itself to be counted as core point. We reiterate that this count is scaled according to maximum daily demand across 2020, so that a metering point with an average demand still counts as one and, for example, a metering point with twice of the average demand would count as 2 neighbors. These values were chosen as they were found to have the fewest number of outlier points.

Figure 4.1 shows the clustering produced by the different methods, together with the clustering baseline from the actual data in Figure 4.1a. All methods, except for DBSCAN, qualitatively conforms with the baseline clustering, in that there are many smaller clusters. DBSCAN contrasts this as being the only method where number of clusters is not a direct input parameter, but rather decides the number of clusters based on the distribution of metering points. In particular, Figure 4.1f shows that regions with a high density of metering points are usually grouped into large clusters.

Table 4.1 shows the number of clusters, total line length, sum of power distance on each cluster, and total power distance for each considered method. For the K-means and GMM methods the number of components were chosen to be the same as the number of substations in the dataset. Note that scikit-learn's GMM implementation converged to a model using fewer clusters. This simply means that the algorithm converged to a model where some gaussian distributions did not have any metering points that most likely belonged to it. This is in and of itself not a problem, and simply

means that the algorithm found it more likely that the data was distributed across fewer clusters.

The power distance measures of all methods come close to the baseline, especially the methods using anchoring, corroborating the idea that higher data quality from a DSO yields better results. The methods without anchoring tend to be below the baseline value, again with the notable exception of DBSCAN.

The same story continues when considering total line length. The methods using anchoring comes closer to the baseline, while the other methods requires less grid, except for DBSCAN.

DBSCAN's discrepancy in total line length and power distance from the other methods and the baseline can at least in part be explained by our choice of treating outlier points as a separate cluster. Figure 4.2 shows the artificial grid constructed for the outlier nodes in this example. Here we see that the outliers cover a large expanse, making large contributions to both the total line length and power distance. This large contribution from the outlier points would certainly be remedied by, for example, adding each outlier point to its nearest cluster instead.

Tables 4.2 to 4.4 show summaries of the distribution of cluster sizes, geographical spread, and demand for the different clustering methods together with the baseline.

As discussed in [3], the computational time of the artificial grid algorithm increases super-linearly with number of metering points in the grid. To keep the computational time of the power distance calculation low, we therefore want the cluster sizes to be as evenly distributed as possible. From Table 4.2 we see that K-means++ performs the best; the baseline and the anchored methods are nearly indistinguishable, while GMM and DBSCAN are the worst.

A similar story is told in Table 4.4, where we see that none of the methods outperform the baseline in evenly distributing the demand, with GMM and DBSCAN as the worst.

