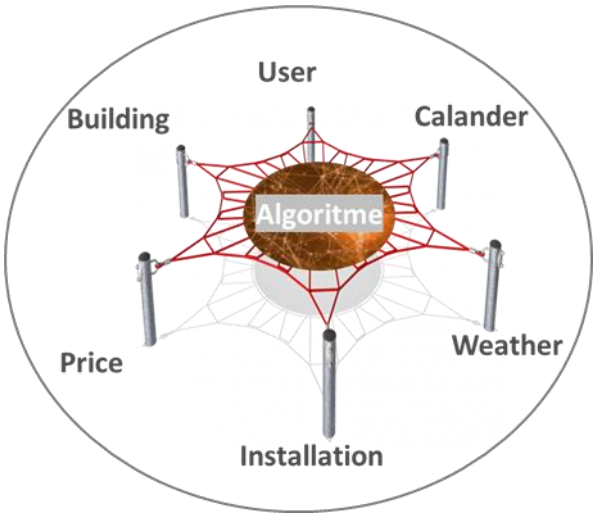
Brains for buildings

Rapportage

08-04-2025



Naam : Engineering Qien
Datum : 23-04-2025

Inhoud

1 Inleiding 6
1.1 Inleiding eindrapportage (gericht op Cloud Energy Optimizer) 6
2 Werkpakket 1 7
2.1 Inleiding op de bijdrage van Cloud Energy Optimizer aan Werkpakket 1 7
3 Verslagpunt 1: Integratie 4S3F-diagnoseframework op platform 8
3.1 Samenvatting en doel 8
3.2 Bevindingen uit de verkenning 8
3.3 Waarom het niet werkt in de praktijk (nu nog) 8
3.4 Praktijkvoorbeelden van barrières 9
3.5 Situaties waarin het wel (enigszins) werkt 9
3.6 Conclusie 9
4 Verslagpunt 2: Analyse eisen data-invoer & output 10
4.1 Samenvatting en doel 10
4.2 Verdieping van de eerdere bevindingen 10
4.3 Waarom deze eisen niet aansluiten bij de praktijk 11
4.4 Voorstel richting MVP (Minimum Viable Product) 12
4.5 Conclusie 12
5 Verslagpunt 3: Validatiegebruik gebruikersdata in diagnose 13
5.1 Samenvatting en doel 13
5.2 Uitgangspunt en context 13
5.3 Belangrijkste bevindingen uit de verkenning 13
5.4 Voorbeeld uit praktijk: CO_2-overschrijding bij bezetting 14
5.5 Richting oplossingsrichting 14
5.6 Conclusie 14
6 Verslagpunt 4: Testen foutdetectie-algoritmen op eigen use-case 15
6.1 Use Case 1 – Vrije Universiteit Brussel 16
6.1.1 1. Data collectie 16
6.1.2 2. Analyse 16
6.1.3 3. Bevindingen 17
6.2 Use Case 2 – BESI Duiven 18
6.2.1 1. Data collectie 18
6.2.2 2. Analyse 18
6.2.3 3. Bevindingen 18
6.2.4 Samenvattende conclusie 19
7 Verslagpunt 5: Input leveren voor visualisatie (interface WP3) 20

7.1 Samenvatting en doel 20
7.2 Gebruikerswensen: divers, persoonlijk en soms tegenstrijdig 20
7.2.1 1. Fysiologisch verschil: man vs. vrouw 20
7.2.2 2. Reismethode beïnvloedt temperatuurbeleving 20
7.2.3 3. Moment van de dag en fysieke gesteldheid 20
7.2.4 4. Kleuren en eenvoud in interface 21
7.2.5 5. Gedragsverandering bij monitoring 21
7.3 Conclusie 21
8 Verslagpunt 6: Automatisering koppeling gebouwdata 22
8.1 Inleiding: Architectuur & datastromen 22
8.1.1 Wat is JSON, en waarom gebruiken we het? 23
8.1.2 Opbouw van de datastructuren 24
8.1.3 Cloudplatform en verwerking 24
8.1.4 Schaalbaarheid en eenvoud 25
8.1.5 Conclusie 25
9 Verslagpunt 7: Opschaling en robuustheidstests 26
9.1 Inleiding: waarom schaalbaarheid essentieel is 26
9.1.1 Eenvoud als voorwaarde voor opschaling 26
9.1.2 Automatisering met wizard en koppelscripts 26
9.1.3 Van 240 uur naar 16 uur implementatietijd 27
9.1.4 Verwijzing naar de roadmap 27
9.1.5 Samenvattende conclusie 28
10 Verslagpunt 8: Bijdrage aan open bibliotheek met DBN-modellen 29
10.1 Wat zijn DBN-modellen? 29
10.1.1 Waarom DBN's voor gebouwbeheer? 29
10.1.2 Vijf voorbeeldmodellen binnen Cloud Energy Optimizer 30
10.2 1. Zone Comfort Model 30
10.3 2. Warmtepomp Efficiëntiemodel 30
10.4 3. Zoninstralingsinvloedmodel 31
10.5 4. Gebruikspatroonmodel 31
10.6 5. Boilerbelasting en verbruikspiekmodel 31
10.7 Conclusie 32
11 DBN-model 1 – Zone Comfort Model 33
11.1 Doel van het model 33
11.2 Stap 1: Definiëren van variabelen 33
11.3 Stap 2: Dynamiek en afhankelijkheden (structuur) 33

11.4 Stap 3: Rekenkundige relaties (voorbeeld) ... 34
11.4.1 Parameters (voorbeeldwaarden): ... 34
11.4.2 Voorbeeldberekening: ... 34
11.5 Stap 4: Outputnode ComfortAchieved(t+1) ... 35
11.5.1 Toepassing in praktijk ... 35
11.5.2 Adaptiviteit ... 35
12 DBN-model 2 – Warmtepomp Efficiëntiemodel ... 36
12.1 Doel van het model ... 36
12.1.1 Stap 1: Definiëren van variabelen ... 36
12.1.2 Stap 2: Dynamiek en afhankelijkheden ... 37
12.1.3 Stap 3: Rekenkundige relaties ... 37
12.1.4 Adaptiviteit en alarmgeneratie ... 39
12.1.5 Toepassing in praktijk ... 39
13 DBN-model 3 – Zoninstralingsinvloedmodel ... 40
13.1 Doel van het model ... 40
13.1.1 Stap 1: Variabelen ... 40
13.1.2 Stap 2: Structuur van het netwerk ... 40
13.1.3 Stap 3: Rekenkundige relaties ... 41
13.1.4 Schematische werking van het Zoninstralingsinvloedmodel ... 43
13.1.5 Stap 1: Bereken de reductiefactor φ ... 43
13.1.6 Stap 2: Bereken de verwachte temperatuurstijging (SolarGain) ... 43
13.1.7 Stap 3: Bepaal de comfortimpact (ΔT) ... 43
13.1.8 Stap 4: Interpreteer de ComfortEffect ... 44
13.1.9 Alternatief scenario: zonder zonwering én ramen open ... 44
13.1.10 Praktische toepassingen ... 44
13.1.11 Adaptieve werking ... 45
13.1.12 Conclusie. ... 45
14 DBN-model 4 – Gebruikspatroonmodel ... 46
14.1 Doel van het model ... 46
14.1.1 Stap 1: Variabelen ... 46
14.1.2 Stap 2: Netwerkstructuur ... 46
14.1.3 Stap 3: Rekenkundige relaties ... 47
14.1.4 Voorbeeld (empirisch bepaald): ... 47
14.1.5 Voorbeeldscenario ... 48
14.1.6 Adaptiviteit ... 48
14.1.7 Toepassing in de praktijk ... 48

14.1.8 Privacy en autonomie 49
14.2 Conclusie 49
15 DBN-model 5 – Boilerbelasting en Verbruikspiekmodel 50
15.1 Doel van het model 50
15.1.1 Stap 1: Variabelen 50
15.1.2 Stap 3: Rekenkundige relaties 51
15.1.3 Stap 4: Risico-inschatting piekbelasting 51
15.1.4 Voorbeeldberekening 51
15.1.5 Adaptieve werking 52
15.1.6 Praktische toepassingen 52
15.2 Conclusie 52
16 Eindconclusie: Van Technologie naar Samenwerking – De Weg naar Slimme Gebouwen 53
16.1 1. Inleiding 53
16.2 2. Samenvatting van de DBN-modellen 53
16.3 3. Belang van Samenwerking 54
16.4 4. Vereenvoudiging en Adaptiviteit 54
16.5 5. Conclusie 54
17 Slot conclusie 55
17.1 Succes zit in Samenwerking, Niet in Technologie Alleen 55
17.2 Techniek is er – maar de data ontbreekt vaak 55
17.3 Stapsgewijze benadering als succesformule 55
17.4 Voorbeeld van een succesvol project: 55
17.5 Samenwerking maakt het verschil 56

1 Inleiding

1.1 Inleiding eindrapportage (gericht op Cloud Energy Optimizer)

Het project *Brains for Building's Energy Systems (B4B)* heeft als doel om utiliteitsgebouwen te voorzien van operationele intelligentie: zelflerende softwarecomponenten die energie-efficiëntie, flexibiliteit en gebruikerscomfort combineren. In nauwe samenwerking met diverse markt- en kennispartijen is een reeks innovatieve prototypes ontwikkeld, getest en gevalideerd in realistische omgevingen.