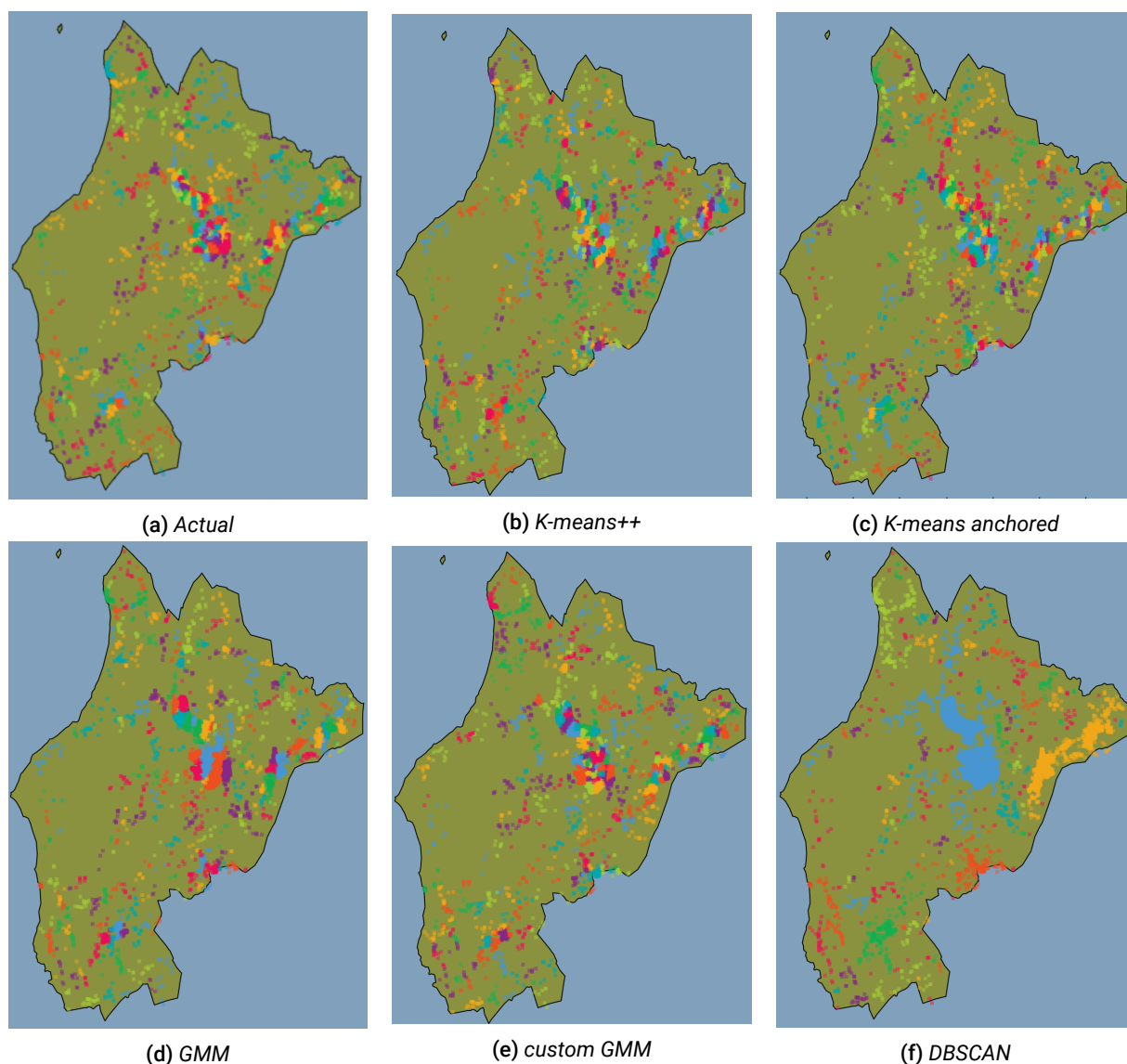


Figure 4.1.: Different clusterings of metering points in Klepp municipality.

Table 4.1.: Summary table for the various clusterings of the Klepp dataset. Here, "PD on clusters" is the sum of the power distances across all clusters, while "Total PD" has the added power distance from the centroid to all cluster roots.

	#clusters	Line length[km]	PD on clusters[(kW) ^α · m]	Total PD[(kW) ^α · m]
Actual	298	341	$4.23 \cdot 10^6$	$1.16 \cdot 10^7$
K-means++	300	299	$3.35 \cdot 10^6$	$1.09 \cdot 10^7$
K-means anchored	298	341	$4.01 \cdot 10^6$	$1.15 \cdot 10^7$
GMM	242	303	$3.69 \cdot 10^6$	$1.08 \cdot 10^7$
custom GMM	298	330	$3.80 \cdot 10^6$	$1.12 \cdot 10^7$
DBSCAN	84	377	$6.77 \cdot 10^6$	$1.23 \cdot 10^7$

Table 4.2.: Statistics for the distribution of cluster sizes for each considered clustering method. The columns show respectively the mean, minimum, maximum and some percentiles.

	mean	min	max	P10	P25	P50	P75	P90
Actual	29.92	1	254	2	8.25	15	41	81.1
K-means++	29.72	1	146	6	9	14	50	70.1
K-means anchored	29.92	1	236	5	9	15	42.75	82
GMM	36.85	1	796	4	7	11	20.75	113
custom GMM	29.92	1	348	3	9	14	35.75	85.6
DBSCAN	106.15	1	4351	2.3	5	10	19	46.5

Table 4.3.: Statistics for the distribution of mean distance of metering points to cluster centroid for each considered clustering method. The columns show respectively the mean, minimum, maximum and some percentiles. All values are displayed in meters.

	mean	min	max	P10	P25	P50	P75	P90
Actual	135.45	0	615.45	24.58	77.95	113.89	195.21	262.07
K-means++	133.23	0	348.60	69.39	84.07	125.51	171.87	212.54
K-means anchored	146.00	0	512.73	68.27	88.28	120.78	191.93	256.32
GMM	152.72	0	369.32	70.88	113.26	149.62	187.27	242.35
custom GMM	136.63	0	375.65	51.63	79.98	118.78	190.37	239.89
DBSCAN	254.43	0	4422.17	30.18	99.33	159.40	256.40	379.27

Table 4.4.: Statistics for the distribution of total demand for each considered clustering method. The columns show respectively the mean, minimum, maximum and some percentiles. All values are displayed in kilo Watts.

	mean	min	max	P10	P25	P50	P75	P90
Actual	$5.75 \cdot 10^3$	16.57	$1.21 \cdot 10^5$	$1.11 \cdot 10^3$	$1.98 \cdot 10^3$	$3.60 \cdot 10^3$	$7.06 \cdot 10^3$	$1.03 \cdot 10^4$
K-means++	$5.59 \cdot 10^3$	50.52	$1.33 \cdot 10^5$	$7.66 \cdot 10^2$	$1.30 \cdot 10^3$	$2.88 \cdot 10^3$	$5.89 \cdot 10^3$	$9.13 \cdot 10^3$
K-means anchored	$5.78 \cdot 10^3$	183.95	$1.31 \cdot 10^5$	$9.86 \cdot 10^2$	$1.59 \cdot 10^3$	$3.46 \cdot 10^3$	$6.50 \cdot 10^3$	$1.05 \cdot 10^4$
GMM	$6.93 \cdot 10^3$	68.36	$1.4 \cdot 10^5$	$5.97 \cdot 10^2$	$9.94 \cdot 10^2$	$1.85 \cdot 10^3$	$4.82 \cdot 10^3$	$1.85 \cdot 10^4$
custom GMM	$5.63 \cdot 10^3$	65.99	$1.20 \cdot 10^5$	$8.62 \cdot 10^2$	$1.41 \cdot 10^3$	$2.78 \cdot 10^3$	$6.69 \cdot 10^3$	$1.09 \cdot 10^4$
DBSCAN	$2.00 \cdot 10^4$	759.17	$5.05 \cdot 10^5$	$8.67 \cdot 10^2$	$1.24 \cdot 10^4$	$2.59 \cdot 10^3$	$4.43 \cdot 10^3$	$2.10 \cdot 10^4$



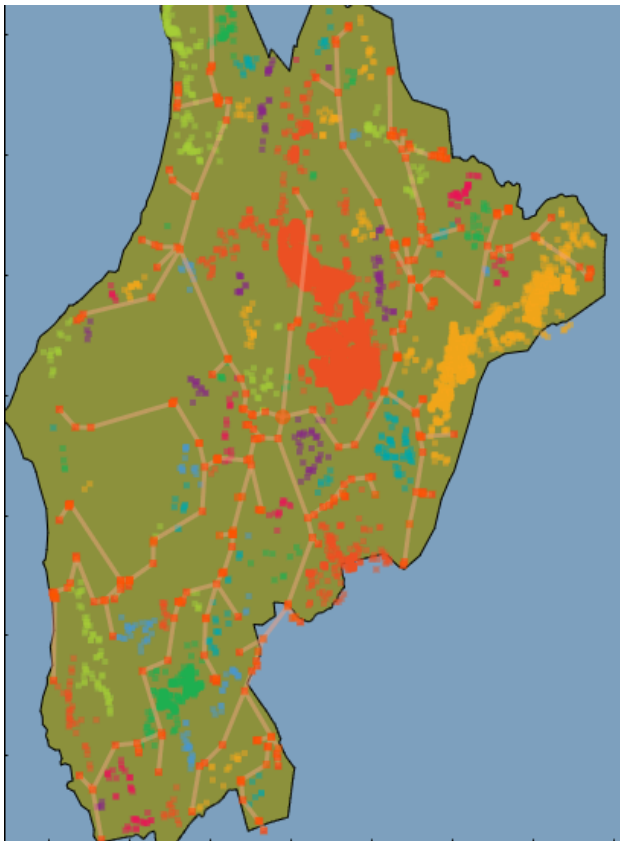


Figure 4.2.: Artificial grid for the outlier points from DBSCAN.