Voor Cloud Energy Optimizer ligt de meerwaarde van dit project in de integratie van slimme diagnosesystemen (WP1), energievoorspellende modellen en sturingsalgoritmes (WP2), en gebruikersgerichte interfaces (WP3) in een open ecosysteem van datakoppelingen (WP4). De inzet op open innovatie, gecombineerd met praktische validatie via living labs en use-cases, maakt opschaling richting markt en toepassing in bestaande diensten goed mogelijk.

De uitkomsten bieden directe aanknopingspunten voor doorontwikkeling van CEOptimizer's platform: denk aan plug-ins voor foutdiagnose, voorspellend onderhoud en energieflexibiliteit die zijn getest in gebouwen met uiteenlopende installaties. Door standaardisatie en open APIs is de aansluiting met bestaande infrastructuur haalbaar en schaalbaar.

Dit rapport bundelt de belangrijkste resultaten per werkpakket, met een specifieke focus op de elementen die van strategische waarde zijn voor Cloud Energy Optimizer.

2 Werkpakket 1

2.1 Inleiding op de bijdrage van Cloud Energy Optimizer aan Werkpakket 1

Binnen Werkpakket 1 van het B4B-project staat de ontwikkeling van **zelfdiagnostische installaties** centraal, met als doel om energieverliezen en storingen in gebouwinstallaties **automatisch te signaleren** en zo **prestatiegericht onderhoud** mogelijk te maken. Cloud Energy Optimizer (CEO) levert hierbij een belangrijke bijdrage als platform dat al inzet op datagedreven energiemanagement.

De rol van CEO binnen dit werkpakket is gericht op het **integreren, testen en valideren** van slimme diagnose-algoritmen binnen de eigen softwareomgeving, en het leveren van waardevolle feedback vanuit praktijktoepassingen. CEO vormt daarbij de brug tussen technische innovaties uit de onderzoekslijn en de **markttoepassing in real-life gebouwen**.

De 8 punten hieronder zijn concrete acties die CEO in het kader van Werkpakket 1 dient uit te voeren. Elk punt is gekoppeld aan een specifieke fase in het project en gericht op het realiseren van duurzame, schaalbare en reproduceerbare oplossingen voor slimme energiediagnostiek

3 Verslagpunt 1: Integratie 4S3F-diagnoseframework op platform

3.1 Samenvatting en doel

Het 4S3F-framework is ontwikkeld als een geavanceerde methode voor automatische fout- en energiediagnose in gebouwinstallaties. Het maakt gebruik van **continue datastromen**, **Bayesiaanse netwerken** en **voorspellende analyse** op basis van gebouwdata, gebruikersgedrag en regelschema's. Binnen het B4B-project is het doel om dit framework te integreren binnen bestaande platforms, waaronder dat van Cloud Energy Optimizer (CEO), en zo praktijktoepassingen te realiseren.

3.2 Bevindingen uit de verkenning

Tijdens de verkennende fase is gebleken dat er een fundamentele kloof bestaat tussen de **academische benadering van het 4S3F-framework** en de **praktische realiteit in de markt**. Hoewel de theoretische aanpak overtuigend is op het gebied van foutdetectie en energiebesparing, zijn er in de praktijk aanzienlijke barrières die een succesvolle en brede toepassing in de weg staan.

3.3 Waarom het niet werkt in de praktijk (nu nog)

De kern van het probleem ligt in het verschil tussen de **kwaliteit en beschikbaarheid van data** in de academische testomgevingen en die in operationele gebouwen. Het 4S3F-model veronderstelt:

- Een **volledig en foutloos werkend gebouwbeheersysteem (GBS)** met hoge datadichtheid;
- **Gestandaardiseerde en consistente regelschema's** voor installatiebesturing;
- Continu beschikbare data van alle relevante gebouwcomponenten (ventilatie, verwarming, koeling, bezetting, comfortmetingen);
- Een omgeving waarin data **onafgebroken wordt opgeslagen, geanalyseerd en beschikbaar is op API-niveau**.

In de praktijk blijkt dit zelden het geval.

3.4 Praktijkvoorbeelden van barrières

1. **Data incompleet of inconsistente sampling**
   In veel gebouwen ontbreken datareeksen van specifieke componenten (bijv. temperatuur per ruimte, of CO_2-concentratie). Sensoren worden niet altijd goed gekalibreerd of werken tijdelijk niet, wat het model direct onbruikbaar maakt.

2. **Geen toegang tot regelschema's of documentatie ontbreekt**
   Veel installaties zijn verouderd, slecht gedocumenteerd of niet meer in eigendom van de originele installateur. Hierdoor is het niet mogelijk om het gedrag van het systeem betrouwbaar te modelleren.

3. **Beperkte opslagcapaciteit of kort retentiebeleid**
   Gebouwbeheersystemen verwijderen regelmatig historische data om opslagkosten te minimaliseren. Hierdoor is er vaak geen betrouwbare dataset over langere periodes beschikbaar voor analyse en kalibratie.

4. **Variatie tussen gebouwen**
   Iedere installatie is maatwerk. Dit maakt het ontwikkelen van generieke Bayesiaanse diagnosemodellen moeilijk. De handmatige aanpassing per gebouw maakt het onpraktisch en kostbaar in uitrol.

5. **Gebrek aan standaardisatie en API-toegang**
   Veel GBS'en werken nog met gesloten protocollen. De koppeling van 4S3F aan platforms zoals CEO vereist open en gestructureerde data, terwijl dat in de praktijk vaak alleen via omwegen beschikbaar is.

3.5 Situaties waarin het wel (enigszins) werkt

Er zijn echter uitzonderingen. In gebouwen waar:

- Recent geïnvesteerd is in moderne sensornetwerken;
- Het GBS compatibel is met externe koppelingen (bijv. via BACnet of Modbus over IP);
- Er duidelijke documentatie is van installatieconfiguraties en regelschema's;
- En er ervaring is met datagedreven beheer;

…kan het 4S3F-framework wél waarde toevoegen, bijvoorbeeld in pilotvorm of voor demonstratie.

3.6 Conclusie

De initiële integratie van het 4S3F-framework op het CEO-platform is voorlopig **niet haalbaar op grote schaal**, vanwege een mismatch tussen de **complexiteit van het model** en de **praktische beperkingen van real-life gebouwdata**. CEO adviseert om binnen het project een **meer gefaseerde aanpak** te hanteren: starten met een vereenvoudigde versie van het diagnosemodel, toepasbaar op de meest data-rijke gebouwen, en van daaruit iteratief uitbreiden. Daarnaast is het aan te bevelen om de eisen voor datakwaliteit en systeemarchitectuur expliciet te definiëren als randvoorwaarde voor succesvolle toepassing.

4 Verslagpunt 2: Analyse eisen data-invoer & output

4.1 Samenvatting en doel

De succesvolle toepassing van geautomatiseerde diagnosesystemen zoals het 4S3F-framework staat of valt met de **kwaliteit, volledigheid en beschikbaarheid van data**. In dit onderdeel is onderzocht **welke eisen realistisch zijn voor data-invoer en output** binnen het Cloud Energy Optimizer platform, met als doel een brug te slaan tussen de complexe academische modellen en de operationele realiteit in gebouwen.

4.2 Verdieping van de eerdere bevindingen

Tijdens het onderzoek naar de integratie van het 4S3F-framework werd al duidelijk dat er een aanzienlijke kloof bestaat tussen theoretische modellen en de praktijk. Die kloof komt vooral voort uit de **aanname dat alle benodigde data zonder beperkingen beschikbaar is**, wat in de praktijk vaak niet het geval is. De analyse van de data-eisen bevestigt dat dit niet slechts een technisch knelpunt is, maar een structureel organisatorisch en infrastructureel probleem binnen de gebouwde omgeving.

Het is gebleken dat **data-invoervereisten uit het 4S3F-framework niet realistisch zijn zonder aanvullende technische voorzieningen** of grondige aanpassing van de bestaande systemen. Tegelijkertijd biedt deze analyse inzicht in welke vormen van data wél breed beschikbaar zijn, en hoe daarmee praktische toepassingen haalbaar worden.

Overzicht van data-eisen uit het 4S3F-framework

De oorspronkelijke eisen voor data-invoer omvatten o.a.:

1. **Realtime data van klimaatsystemen** (temperatuur, luchtvochtigheid, CO_2, luchtdebiet);
2. **Historische gegevens van min. 1 jaar met hoge resolutie** (≥15-min interval);
3. **Volledig inzicht in regelschema's en instellingen**;
4. **Bezettingsdata op ruimteniveau**;
5. **Data van decentrale gebruikersacties** (bv. aanpassing thermostaat, openen ramen);
6. **Metingen van energieverbruik per component** (submeters);
7. **Koppeling aan externe bronnen (weerdata, energietarieven)**.

4.3 Waarom deze eisen niet aansluiten bij de praktijk

1. Geen standaardisatie in meetdata:
Data uit gebouwbeheersystemen komt vaak in **niet-genormaliseerde vorm**, wat integratie en interpretatie sterk belemmert. Volgens [Turk, 2020] en [Pauwels et al., 2017] is gebrek aan semantische standaardisatie een belangrijke oorzaak van interoperabiliteitsproblemen.