4.2. Case 2: Mørenett

As a second case study we consider the metering points belonging to Mørenett. As already mentioned, we do not have substation data here, which leaves us with no baseline, and methods requiring anchoring are no longer feasible. Still, the geographical area covered by the dataset provides a set of challenges that the considered methods should be able to handle, in particular the artificial nodes used in the power distance calculations should be located on land, and grid lines should not cross bodies of water. The values for the user defined parameters used to generate these results were as follows:

- **GMM and K-means++:** For these methods we used 700 clusters, as this closely resembled the number of points to number of clusters ratio in the previous case.
- **DBSCAN:** Here we used $\epsilon = 400\text{m}$ and a point had to have at least 10 neighboring points to be considered a core point.

In figure 4.3 we can see the clustering from K-means++, GMM and DBSCAN viewed on a section of Ålesund municipality. Qualitatively, we see the same effects as in the Klepp study. K-means++ and GMM favor smaller clusters, whereas DBSCAN can generate clusters with a high number of nodes in dense areas. Also worth noting are the oblong cluster shapes generated by the different methods, and most pronounced in the K-means++ case. This effect stems from a property of the K-means algorithm: A metering point is placed in a node's cluster if it is closer to that node than any of the others. For boundary nodes, i.e. nodes with no other nodes further out from it, all metering points further out will belong to that node. Thus, if there are several boundary nodes placed side by side, the resulting clusters will look long and thin.

The oblong cluster shapes resulting from the GMM method can be similarly explained by the cluster centers being initialized using K-means.

In figure 4.4 we see two examples of artificial grids generated for different GMM clusters. Figure

4.4a shows how the cluster centroid (here shown as a larger circular dot) tends to be placed on land. This should not be too surprising, as the majority of metering points in a cluster tend to be on the same area of a continuous section of land. This is, however, no guarantee for all nodes in a cluster not to be separated by water. We can see an example of this in figure 4.4b, where two grid lines are forced to cross water.

Finally, table 4.5 summarizes some attributes of the considered clustering methods on this dataset. It reiterates the points made in the Klepp study. Still DBSCAN overestimates the power distance compared to K-means++ and GMM. The same holds even more dramatically for line length. Again, this can mainly be attributed to the way we have chosen to handle outlier points as a separate cluster covering a wide expanse.

Note also that even if the Mørenett dataset is much larger than the Klepp dataset, roughly 25000 and 9000 metering points, respectively, the number of clusters output by DBSCAN is roughly the same. This showcases the difficulty in the parameter tuning of DBSCAN if you want to increase the number of clusters or balancing of cluster sizes.