2. Ontbreken van bezettingsdata:
Bezettingssensoren zijn zelden standaard aanwezig. Toch wordt in het model aangenomen dat er een directe relatie is tussen bezetting en energiegebruik. Dit maakt het model minder toepasbaar op oudere of minder geavanceerde gebouwen.

3. Beperkte retentie van historische data:
Veel GBS'en verwijderen data na enkele weken of maanden. Structurele opslag ontbreekt, tenzij expliciet ingericht met additionele software of cloudoplossingen. Dit wordt ook benadrukt in [Verbeke et al., 2020].

4. Geen of beperkte logging van gebruikersacties:
Veel systemen registeren geen interacties op ruimteniveau. Het gedrag van gebruikers (zoals handmatig openen van ramen of gebruik van zonwering) blijft daardoor buiten beeld. Dit zorgt voor afwijkingen in de diagnose die het model zelf niet kan verklaren.

5. Outputverwachtingen zijn te complex:
De oorspronkelijke modeloutput richt zich op het signaleren van systeemfouten én het geven van geautomatiseerd advies. Maar gebouwbeheerders willen vooral weten: *"Is er iets mis, en zo ja, waar en wat moet ik doen?"* — deze vereenvoudigde output moet leidend worden.

Synergie met punt 1

Waar punt 1 vooral aangaf dat **de implementatie van het 4S3F-framework beperkt wordt door de praktijk**, laat dit tweede onderdeel zien **waar die beperkingen exact vandaan komen op dataniveau**. Door de data-eisen systematisch in kaart te brengen, ontstaat er zicht op:

- Welke vereisten *verzacht* moeten worden (bijv. historische diepte verkorten);
- Welke onderdelen van het model *gefaseerd* kunnen worden geïmplementeerd (bijv. foutdetectie zonder gebruikersdata);
- Waar aanvullende sensoren of koppelingen nodig zijn om bepaalde functionaliteiten mogelijk te maken.

Er ontstaat zo een **synergie tussen praktijk en theorie**, waarbij de inzet is om een **"light-versie"** van het framework te ontwikkelen, gericht op de 20% van de gebouwen die 80% van de data-infrastructuur al op orde hebben.

4.4 Voorstel richting MVP (Minimum Viable Product)

Op basis van de analyse is voorgesteld om het model in eerste instantie toe te passen op gebouwen die minimaal voldoen aan:

- Continue meting van binnenklimaat (minimaal temperatuur en temperatuur verdeelstuk);
- Historische data van minimaal 24 uur;
- Toegang tot GBS-data via BACnet of Modbus en via live interface;
- Basisregelset beschikbaar of eenvoudig te reconstrueren;

Voor deze doelgroep kan een eerste prototype worden uitgewerkt dat zich richt op **herkenning van patronen en afwijkingen**, zonder meteen geavanceerde diagnose of onderhoudsplanning.

4.5 Conclusie

De analyse van data-invoer en output maakt duidelijk dat **een generieke toepassing van het 4S3F-framework voorlopig niet realistisch is**, maar dat er wél mogelijkheden zijn voor een **gefaseerde, praktijkgerichte aanpak**. Door de eisen aan te passen aan de meest voorkomende datastructuren in de markt en het outputniveau af te stemmen op de eindgebruiker, ontstaat ruimte voor een haalbare implementatie binnen het Cloud Energy Optimizer platform.

5 Verslagpunt 3: Validatiegebruik gebruikersdata in diagnose

5.1 Samenvatting en doel

Binnen het B4B-project wordt gebruikersgedrag gezien als een belangrijke factor in het optimaliseren van gebouwprestaties. Gebruikersdata—zoals aanwezigheid, comfortvoorkeuren, en handmatige aanpassingen—kunnen storingen verklaren of zelfs voorspellen. Doel van dit onderdeel is om te onderzoeken in hoeverre dergelijke data **gevalideerd en bruikbaar** zijn binnen het 4S3F-framework en hoe deze effectief geïntegreerd kunnen worden binnen het Cloud Energy Optimizer (CEO) platform.

5.2 Uitgangspunt en context

In eerdere punten is vastgesteld dat het 4S3F-framework vergaande aannames doet over beschikbare data (punt 1) en dat veel van die data in de praktijk ontbreekt of niet gestandaardiseerd is (punt 2). Gebruikersdata vormen hierop geen uitzondering: hoewel ze van grote waarde zijn, zijn ze vaak ongestructureerd, subjectief en lastig te interpreteren.

Tegelijkertijd is de potentie enorm: gebruikersgedrag verklaart vaak de helft van de fluctuaties in binnenklimaat en energieverbruik. Maar om dat potentieel te benutten, moet die data eerst gevalideerd en geïnterpreteerd kunnen worden op een schaalbare en betrouwbare manier.

5.3 Belangrijkste bevindingen uit de verkenning

1. Gebruikersdata is heterogeen en ongestructureerd

Data over gebruikersgedrag komt in vele vormen: sensorgegevens (aanwezigheidsdetectie), logbestanden (bedieningsacties), comfortvragenlijsten (subjectief), of zelfs feedback via mobiele apps. Dit maakt het lastig om tot één uniforme datastructuur te komen die bruikbaar is voor diagnoses. De validatie van zulke gegevens vergt slimme conversie en filtering.

2. Validatie vereist hoge rekenkracht

Het verifiëren van gebruikersdata tegen andere datastromen (bijvoorbeeld: binnenklimaatparameters, regelacties en historische afwijkingsdetectie) vereist **combinatie van datasets**, **tijdreeksanalyse** en **modelvergelijkingen** op detailniveau. Dit vraagt om krachtige omgevingen zoals **Azure Machine Learning**, **Databricks**, of vergelijkbare AI-gedreven infrastructuur. De kosten hiervan zijn hoog, en de benodigde technische expertise is niet standaard aanwezig bij veel gebouweigenaren of installatiebedrijven.

3. Gebrek aan kennis vormt extra drempel

Zelfs bij beschikbaarheid van technische middelen blijkt uit de praktijk dat er **gebrek is aan kennis** over hoe gebruikersdata geïnterpreteerd en geanalyseerd moeten worden. Dit geldt voor zowel eindgebruikers (die hun eigen gedrag niet herkennen in het systeem) als gebouwbeheerders (die de diagnose als een ‘black box’ ervaren). Zonder goede vertaalslag van ruwe data naar begrijpelijke inzichten verliest het systeem aan bruikbaarheid én acceptatie.

Verband met eerdere punten

Net als bij punt 1 (implementatie 4S3F) en punt 2 (data-eisen), is hier opnieuw sprake van een **spanningsveld tussen ambitie en realiteit**. De ambitie om gebruikersdata mee te nemen in foutdetectie is terecht en kansrijk, maar staat of valt met de beschikbaarheid van:

- **betrouwbare datastromen** (zie punt 2),
- **schaalbare AI-infrastructuur**, en
- **deskundigheid aan gebruikerszijde**.

De validatie is dus niet alleen een technische opgave, maar ook een **organisatievraagstuk**.

5.4 Voorbeeld uit praktijk: CO_2-overschrijding bij bezetting

In een van de pilots binnen CEO werd onderzocht hoe vaak CO_2-overschrijdingen (boven 1200 ppm) samenhangen met aanwezigheidspieken. De conclusie: er is inderdaad een verband, maar zonder betrouwbare aanwezigheidsdata (zoals van passieve IR-sensoren of WiFi-tracking) is de verklaring onvoldoende hard. Bovendien bleken handmatige interventies (zoals het openen van ramen) buiten het systeem te vallen, wat de diagnose verder compliceert. Dit illustreert de noodzaak van zowel goede sensoren als contextbewuste algoritmes.

5.5 Richting oplossingsrichting

Op basis van deze bevindingen stelt CEO voor om in eerste instantie te focussen op gebouwen waar:

- Aanwezigheidssensoren of badge-logging beschikbaar zijn;
- Ruimtelijke scheiding per zone of verdieping mogelijk is;
- Gebruik gemaakt wordt van systemen met log van handmatige ingrepen (bijv. smart thermostaten, slimme zonwering);
- Er bereidheid is om gebruikersfeedback te verzamelen via apps of dashboards (eventueel gekoppeld aan incentiveprogramma's).

Daarnaast moet de **validatie-infrastructuur** bij voorkeur plaatsvinden in cloudomgevingen die reeds ingericht zijn voor AI-toepassingen — zoals Azure, AWS of GCP. Hierbij is samenwerking met partijen die over deze infrastructuur én kennis beschikken van essentieel belang.

5.6 Conclusie

Het meenemen van gebruikersdata in het diagnoseproces binnen 4S3F is **theoretisch waardevol, maar praktisch uitdagend**. De validatie ervan vergt geavanceerde technische middelen, hoge rekenkracht en specifieke kennis die niet breed beschikbaar is in de markt. Toch is er synergie met de eerdere analysepunten: alleen door data- en diagnosevereisten te **herijken en stapsgewijs te implementeren** ontstaat een haalbaar en schaalbaar model. Cloud Energy Optimizer adviseert om gebruikersdata in een latere fase van het diagnoseproces te integreren, met nadruk op datarijke gebouwen als testcases.

6 Verslagpunt 4: Testen foutdetectie-algoritmen op eigen use-case

Binnen het B4B-project vormt foutdetectie een essentieel onderdeel van de ambitie om gebouwinstallaties slimmer, efficiënter en duurzamer aan te sturen. Voor Cloud Energy Optimizer (CEO) betekent dit: het signaleren van structurele afwijkingen in temperatuur, energieverbruik of comfort zonder afhankelijk te zijn van vaste foutcodes of handmatige meldingen.

Daarom heeft CEO ervoor gekozen om een **eigen foutdetectiemethode** toe te passen, gebaseerd op het principe van **voorspeld versus gerealiseerd gedrag**. Deze aanpak maakt het mogelijk om afwijkingen te signaleren in installaties die op het eerste gezicht 'normaal' functioneren, maar in werkelijkheid suboptimaal presteren of onnodig energie verbruiken.

De foutdetectie werkt als volgt:

- Het platform voorspelt het verwachte gedrag van een gebouw (bijvoorbeeld: aanvoertemperaturen, energieverbruik of comfortwaarden) op basis van historische gegevens, gebruikspatronen, weersomstandigheden en installatiestructuur.
- Dit voorspelde gedrag wordt vergeleken met het daadwerkelijke gedrag van het gebouw, zoals gemeten via het GBS of sensornetwerken.
- **Afwijkingen** tussen beide worden gekwantificeerd en geclassificeerd. Grote of herhaalde afwijkingen wijzen op foutieve of inefficiënte aansturing.

Binnen deze methode maakt CEO onderscheid tussen:

1. **Lineaire sturingen** – traditionele systemen die vaste relaties volgen, bijvoorbeeld op basis van buitentemperatuur of kloktijden.
2. **Verrijkte machine learning-modellen** – adaptieve modellen die geleerd hebben van eerdere situaties en rekening houden met dynamische invloeden zoals zoninstraling, bezetting of gebruikersgedrag.

Juist door deze vergelijking ontstaat een krachtig instrument voor foutdetectie: lineaire sturingen laten vaak structurele afwijkingen zien op momenten waarop de omstandigheden afwijken van het gemiddelde. De machine learning-modellen van CEO herkennen deze patronen en anticiperen proactief, waardoor het verschil tussen beide sturingswijzen direct zichtbaar wordt als indicator voor fouten of optimalisatiepotentieel.

In deze rapportage worden twee praktijkvoorbeelden gepresenteerd – bij de Vrije Universiteit Brussel en BESI Duiven – waarin deze foutdetectiemethode is toegepast. Beide use-cases laten zien hoe afwijkingsanalyse niet alleen technische storingen zichtbaar maakt, maar ook leidt tot concrete aanbevelingen voor energiebesparing en comfortverbetering.

6.1 Use Case 1 – Vrije Universiteit Brussel

6.1.1 1. Data collectie

Voor deze use case is er tussen **november 2020 en juli 2021** dataverzameling gedaan op het bestaande gebouwbeheersysteem (GBS) van de Vrije Universiteit Brussel. De verzameling werd uitgevoerd na de opstartfase van het project (initieel gepland voor 2019, feitelijk begonnen eind 2020 na uitstel). De focus lag op temperatuurdata op verdeelstukken, zone-temperaturen, de werking van de warmtepomp, het schakelen tussen energiebronnen en comfortgrafieken.

- Er zijn **uur metingen** verzameld, gefilterd op **kantooruren**, met uitzondering van vakanties en feestdagen.
- Datapunten:
  - Geregistreerde aanvoertemperaturen t.o.v. de theoretisch optimale waarden volgens het CEO-algoritme.
  - Comfortmetingen per zone (binnen versus buitentemperatuur, afgezet tegen ISO 7730 adaptieve comfortgrenzen).
  - Retourtemperaturen van het verwarmingssysteem.
  - Foutdetectie in maandelijkse duurkrommen van afwijkingen.
  - Beïnvloeding van luchtbehandelingskasten door windrichting en -snelheid.

6.1.2 2. Analyse

De foutdetectie is gebaseerd op het vergelijken van het actuele gedrag van de installatie met het gedrag dat de algoritmes van Cloud Energy Optimizer (CEO) voorspellen op basis van eerdere, vergelijkbare omstandigheden (weer, gebruik, comfortzones).

6.1.2.1 Methode:

- Per maand zijn histogrammen opgesteld van temperatuurafwijkingen in stappen van 5°C.
- Daarnaast zijn comfortgrafieken opgesteld waarin de binnentemperatuur per zone is afgezet tegen de buitentemperatuur, inclusief ISO-comfortgrenzen.
- In specifieke zones zijn retourtemperaturen van het verwarmingssysteem geanalyseerd in combinatie met de inzet van de warmtepomp.

Berekening voorbeeld afwijkingsindex:

$$\Delta T = T_{\text{real}} - T_{\text{Qien}}, \quad \text{gem. afwijking} = \frac{\sum_{i=1}^{n} |\Delta T_i|}{n}$$

Gemiddeld lag de afwijking in koude maanden tussen de 3,2 en 5,1°C.

6.1.3 3. Bevindingen

De foutdetectie bracht verschillende structurele problemen aan het licht, waarvan de **niet-functionerende warmtepomp** het meest impactvol is:

- De retourtemperatuur bleek **structureel te hoog**. Een warmtepomp werkt efficiënt bij een laag temperatuurverschil tussen aanvoer en retour (typisch een delta T van ca. 5-7°C). In de VUB-case liep de retourtemperatuur echter op tot boven de aanvoertemperatuur van de warmtepomp. Hierdoor **valt de warmtepomp uit als effectieve warmtebron**.
- **Waarom heeft inzet van de warmtepomp dan geen zin?** Een warmtepomp is ontworpen om warmte te onttrekken aan een relatief koel retourcircuit. Als deze retour te warm is, kan het koelmiddel in de verdamper geen warmte meer opnemen, waardoor het compressieproces faalt en de pomp zichzelf uitschakelt of ineffectief draait. De SCOP (Seasonal Coefficient of Performance) daalt dan onder 1, wat betekent dat het systeem meer energie kost dan het oplevert.
- **Mogelijke oorzaken van de hoge retourtemperatuur:**
    - Een defecte of slecht afgestelde **driewegklep** die het water verkeerd mengt;
    - Een fout in de **bypass-regeling** die ervoor zorgt dat retourwater direct wordt teruggestuurd zonder afkoeling;
    - **Warmtelevering vanuit het stadswarmtenet** die het systeem onnodig opwarmt op momenten dat dit niet nodig is (bijvoorbeeld bij overschakeling).

Daarnaast is zichtbaar dat **luchtbehandelingskasten (LBK's)** rendement verliezen bij bepaalde weersituaties:

- **Analyse windrichting en snelheid** toont aan dat met name wind uit **noordoostelijke richting** bij hogere snelheden (>5 Bft) tot drukverschillen leidt in het gebouw, waardoor retourlucht anders circuleert of bypasskleppen vaker activeren. Dit leidt tot extra energieverliezen.
- Dit fenomeen manifesteerde zich vooral in de maanden maart t/m mei 2021, met verhoogde energiepieken en afwijkende comfortgrafieken in zones 4, 9 en 12.

Samenvattend toont deze case aan dat foutdetectie via afwijkingsanalyse:

- Belangrijke technische onvolkomenheden aan het licht brengt;
- Inzicht geeft in systeemsamenhang tussen warmtepomp, GBS-instellingen en fysieke gebouwinvloeden;
- Bijdraagt aan prioritering van maatregelen zoals herinregelen, aanpassing van regeltechniek of vervanging van componenten.

6.2 Use Case 2 – BESI Duiven

6.2.1 1. Data collectie

In BESI Duiven werd in **november 2020** een nulmeting uitgevoerd. Gedurende 2021 werd data verzameld over:

- Aanvoertemperaturen per verdeelstuk;
- Energieverbruik over het kalenderjaar;
- Invloed van windrichting en zoninstraling;
- Temperatuurmetingen gekoppeld aan zonering van het gebouw;
- GBS-configuratie en manuele bedieningen via thermostaatkranen.

Er is gebruik gemaakt van zowel historische klantdata als RVO-kengetallen ter vergelijking.

6.2.2 2. Analyse

Het algoritme van CEO voorspelde aanvoertemperaturen op basis van seizoensinvloeden, zoninstraling, gebruikspatronen en comfortzones. Vergelijking vond plaats op maandniveau tussen:

- Werkelijke ingestuurde temperatuur (traditionele GBS);
- Voorspelde aanvoertemperatuur door CEO;
- Theoretisch optimale temperatuur.

Voorbeeldberekening afwijking:

$$\text{Afwijking}_{\%} = \left(\frac{T_{\text{GBS}} - T_{\text{CEO}}}{T_{\text{GBS}}}\right) \times 100\%$$

In februari bedroeg de GBS-aanvoer 83,4°C tegenover een voorspelde 58,3°C → afwijking 30,1%.

Uit grafieken bleek dat het GBS structureel te hoge temperaturen stuurt, ook in overgangsmaanden waarin dit niet nodig is.

6.2.3 3. Bevindingen

De foutdetectie bracht o.a. het volgende aan het licht:

- De traditionele regeling werkt op een **buitentemperatuurafhankelijke stooklijn**, zonder rekening te houden met zonbelasting, bezetting of thermische massa → overcompensatie.
- **Radiatoren** worden handmatig bediend, zonder terugkoppeling naar het GBS. Hierdoor wordt warmte verstookt terwijl ruimtes soms al op temperatuur zijn.
- **Vloerverwarming** wordt geregeld via een thermostaat in een *andere ruimte* → structureel verkeerde aanvoertemperatuur.
- De **luchtbehandeling en verwarming werken soms gelijktijdig**, zonder afstemming → energieverlies.
- Windinvloed: bij oostelijke en zuidelijke windrichting is zichtbaar dat LBK-regeling vaker overschakelt naar bypass-modus om overdruk te compenseren → rendementsverlies van de warmteterugwinning.