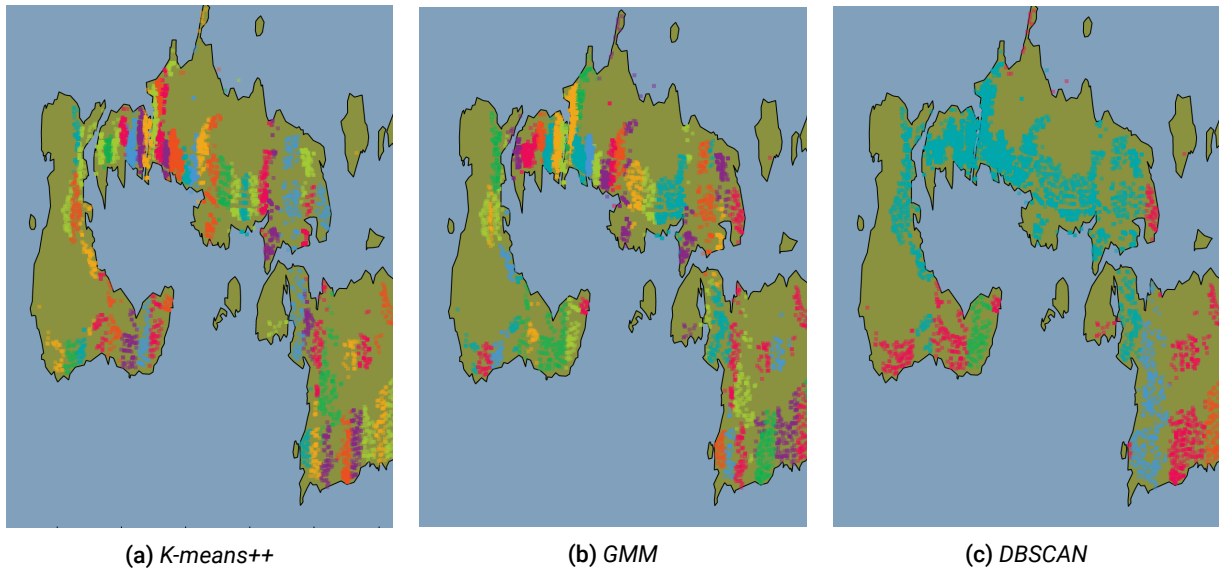


Figure 4.3.: Different clusterings of metering points in a section of Ålesund.

Table 4.5.: Summary table for the various clusterings of the Mørenett dataset. Here, "PD on clusters" is the sum of the power distances across all clusters, while "Total PD" has the added power distance from the centroid to all cluster roots.

	#clusters	Line length[km]	PD on clusters[(kW) ^α · m]	Total PD[(kW) ^α · m]
K-means++	700	1182	$1.43 \cdot 10^7$	$4.88 \cdot 10^9$
GMM	700	1156	$1.42 \cdot 10^7$	$4.88 \cdot 10^9$
DBSCAN	86	11024	$6.95 \cdot 10^8$	$5.46 \cdot 10^9$

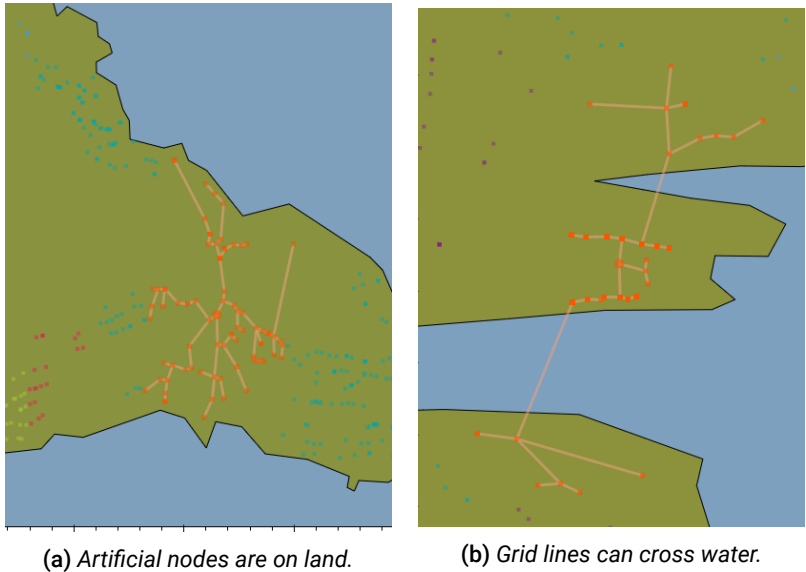


Figure 4.4.: Artificial grid for a GMM cluster where we see an edge cross water.

5. Discussions and recommendations

In this chapter we discuss the output results presented in Chapter 4, how the results should be interpreted and how clustering can be used in the calculation of power distance. We conclude by giving some final recommendations on applicability.

5.1. Interpretation of output results

When evaluating and comparing the clustering algorithms, it is important to differentiate between algorithm validation when the goal is algorithm improvements and tuning, and validation when the goal is testing the applicability of the algorithms. For the former, looking at the distribution of individual cluster size, distance between centroid and each node and geographical feasibility can provide insights into how the methods can be improved. For the latter, only the aggregated results are of interest. The cluster results are to be used as input for the calculation of a power distance metric for each grid area which in the next step is used to benchmark the grid companies. The implications from this observation are 1) that biases in the power distance calculation affecting all grid companies proportionally equal are irrelevant, implying that biases in the construction of clustering of nodes affect the calculation of power distance proportionally equal for all grid companies are irrelevant, and 2) variations in the construction of individual clusters that cancel out when viewing a grid area as a whole are irrelevant.

In Chapter 4, we presented the results of applying the clustering algorithms to test cases. It is easy to observe the algorithm outputs on a nodal-level. The results are important to illustrate how the algorithms work and for algorithm improvements.

But care must be taken when studying the results with the purpose of evaluating applicability.