Ook hier blijkt dat de inzet van een slim algoritme zoals dat van CEO leidt tot:

- Significant lagere ingestuurde temperaturen;
- Lagere pieken (dus minder belasting op de installatie);
- Gecontroleerde regeling op basis van comfortbehoefte i.p.v. vaste klokinstellingen;
- Minder energieverlies door betere samenwerking tussen subsystemen.

6.2.4 Samenvattende conclusie

Door het toepassen van foutdetectie op basis van afwijkingen ten opzichte van voorspellingen zijn in beide use-cases technische knelpunten en energetische onvolkomenheden zichtbaar geworden die zonder deze methodiek verborgen waren gebleven.

De case van de VUB laat zien dat zelfs in een modern gebouw **een warmtepomp ineffectief wordt** bij een verkeerd ingestelde retourregeling. In combinatie met windgevoelige LBK's leidt dit tot fors energieverlies. Bij BESI is vooral de **starre GBS-regeling** de oorzaak van foutief gedrag, versterkt door manuele bediening en slechte zonering.

Deze use-cases onderstrepen het belang van:

- Valide voorspellingen (punt 1);
- Begrensde maar goed gemeten data-invoer (punt 2);
- Doelgerichte validatie (punt 3).

Cloud Energy Optimizer heeft hiermee een robuuste methode in handen voor geautomatiseerde foutdetectie die schaalbaar én praktisch toepasbaar is.

7 Verslagpunt 5: Input leveren voor visualisatie (interface WP3)

7.1 Samenvatting en doel

Binnen Werkpakket 3 draait alles om het verbeteren van comfort, gezondheid en energie-efficiëntie via **gebruikersinterfaces en feedbacksystemen**. Cloud Energy Optimizer (CEO) heeft actief bijgedragen aan dit werkpakket door zowel **praktijkdata** als **gebruikerservaringen** beschikbaar te stellen uit lopende projecten. Concreet zijn gegevens en inzichten gedeeld van:

- **MN – Zilveren Toren, Den Haag**
- **Provinciehuis Overijssel, Zwolle**

Beide gebouwen maken gebruik van de CEO-sturing, en zijn representatief voor grote kantoorgebouwen met verschillende typen gebruikers, uiteenlopende installaties en hoge eisen aan comfort. De gebruikerswensen die hieruit naar voren kwamen, zijn cruciaal geweest voor het ontwerpen van toegankelijke en effectieve visualisaties.

7.2 Gebruikerswensen: divers, persoonlijk en soms tegenstrijdig

Een van de belangrijkste bevindingen uit de praktijk is dat **gebruikersbehoeften sterk variëren per type persoon en situatie**. Waar technische installaties vaak streven naar 'objectieve' comfortwaarden, blijkt comfort in werkelijkheid **subjectief en contextafhankelijk**.

7.2.1 1. Fysiologisch verschil: man vs. vrouw

Het is wetenschappelijk vastgesteld dat vrouwen gemiddeld bij een **lagere lichaamstemperatuur** meer last ervaren van kou dan mannen. Onderzoek gepubliceerd in *Nature Climate Change* (Kingma & van Marken Lichtenbelt, 2015) toont aan dat vrouwen zich prettiger voelen bij binnentemperaturen rond de 24–25 °C, terwijl mannen vaak tevreden zijn met 21–22 °C.

Toch leidt dat niet altijd tot een hogere energievraag: veel mannen passen hun gedrag aan (bijvoorbeeld door in T-shirt te werken), terwijl vrouwen hun kleding minder snel aanpassen — mogelijk vanwege kledingvoorschriften of persoonlijke voorkeur. Interfaces moeten daarom ruimte bieden voor persoonlijke feedback, maar ook bewust zijn van deze culturele verschillen.

7.2.2 2. Reismethode beïnvloedt temperatuurbeleving

Medewerkers die **op de fiets** komen, hebben bij aankomst vaak een verhoogde lichaamstemperatuur. Zij ervaren een ruimte eerder als te warm. Daarentegen zijn collega's die vanuit een **warme auto** binnenkomen juist gebaat bij iets hogere starttemperaturen. Dit effect is tijdelijk, maar relevant tijdens piekmomenten zoals begin werkdag of na lunchpauzes.

7.2.3 3. Moment van de dag en fysieke gesteldheid

Gebruikers die vroeg beginnen, hebben vaak andere verwachtingen dan wie later op de dag binnenkomt. Vroege starters waarderen een snellere opwarming, terwijl laatkomers doorgaans flexibel zijn. Ook iemands fysieke gesteldheid speelt mee: bij ziekte of vermoeidheid neemt de gevoeligheid voor koude toe. De visualisatie moet hiermee kunnen omgaan door dagcurves te tonen of persoonlijke voorkeuren te verwerken.

7.2.4 4. Kleuren en eenvoud in interface

De **visuele opmaak** van de interface is essentieel. Kleuren roepen emotie op: rood wordt geassocieerd met gevaar of hitte, blauw met kou of rust. Heldere, zachte kleuren werken beter voor comfortvisualisaties. Felle contrasten kunnen onbedoeld stress veroorzaken.

Daarnaast geldt: **eenvoud is cruciaal**. Gebruikers verschillen sterk in achtergrond. Een technisch beheerder wil gedetailleerde grafieken, terwijl een doorsnee medewerker vaak alleen wil weten: *"Is het goed of niet?"* Daarom moeten interfaces lagen bevatten: eenvoudig aan de voorkant, met diepgang voor wie dat nodig heeft.

7.2.5 5. Gedragsverandering bij monitoring

Een opvallend patroon dat CEO heeft waargenomen is het verschil in gedrag bij **wel of niet communiceren dat monitoring plaatsvindt**:

- Als gebruikers **weten dat ze gemonitord worden**, verbeteren hun acties meteen. Thermostaten worden minder vaak aangepast, ramen blijven gesloten bij verwarming en klachten nemen af.
- Echter: dit effect **neemt na verloop van tijd af**. Gebruikers wennen aan het idee, voelen zich minder betrokken of merken geen directe terugkoppeling.

De reden is psychologisch: gedragsverandering is **reactief en tijdelijk**, tenzij deze wordt ondersteund door **continue feedback of beloning**. Alleen weten dat iets wordt gemeten is onvoldoende om blijvend gedrag te beïnvloeden. Daarom zijn visuele terugkoppelingen, zoals comfortscore-kaarten of energiebesparingsmeters, essentieel voor langdurige betrokkenheid.

7.3 Conclusie

CEO concludeert dat effectieve visualisatie veel verder gaat dan het tonen van temperatuur of verbruik. Het draait om **communicatie, herkenbaarheid en vertrouwen**. De interface moet:

- Eenvoudig en intuïtief zijn voor alle gebruikers;
- Rekening houden met persoonlijke voorkeuren en context;
- Feedback geven op een manier die gedrag positief beïnvloedt;
- Niet sturen, maar **ondersteunen**: de gebruiker blijft eigenaar van zijn of haar comfort.

De ervaringen uit Den Haag en Zwolle tonen aan dat goede visualisatie leidt tot meer acceptatie van energiesturing en uiteindelijk betere prestaties van het gebouw.

8 Verslagpunt 6: Automatisering koppeling gebouwdata

8.1 Inleiding: Architectuur & datastromen

Binnen Cloud Energy Optimizer (CEO) is een standaardarchitectuur ontwikkeld die ervoor zorgt dat gebouwdata op een consistente en schaalbare manier automatisch gekoppeld kan worden aan het optimalisatieplatform. In deze architectuur wordt onderscheid gemaakt tussen drie soorten data:

1. **Vaste data**: gegevens die zelden veranderen, zoals het aantal ketels, de capaciteit van een warmtepomp of de configuratie van verdeelstukken. Deze data worden via het configuratiegedeelte van het **CEO Portal** ingevoerd door de installateur.
2. **Historische data**: meetwaarden van installaties, ruimten en sensoren die elke 5 minuten binnenkomen. Denk aan temperaturen, energieverbruik, flows en comfortmetingen. Deze data worden via de putHistory-interface opgeslagen in de **CEO Cosmos-database**.
3. **Voorspeldata (forecast)**: op basis van historische data, gebruikersvoorkeuren en weersverwachting genereert CEO elke 30 minuten een **nieuwe 24-uurs forecast**. Alleen de **actuele 'running forecast'** wordt opgeslagen, om databelasting en kosten te beperken. De forecast wordt via de getForecast-interface beschikbaar gesteld aan het gebouwbeheersysteem (GBS).

De **Convertor** vormt de spil tussen gebouw en algoritme. Het vertaalt ruwe GBS-data naar gestandaardiseerde JSON-structuren en voert die door naar de AI-algoritmen in de Microsoft Azure Cloud. Tegelijkertijd zet hij berekende forecasts om naar stuursignalen voor het GBS.

8.1.1 Wat is JSON, en waarom gebruiken we het?

JSON staat voor **JavaScript Object Notation**. Het is een **lichtgewicht gegevensformaat** dat ideaal is voor het verzenden van gestructureerde data tussen systemen via API's (Application Programming Interfaces). JSON is makkelijk leesbaar door zowel mensen als machines, en wordt breed ondersteund binnen vrijwel alle moderne softwareplatformen.

Een voorbeeld van een JSON-bericht (vereenvoudigd):

json

KopiërenBewerken

```
{
  "ProjectId": "ceo123456",
  "Timestamp": "2025-01-01T08:00:00",
  "Boilers": [
    {
      "Id": 1,
      "SupT_oC": 68.5,
      "RetT_oC": 58.3,
      "Gas_m3": 3.1,
      "Failure_Bit": 0
    }
  ]
}
```

In dit bericht worden keteldata van een gebouw verstuurd op een specifiek moment.

8.1.2 Opbouw van de datastructuren

Zoals je ziet in het overzicht **"Verkorte JSON Structuuroverzicht"**, is elk JSON-object opgebouwd uit een aantal vaste onderdelen:

Type	Groep	Label	Eenheid	Omschrijving
H	Boilers	SupT_oC	°C	Aanvoertemperatuur ketel
H	Boilers	RetT_oC	°C	Retourtemperatuur ketel
H	Boilers	Gas_m3	m^3	Verbruik gas
H	Boilers	Failure_Bit	Bit	Foutcode ketel

- De kolom **Type** geeft aan of het om Historische data of Forecast data gaat.
- Elke waarde wordt aangeleverd in een standaard **eenheid** (bijvoorbeeld °C, m^3, kWh).
- Elk object bevat een unieke Id, die refereert aan de fysieke component in het gebouw.
- De Timestamp bepaalt het moment van de meting of aansturing.

Er zijn momenteel ruim **100 verschillende datavelden** gedefinieerd, verspreid over componenten zoals boilers, buffers, warmtepompen, LBK's, zones, sensoren, meetinstrumenten en netwerken (HT, LT).

8.1.3 Cloudplatform en verwerking

Alle verwerking vindt plaats binnen de **Microsoft Azure-cloud**. Er is gekozen voor een **serverless architectuur** die werkt met:

- **Azure Functions** voor het verwerken van binnenkomende en uitgaande JSON-berichten;
- **Azure IoT Hub** als veilige gateway voor devices en sensoren;
- **Azure Cosmos DB** als schaalbare opslag voor alle inkomende en verwerkte data;
- **Azure Machine Learning** voor het draaien van de voorspellende algoritmen.

Omdat het platform serverless is, zijn er **geen vaste servers nodig**. De verwerkingscapaciteit schaalt automatisch mee met het aantal gebouwen of datapunten, wat resulteert in lagere kosten en hogere betrouwbaarheid.

8.1.4 Schaalbaarheid en eenvoud

Een van de grootste voordelen van deze opzet is de **eenvoudige uitbreidbaarheid**:

- Wil je een nieuw type sensor toevoegen (bijvoorbeeld fijnstof of geluid)? Dan hoeft alleen een nieuw JSON-object met de juiste velden te worden gedefinieerd.
- Nieuwe modules kunnen deze data **automatisch detecteren**, doordat de Convertor elk object herkent op basis van zijn Id en Label.
- Deze modules werken **autonoom** en communiceren alleen wanneer nodig. Daardoor blijft de **dataload laag** en het systeem snel en efficiënt.

Bovendien kunnen gebouwen onafhankelijk functioneren: elke site heeft een eigen ProjectId, en alle data zijn gescheiden verwerkt en opgeslagen. Dat maakt het platform veilig én geschikt voor multi-tenant gebruik.

8.1.5 Conclusie

Met de automatisering van gebouwdata via gestandaardiseerde JSON-structuren, een slimme convertor en serverless Azure-technologie heeft Cloud Energy Optimizer een **krachtige, flexibele en schaalbare oplossing** gerealiseerd.

Deze oplossing maakt het mogelijk om:

- met minimale configuratie nieuwe gebouwen aan te sluiten;
- foutbronnen en comfortafwijkingen automatisch te detecteren;
- gebouwdata gestructureerd op te slaan, analyseren en optimaliseren;
- en tegelijkertijd de datalast en beheerkosten laag te houden.

De gekozen aanpak biedt toekomstvastheid, want door het modulaire ontwerp kunnen **nieuwe datavelden of componenten zonder impact worden toegevoegd**, en integraties met andere software of systemen verlopen snel via de bestaande API-structuur.

Cloud Energy Optimizer is hiermee klaar voor grootschalige uitrol in uiteenlopende omgevingen — van kleine zorginstellingen tot complexe universiteitscampussen.

9 Verslagpunt 7: Opschaling en robuustheidstests

9.1 Inleiding: waarom schaalbaarheid essentieel is

De ambitie van het B4B-project is om slimme, zelflerende gebouwaansturing breed toepasbaar te maken. Niet als maatwerk voor een paar pilotgebouwen, maar als schaalbare oplossing voor **tienduizenden utiliteitsgebouwen** in Nederland en daarbuiten. Cloud Energy Optimizer (CEO) heeft hier specifiek op ingezet door de volledige architectuur, het algoritme én de implementatie-aanpak te ontwerpen met opschaling in gedachten.

Het is gebleken dat de **oorspronkelijke methodieken uit het onderzoekscollectief niet schaalbaar zijn**. Ze veronderstellen te veel detaildata, maatwerkmodellering en langdurige installatie- en afstemtijd. Dat is in de praktijk niet haalbaar voor bredere uitrol.

9.1.1 Eenvoud als voorwaarde voor opschaling

CEO heeft daarom gekozen voor een **radicaal vereenvoudigde aanpak**, die werkt op basis van:

- Een **beperkt aantal datapunten**;
- Slimme detectie van beschikbare data vóórdat het algoritme wordt geactiveerd;
- En een architectuur die modulair werkt, zodat alleen data die beschikbaar is wordt benut.

Dit is mogelijk gemaakt door een geheel **nieuw algoritme**, dat in **maart 2024 live is gebracht** (zie roadmap slide: *Release V6 Algoritme*). Het algoritme werkt op basis van een gelaagd model: in stap 1 wordt bekeken welke data beschikbaar is, en pas daarna wordt het algoritme op maat geconfigureerd. Daarmee vervalt de noodzaak voor voorafgaande datamodellering, en kunnen ook gebouwen met beperkte sensordata aansluiten.

9.1.2 Automatisering met wizard en koppelscripts

Om deze aanpak nog verder te versnellen is in **2024** een speciale **wizard** ontwikkeld. Deze wizard stelt installateurs en gebruikers in staat om via een eenvoudige webinterface een gebouw aan te melden. Door het invullen van een vragenlijst (zoals aantal zones, type energiebronnen, sensoren en regeltechniek) genereert het systeem automatisch:

- De benodigde **koppelpagina's**;
- Een **projectstructuur** in de database;
- Een op maat gemaakte set van verwachte dataprofielen.

Hierdoor wordt de foutmarge aanzienlijk verlaagd en is ook direct inzichtelijk wat er ontbreekt om een gebouw volledig te optimaliseren. Dankzij de standaardisatie in JSON-formaat (zie punt 6) en automatische herkenning van componenten, kan een gebouw tegenwoordig **binnen één werkdag** live worden gezet, waar dit voorheen weken of maanden kostte.

9.1.3 Van 240 uur naar 16 uur implementatietijd

Een belangrijke uitkomst van deze schaalbaarheidsaanpak is de drastische reductie in implementatietijd. Waar een volledig traject in de beginfase van CEO gemiddeld **240 uur** in beslag nam (inclusief dataverzameling, modellering, koppeling, testen en oplevering), is dat dankzij:

- het nieuwe algoritme,
- de wizard,
- en automatische datastructurering

teruggebracht naar **gemiddeld 16 uur** per gebouw. Dit maakt het mogelijk om niet alleen sneller op te schalen, maar ook tegen aanzienlijk lagere kosten per site.

Verhoogde betrouwbaarheid en robuustheid

Parallel aan deze versimpeling is het systeem getest op **robuustheid**:

- De algoritmen detecteren automatisch afwijkingen of ontbrekende waarden en schakelen tijdelijk over op fallback-logica;
- Elke component (bijvoorbeeld een warmtepomp of LBK) draait in een **zelfstandig datablok**, waardoor uitval van één sensor de rest van het systeem niet beïnvloedt;
- Door de combinatie van historische en actuele forecastdata is het systeem bestand tegen fluctuaties en dataverlies.

Deze aanpak zorgt ervoor dat CEO niet alleen schaalbaar is, maar ook geschikt is voor toepassingen in gebouwen met kritieke functies (zoals zorginstellingen of datacenters).

9.1.4 Verwijzing naar de roadmap

In de **B4B Roadmap 2024–2025** is de opschaling duidelijk terug te zien in onder andere:

- **Maart 2024 – Release V6 Algoritme** (fundamentele vernieuwing ten behoeve van schaalbaarheid)
- **Mei 2024 – Installatie visualisatie** (visuele wizard)
- **November 2024 – Foutdetectie V1 + rapportage** (terugkoppeling gebruikersdata)
- **2025 – Portfolio congestiemodel & V8 algoritme** (vooruitblik op multi-site optimalisatie)

9.1.5 Samenvattende conclusie

Cloud Energy Optimizer heeft in dit kader een **volledig schaalbare architectuur gerealiseerd**, gebaseerd op standaardisatie, automatische datadetectie en eenvoudige implementatie. Door afstand te nemen van oorspronkelijke, data-intensieve methodieken is het nu mogelijk om gebouwen met minimale middelen aan te sluiten en te optimaliseren.