In the analyzed examples, no points were located in water while some artificial grid lines crossed bodies of water. Other area types were not analysed. This behavior could also be observed for artificial grid methods without clustering [4] and for the case of Mørenett, the real grid also crosses fjords in multiple locations. The main difference being that the latter examples refer to the HVD grid, while the example in Figure 4.3b reflects the LVD grid level. In a real grid system, it is unlikely that low-voltage lines cross (large) bodies of water. As a result, the question arises to what extent a grouping of metering points that are separated by geographical obstacles is a good representation of the task.

As mentioned in Section 2.2.5, what features to include in a clustering algorithm is a question of how well they work, available data, and stakeholder preferences. To that end we have implemented as suite of different algorithms. The anchored methods (anchored K-means and custom GMM) resembles the actual, real grid more closely than the non-anchored methods and conform more to the decisions and choices made by the grid companies when building the grid. It is thus to be expected to find more examples of non-realistic cluster structures for the non-anchored methods. A comparison of the output between the methods on the aggregated level, which is the level that matters, however is more challenging.

5.2. Stability

The theory of clustering stability is a big topic, but is focused on the investigation of whether points remain in the same clusters across different clus-



terings — either with the same method, but different input parameter values, or across different clustering algorithms. It does so by varying the data set through perturbation, removal of points and adding noise to the data. For an overview of the topic see [9]. However, measuring how well a point stays the same irrespective of change in the data is not a good fit for analyzing artificial grids. The reason is that the quality of measure, how much a point remains in the same cluster, is a measure of the grids microscopic features. This to a large extent is an analysis of the data, and important if we are interested in knowing if the same clusters will always form. Our interest for this report on the other hand is related to the macroscopic features, with stability in power distance being the most important. Given this, we have considered doing classical stability analysis as outside the scope of this report.

Without a formal framework for doing cluster stability analysis, we instead take the simpler approach of looking at the simple sensitivity of the output score. The basic idea behind sensitivity analysis is to observe how small changes in input parameters will affect the output score. Here by parameter we mean variables that we can control from the outside. In our context we can define the choice of clustering algorithm as one parameter and the choice of area as another. As for output score, we will look exclusively at total power distance. In other words, we are interested in how much the power distance changes as a function of changing algorithm and test area.

Sensitivity analysis is, like cluster stability, a bigger topic with multiple approaches available, a derivative-based method being perhaps the most popular, where a high derivative of the power distance indicates low stability. However, since we do not have full control over the models (we can not get the derivative of a clustering method), we can not use this. Instead we will look at variance-based sensitivity analysis, and in particular first order total order sensitivity indices (also known as the main

and total Sobol indices):

$$S_i = \frac{\mathbb{V}(\mathbb{E}(Y | Z_i))}{\mathbb{V}(Y)} \quad S_{Ti} = \frac{\mathbb{E}(\mathbb{V}(Y | Z_{\setminus i}))}{\mathbb{V}(Y)}$$

where \mathbb{E} is the mean, \mathbb{V} is the variance, Y is power distance, Z_i are parameter with index i , and $Z_{\setminus i}$ are all parameter except the one indexed i . This method has the advantage of being easy to implement while also giving an intuitive score of how much a model changes. To estimate these values we use Saltelli's method with resampled values — a particular kind of Monte Carlo method.

Note though, that this analysis is based on the available data and analysis as described in the previous sections. In particular, as there are only two areas available, the results should first and foremost be interpreted as indicative, and will likely change as more areas are included.

The first thing to note is that the total variance of all experiments is 5.718×10^{18} , but the coefficient of variance is just 1.41. This means that though the variation is quite big, it is not very big relative to the scale of the power distance values. Breaking down this variance using variance based sensitivity analysis and we get

$$\begin{aligned} S_{algo} &= 0.002 & S_{area} &= 0.502 \\ S_{Talgo} &= 0.025 & S_{Tarea} &= 0.749 \end{aligned}$$

This main indices can be interpreted as the direct fractional contribution from a parameter to the total variance. So in particular $0.022 \times 5.718 \times 10^{18} = 0.174 \times 10^{18}$ for the choice of algorithm and $0.563 \times 5.718 \times 10^{18} = 3.124 \times 10^{18}$ for the choice of area. Another way to look at this is how it changes the coefficient of variance. The part of the coefficient that the choice of algorithm affects is 0.076. The same number for the selection of the area is 0.997. In other words, the selection of the area significantly contribute to the coefficient's size. The selection of algorithm does not. This means that the power distance is much more dependant on which grid area it is calculated for than what clustering algorithm is used.