Met de wizard en het nieuwe algoritme zijn implementatietijd en foutmarge drastisch gereduceerd:

- **Van 240 naar 16 uur per gebouw**
- **Van maatwerk naar standaardkoppeling**
- **Van afhankelijkheid naar autonomie**

Daarmee is de weg vrijgemaakt voor grootschalige toepassing van smart building technologie, op een manier die wérkt in de praktijk.

10 Verslagpunt 8: Bijdrage aan open bibliotheek met DBN-modellen

10.1 Wat zijn DBN-modellen?

DBN staat voor **Dynamic Bayesian Network** (dynamisch Bayesiaans netwerk). Dit is een type **probabilistisch model** dat gebruikt wordt om systemen te modelleren die zich ontwikkelen over de tijd. Het combineert elementen van kansrekening, statistiek en causaliteit om voorspellingen te doen of afwijkingen te detecteren.

In een DBN worden variabelen (zoals temperatuur, stroomverbruik of gebruikersgedrag) weergegeven als knopen in een netwerk. De verbindingen tussen deze knopen tonen de onderlinge afhankelijkheden. In tegenstelling tot statische Bayesiaanse netwerken houden DBN's ook rekening met hoe die relaties veranderen over de tijd (tijdstappen of 'slices').

10.1.1 Waarom DBN's voor gebouwbeheer?

Voor een gebouw zijn veel factoren onderling afhankelijk en tijdsgevoelig:

- Als de buitentemperatuur stijgt, daalt de warmtevraag.
- Als de zoninstraling toeneemt, wordt het binnen warmer – tenzij er zonwering actief is.
- Als er mensen aanwezig zijn, verandert het CO_2-niveau en de warmteproductie.

Een DBN is bij uitstek geschikt om deze relaties **op een slimme, lerende manier** te modelleren. Ze maken het mogelijk om te:

- **Voorspellen** wat er gaat gebeuren (bijv. temperatuur over 1 uur);
- **Detecteren** of iets niet klopt (bijv. ruimte te koud ondanks actieve verwarming);
- **Adviseren** wat de beste sturing is (bijv. ketel aan, ventilatie omhoog).

Cloud Energy Optimizer gebruikt vereenvoudigde en geoptimaliseerde DBN-modellen als onderdeel van haar algoritmische kern, met name binnen de **forecastmodule en foutdetectielaag**.

10.1.2 Vijf voorbeeldmodellen binnen Cloud Energy Optimizer

Hieronder volgen vijf voorbeelden van DBN-modellen die momenteel zijn gebruikt of in ontwikkeling zijn binnen het platform:

10.1.2.1 1. Zone Comfort Model

Doel: voorspellen van de temperatuur in een ruimte, gegeven de huidige sturing, bezetting en weersomstandigheden.

Variabelen:

- Inblaastemperatuur (AirT_oC)
- Buitentemperatuur (OutT_oC)
- Aanwezigheid (Presence_Bit)
- Zonkracht (Rad_W_m2)
- Comforttemperatuur (AirDeT_oC)

Gebruik: Dit model wordt gebruikt om te voorspellen of een ruimte het gewenste comfort zal bereiken binnen een bepaald tijdsvenster (bijv. 30 minuten), en of de huidige aansturing toereikend is.

10.1.2.2 2. Warmtepomp Efficiëntiemodel

Doel: voorspellen van het rendement van de warmtepomp (COP) onder verschillende condities.

Variabelen:

- Retourtemperatuur (HRetT_oC)
- Aanvoertemperatuur (HSupT_oC)
- Modus (verwarmen/koelen) (Mod_Bit)
- Buitenluchttemperatuur (OutT_oC)
- Energieverbruik (Electra_kWh)

Gebruik: Detecteert automatisch wanneer de warmtepomp buiten zijn optimale werkgebied draait (bijv. door te hoge retourtemperatuur) en adviseert herconfiguratie of back-upverwarming.

10.1.2.3 3. Zoninstralingsinvloedmodel

Doel: voorspellen van de invloed van zonlicht op de binnentemperatuur in combinatie met zonwering en raamstand.

Variabelen:

- Zonkracht (Rad_W_m2)
- Ramen open/dicht (Roof_Bit)
- Zonwering actief (SolarScreen_Bit)
- Inkomende warmtelast
- Comfortafwijking (Temp_oC - AirDeT_oC)

Gebruik: Helpt om passieve verwarming of oververhitting door zon tijdig te detecteren en corrigeert de aansturing van koeling of zonwering.

10.1.2.4 4. Gebruikspatroonmodel

Doel: voorspellen wanneer een ruimte bezet is en hoeveel mensen aanwezig zijn op basis van historische patronen.

Variabelen:

- Aanwezigheidsensor (Presence_Bit)
- Weekdag/tijdstip
- Kalenderdata (schoolvakanties, feestdagen)
- Handmatige overrules

Gebruik: Wordt ingezet om verwarming en ventilatie **voorafgaand aan gebruik** op te starten of juist tijdig af te schakelen.

10.1.2.5 5. Boilerbelasting en verbruikspiekmodel

Doel: voorspellen van verbruikspieken op basis van weersomstandigheden en vraagontwikkeling.

Variabelen:

- Buitenluchttemperatuur (OutT_oC)
- Aanvoertemperatuur (SupT_oC)
- Gasverbruik (Gas_m3)
- Historisch verbruik op vergelijkbare dagen

Gebruik: Dient voor load-balancing, vooral in gebouwen met stadswarmte of beperkte broncapaciteit, en voorkomt onnodige piekbelasting.

10.2 Conclusie

Dynamic Bayesian Networks zijn een krachtige basis voor foutdetectie, voorspellingen en intelligente sturing binnen het Cloud Energy Optimizer-platform. Door gebruik te maken van **vereenvoudigde, praktijkgerichte DBN's** kan het systeem slim inspelen op het gedrag van het gebouw en zijn gebruikers, zonder afhankelijk te zijn van grote hoeveelheden data of maatwerk.

De vijf gepresenteerde modellen vormen de kern van de huidige toepassingen en zijn flexibel genoeg om in de toekomst verder te verfijnen. Door deze kennis via een open bibliotheek beschikbaar te stellen, draagt CEO bij aan **versnelde innovatie, samenwerking en opschaling binnen de markt voor slimme gebouwen**.

11 DBN-model 1 – Zone Comfort Model

11.1 Doel van het model

Dit **Dynamic Bayesian Network** is bedoeld om te voorspellen of de **temperatuur in een ruimte (zone)** binnen een gewenst tijdsvenster het comfortniveau bereikt. Het model houdt rekening met omgevingsfactoren, bezetting en het gedrag van het systeem.

11.2 Stap 1: Definiëren van variabelen

Variabele	Omschrijving	Type	Domein/Eenheid
AirT_oC(t)	Actuele gemeten temperatuur in de zone op tijdstip t	Observabel	continue (°C)
AirDeT_oC(t)	Gewenste temperatuur op tijdstip t	Observabel	continue (°C)
Presence_Bit(t)	Aanwezigheid in de ruimte (0 = afwezig, 1 = aanwezig)	Binaire node	{0, 1}
Rad_W_m2(t)	Zoninstraling op gebouwdeel in W/m²	Observabel	continue (W/m²)
OutT_oC(t)	Buitentemperatuur	Observabel	continue (°C)
VentActive_Bit(t)	Status ventilatie/inblaas in zone (0 = uit, 1 = aan)	Binaire node	{0, 1}
DeltaT(t+1)	Verwachte temperatuurverandering in de volgende tijdstap	Afgeleid	continue (°C)
ComfortAchieved(t+1)	Krijgt zone de gewenste temperatuur in t+1	Doelnode	{ja, nee}

11.3 Stap 2: Dynamiek en afhankelijkheden (structuur)

De tijdsdimensie in dit DBN bestaat uit **tijdstappen van 5 minuten** (in lijn met CEO-datalogging). De relaties tussen de knopen worden als volgt gemodelleerd:

```
t:  [AirT_oC(t)] ------+
                       |
                       v
              [DeltaT(t+1)] ----> [AirT_oC(t+1)] ----> [ComfortAchieved(t+1)]
             ^     ^     ^                        ^
             |     |     |                        |
             |     |     |                        |
   [Rad_W_m2(t)] [OutT_oC(t)] [Presence_Bit(t)]   |
                       |                          |
                       +--> [VentActive_Bit(t)] --+
```

11.4 Stap 3: Rekenkundige relaties (voorbeeld)

We maken een vereenvoudigd **regressiemodel binnen het DBN** voor DeltaT(t+1) als lineaire combinatie van inputs:

Formule:

Formule:

$$\Delta T_{t+1} = \alpha_1 \cdot (AirDeT_oC_t - AirT_oC_t) + \alpha_2 \cdot Presence_t + \alpha_3 \cdot Rad_t + \alpha_4 \cdot Vent_t + \alpha_5 \cdot (OutT_oC_t - AirT_oC_t)$$

11.4.1 Parameters (voorbeeldwaarden):

Coëfficiënt	Beschrijving	Waarde (initieel)
α_1	Correctievermogen systeem	0.3
α_2	Warmtebijdrage door aanwezigheid	0.1
α_3	Warmte-invloed van zoninstraling	0.05
α_4	Effect ventilatie/inblaas	0.2
α_5	Invloed temperatuurverschil buiten/binnen	0.15