5.3. Implications for the power distance variable

In Chapter 1 we stated the goal of developing methods to improve the computation of new output parameters without negatively affecting the fairness in the DEA model. However, the dual of this goal might also be desirable, namely to aid the development of a more fair DEA model without negatively affecting the computational burden.

The distribution power grid is inherently layered, as defined by the voltage level. Most fundamentally, the distinction between the LVD and HVD grid as defined in Section 1.3. The number of metering points and substations, and consequently also the amount of data, as well as different legislation may warrant the use of individual methods for the two grid levels. Overall, any method to compute new output parameters in the distribution grid should reflect the task of supplying power to all customers, covering both grid levels. As discussed in [4], this introduces questions as to how methods developed for each level should be combined and potentially weighted against each other. Furthermore, the separation point between LVD and HVD is, at least partly, endogenously chosen by the grid company. The choice between building 230 V vs 400 V LVD grids is an example of this, as the voltage level dictates the maximum line length (without experiencing disproportionate high power losses).

A cluster of metering points will implicitly define the separation between LVD and HVD, and by using the same user defined parameter sets (which varies from algorithm to algorithm) across all grid areas, the problem of variations in how LVD and HVD is separated will at least partly be reduced.

The constructed clusters of metering points can be used as input to the computation of power distance in two ways. The first option is to only use the topology defined by the cluster. The power distance output variables can be calculated individually for each cluster using any method suitable for the LVD grid and then combined with the power

distance for the grid defined by cluster centroids, cluster root node, or any other point associated with each cluster and the transformer stations. This method is similar to the alternatives discussed in [4] except that the LVD/HVD separation is algorithmically decided.

The second option is to also use cluster meta-data in addition to the topology defined by the clusters. The meta-data containing information about, among other things, number of metering points in the cluster, demand distribution and installed capacity. This information can be used to weight the nodes in the HVD grid, and thus avoid the need to compute power distance for the LVD grid.

The discussion on how to combine the LVD and HVD grid is however not dependent on the decision on whether to use clustering methods or not. For the case where nodes are anchored to substation locations this will give similar results as using the dataset from Multiconsult [5], where metering data was aggregated to the closest substation. Such an approach would not necessarily reflect the distribution of customers in the LVD grid. For methods where nodes are not anchored, the properties of the resulting clusters implicitly account for the distribution of metering points.

5.4. Other considerations

5.4.1. Incentive effects and data quality

As discussed above, using a clustering algorithm should yield a more exogenous power distance measure as the choice between LVD and HVD is determined through the algorithm. We will also argue that the grid companies will not have incentives or opportunities for delivering low quality data. With respect to metering data, these are data that are important for the imbalance settlement between generators, suppliers and large end-users, carried out by eSett on behalf of Statnett in the Norwegian electricity market. Hence, incorrect metering values will not be acceptable from the perspective



Table 5.1.: Evaluation criteria for the considered algorithms.

	Exogeneity	Computational complexity	Tuneability	Computational cost
Actual	low	n/a	n/a	low
K-means++	high	low	low	low
K-means anchored	medium	low	low	low
GMM	high	medium	low	high
custom GMM	medium to high	high	high	high
DBSCAN	high	medium	high	low

of these stakeholders. For the geographical coordinates, these can be corrected (to a large extent at least) using other data sources that cannot be controlled by the grid companies.

In any case we consider that it will be difficult for the grid companies to use strategic reporting of data in practice, as the calculations involved are complex and the optimal reporting strategy can depend on the behaviour of other grid companies.

5.4.2. Distributed generation and flexibility

All analyses performed in this study used the maximum net demand per metering point as input, in line with suggestions from [7]. Consequently, distributed generation is indirectly assumed to alleviate the task of supplying power to any customer/producer in a cluster, i.e. production occurring in the same hour as consumption reduces the required power flow to a metering point. Given that the main clustering criterion is minimal distance, a different way of accounting for distributed generation would not strongly impact the clustering algorithms, and rather affect the calculation of the power distance.

5.5. Recommendations

In this report we have presented a suite of algorithms that can be used to cluster metering points. The algorithms, as summarized in

Table 2.1, have different features, level of complexity and computational cost.

The first question we need to answer when forming a final recommendation is whether to use clustering or not. The alternative to clustering when computing the power distance is either to use the real grid as is, or simple hybrid approaches such as allocating each metering point to existing substations using simple closeness metrics, i.e. the method described in [5]. Considering exogeneity, the arguments in favor of using clustering are strong: The separation between LVD and HVD, cluster size and topology of the LVD as a whole are algorithmically decided. Complete exogeneity in the LVD grid is achieved using non-anchored methods, while the anchored methods can be viewed as an intermediate step between the non-anchored methods and relying on the real grid topology. The next argument would investigate what method best represents the *task of the grid companies to supply power*. As the true optimal grid is unknown, there is no natural way to test to what extent the various methods represents this goal. However, in this report we have demonstrated the flexibility of the proposed algorithms. By tuning the hyper parameters the output clusters can be made to resemble properties of the real grid, or any other grid deemed optimal. Thus to conclude the argument on whether to use clustering or not, designing a fair benchmarking that represent the true task of the grid companies is challenging, but using clustering algorithms will not set limitations in achieving that goal.

Given that using clustering algorithms is a good idea, the second question we need to answer is what clustering algorithm to favor. We identify four criteria: exogeneity, complexity, tuneability and computational cost. The second and third criteria represent a direct trade-off. Tuneability is needed to not restrict the possibility of designing a fair benchmark as described in the previous paragraph. On the other hand, excessive complexity is not wanted either in that unnecessarily complex methods are opaque, more difficult to understand and would be treated more as a black box by a DSO. A qualitative comparison of the considered algorithms are given in Table 5.1 . In this regard, we have no clear recommendation, but RME should consider the criteria and the properties of the different algorithms further before making a decision.

A. Acronyms

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DEA Data Envelopment Analysis

DSO Distribution System Operator

GMM Gaussian Mixture Model

HVD high-voltage distribution

LVD low-voltage distribution

NVE Norges Vassdrags- og Energidirektorat (Norwegian Water Resources and Energy Directorate)

RME Reguleringsmyndigheten for Energi (Regulatory authority for Energy)

B. References

- [1] NVE. *The Norwegian power system. Grid connection and licensing*. 2018. URL: https://publikasjoner.nve.no/faktaark/2018/faktaark2018_03.pdf.
- [2] THEMA Consulting Group. *Computing the power distance parameter*. 2018.
- [3] THEMA Consulting Group. *The power distance as an output parameter for grid companies*. 2019.
- [4] THEMA Consulting Group. *Methods for calculating power and energy distance*. 2021.
- [5] Multiconsult. *Developing Methods for Combining Data that Can Be Used for Calculating Power Distance*. 2020.
- [6] David Arthur and Sergei Vassilvitskii. *k-means++: The Advantages of Careful Seeding*. Technical Report 2006-13. Stanford InfoLab, June 2006.
- [7] THEMA Consulting Group. *Variables for capturing the task of reliability*. 2021.
- [8] Geonorge. 2021. URL: <https://ws.geonorge.no/adresser/v1/https://ws.geonorge.no/adresser/v1/>.
- [9] Ulrike von Luxburg. 'Clustering stability: an overview'. In: (2010). URL: <https://arxiv.org/pdf/1007.1075>.

About THEMA:

THEMA Consulting Group is a consulting firm focused on electricity and energy issues, and specializing in market analysis, market design and business strategy.

About Expert Analytics:

Expert Analytics is a science and technology consultancy company offering high-quality technology solutions. Primary fields of expertise are data science, mathematical modelling and advanced back-end software development.



THEMA Consulting Group

Øvre Vollgate 6
0158 Oslo, Norwegen

thema.no/
expertanalytics.no/

Expert Analytics

Møllergata 8,
0179 Oslo, Norway



NVE

Reguleringsmyndigheten
for energi – RME

Reguleringsmyndigheten for energi

.....

MIDDELTHUNS GATE 29
POSTBOKS 5091 MAJORSTUEN
0301 OSLO
TELEFON: (+47) 22 95 95 95

www.reguleringsmyndigheten.no