11.4.2 Voorbeeldberekening:

Stel:

- AirT_oC(t) = 20.0 °C
- AirDeT_oC(t) = 22.0 °C
- Presence_Bit(t) = 1
- Rad_W_m2(t) = 300
- VentActive_Bit(t) = 1
- OutT_oC(t) = 12.0 °C

$$\Delta T = 0.3 \cdot (22 - 20) + 0.1 \cdot 1 + 0.05 \cdot 300 + 0.2 \cdot 1 + 0.15 \cdot (12 - 20)$$

$$\Delta T = 0.3 \cdot 2 + 0.1 + 15 + 0.2 - 1.2 = 0.6 + 0.1 + 15 + 0.2 - 1.2 = 14.7$$

→ Verwachte temperatuurstijging binnen 5 minuten = **+1.47°C**

→ Nieuwe verwachte temperatuur = 20.0 + 1.47 = **21.47°C**

→ ComfortAchieved = Nee (want doel = 22.0°C)

11.5 Stap 4: Outputnode ComfortAchieved(t+1)

Deze node voorspelt of de temperatuur **binnen het bereik van +/− 0.5°C van de gewenste waarde** komt:

$$ComfortAchieved(t+1) = \begin{cases} \text{Ja}, & \text{als } |AirT_{t+1} - AirDeT_t| \leq 0.5 \\ \text{Nee}, & \text{anders} \end{cases}$$

In bovenstaand voorbeeld:
21.47°C is nog **0.53°C onder** de doelwaarde → dus: Nee.

11.5.1 Toepassing in praktijk

- De uitkomst ComfortAchieved = Nee kan automatisch leiden tot aanpassing van de aansturing (bijv. ventilatie verhogen, nachtkoeling starten).
- Als DeltaT structureel te laag is → **foutmelding of adviesregel** ("zone haalt comfort niet ondanks verwarming actief").
- Kan gecombineerd worden met gebruikersfeedback om het model verder te leren (bijv. comfortscore via app).

11.5.2 Adaptiviteit

CEO past de coëfficiënten (α_1 t/m α_5) **automatisch aan op basis van historische afwijkingen**. Als bijvoorbeeld blijkt dat zoninstraling in een bepaalde ruimte nauwelijks effect heeft, daalt α_3 in waarde.

12 DBN-model 2 – Warmtepomp Efficiëntiemodel

12.1 Doel van het model

Dit model is ontworpen om de **efficiëntie van de warmtepomp (COP)** te voorspellen en bewaken. Het stelt vast of de warmtepomp binnen het optimale werkgebied functioneert en detecteert afwijkend gedrag, zoals inefficiëntie door te hoge retourtemperatuur, ongeschikte weerscondities of verkeerde modusinzet (koelen/verwarmen).

12.1.1 Stap 1: Definiëren van variabelen

Variabele	Omschrijving	Type	Domein/Eenheid
HSupT_oC(t)	Aanvoertemperatuur warme zijde warmtepomp	Observabel	continue (°C)
HRetT_oC(t)	Retourtemperatuur warme zijde warmtepomp	Observabel	continue (°C)
Mod_Bit(t)	Modus warmtepomp (1 = verwarmen, 0 = koelen)	Binaire node	{0, 1}
OutT_oC(t)	Buitentemperatuur	Observabel	continue (°C)
Electra_kWh(t)	Elektrisch verbruik warmtepomp per tijdstap	Observabel	continue (kWh)
Thermal_kWh(t)	Thermisch geleverde energie per tijdstap	Afgeleid	continue (kWh)
COP(t)	Coëfficiënt of Performance (rendement warmtepomp)	Afgeleid	continue (eenheidsloos)
COP_Expected(t)	Verwachte COP onder ideale omstandigheden	Afgeleid	continue (–)
COP_Deviation(t)	Afwijking t.o.v. verwacht rendement	Afgeleid	continue (–)
EfficiencyOK(t)	Is warmtepomp efficiënt?	Doelnode	{ja, nee}

12.1.2 Stap 2: Dynamiek en afhankelijkheden

```
HSupT_oC(t) ----+
                |
HRetT_oC(t) ----/              +--> Thermal_kWh(t)
                |              |
OutT_oC(t) -----/              +---> COP(t) ------+
                |                                 |
Mod_Bit(t) -----+                                 +---> COP_Deviation(t) ---> EfficiencyOK(t)
                                                 ^
Electra_kWh(t) ---------------------------------+
```

12.1.3 Stap 3: Rekenkundige relaties

12.1.3.1 1. Thermisch geleverde energie

$$Thermal_{kWh}(t) = c_p \cdot \rho \cdot Q \cdot \Delta T$$

Waarbij:

- cpc_pcp = soortelijke warmte water ≈ 4.18 kJ/kg·°C
- ρ\rhoρ = dichtheid water ≈ 1.0 kg/L
- QQQ = debiet (liter/sec) – *indirect gemeten of geschat*
- $\Delta T = HSupT_oC(t) - HRetT_oC(t)$ in °C

Aangezien Q vaak onbekend is, nemen we:

$$Thermal_{kWh}(t) \approx k \cdot \Delta T$$

12.1.3.2 2. COP (Coëfficiënt of Performance)

$$COP(t) = \frac{Thermal_{kWh}(t)}{Electra_{kWh}(t)}$$

12.1.3.3 3. Verwachte COP onder ideale condities

Op basis van een empirisch verband:

$$COP_{Expected}(t) = a - b \cdot (HSupT_oC(t) - OutT_oC(t))$$

Met:

- a=5.5a = 5.5a=5.5 (ideale COP bij klein delta T)
- b=0.05b = 0.05b=0.05 (hoeveel COP daalt per °C verschil)

Voorbeeld:

- HSupT_oC = 45°C
- OutT_oC = 10°C → ΔT=35\Delta T = 35ΔT=35
  → $COP_{Expected} = 5.5 - 0.05 \cdot 35 = 3.75$

12.1.3.4 4. Afwijking en besluitlogica

$$COP_{Deviation}(t) = COP(t) - COP_{Expected}(t)$$

$$EfficiencyOK(t) = \begin{cases} \text{Ja}, & \text{als } COP_{Deviation}(t) > -0.5 \\ \text{Nee}, & \text{als } COP_{Deviation}(t) \leq -0.5 \end{cases}$$

12.1.3.5 Voorbeeldberekening

Stel:

- $HSupT_oC = 45, HRetT_oC = 35 \rightarrow \Delta T = 10$
- $Electra_{kWh} = 2.0$
- $Thermal_{kWh} = 0.25 \cdot 10 = 2.5$
- $COP = 2.5/2.0 = 1.25$
- $OutT_oC = 5 \rightarrow COP_{Expected} = 5.5 - 0.05 \cdot (45 - 5) = 3.5$
- $COP_{Deviation} = 1.25 - 3.5 = -2.25$

→ **EfficiencyOK = Nee** (te grote afwijking)

12.1.4 Adaptiviteit en alarmgeneratie

- **Lerend vermogen**: het model wordt getraind op historische data. Als bij bepaalde buitentemperaturen de werkelijke COP structureel lager ligt, worden de verwachte waarden aangepast.
- **Foutmeldingen**: wanneer EfficiencyOK = 'Nee' gedurende meerdere cycli, wordt een **automatische notificatie of aanpassing van de sturing geactiveerd**.
- **Koppeling met andere modellen**: als hoge retourtemperaturen optreden (uit model 1 of GBS), dan wordt dat meegenomen als *causale factor*.

12.1.5 Toepassing in praktijk

- Het model draait in de Azure-omgeving binnen een **serverless functie** (Azure Functions).
- De input komt via de JSON-koppeling met GBS (zie punt 6).
- De output wordt opgeslagen in **Cosmos DB** en visueel weergegeven via het **CEO dashboard**.

Bij afwijkingen worden dashboards rood gemarkeerd en suggesties gedaan:

- Check bypass-klep of driewegklep
- Controleer warmtevraag
- Pas stooklijn of setpoints aan

~~Wil je dat ik nu doorga met **model 3: Zoninstralingsinvloedmodel**? Dat model is perfect om te koppelen aan comfort- en energieverbruikspatronen.~~

13 DBN-model 3 – Zoninstralingsinvloedmodel

13.1 Doel van het model

Dit **Dynamic Bayesian Network** (DBN) is ontwikkeld om te voorspellen **hoeveel invloed zoninstraling heeft op de temperatuur in een ruimte**. Het doel is om tijdig te anticiperen op oververhitting of juist gratis verwarming, afhankelijk van zonstand, gebouworiëntatie en instellingen zoals zonwering en raamstand.

Het model stelt vast of er actie nodig is (koelen of zonwering activeren) en levert input aan comfortoptimalisatie en energie-efficiëntie.

13.1.1 Stap 1: Variabelen

Variabele	Omschrijving	Type	Domein/Eenheid
Rad_W_m2(t)	Zoninstraling in W/m²	Observabel	continue
Roof_Bit(t)	Ramen open/dicht (1=open, 0=dicht)	Binaire node	{0, 1}
SolarScreen_Bit(t)	Zonwering actief (1=actief, 0=niet actief)	Binaire node	{0, 1}
Presence_Bit(t)	Aanwezigheid in ruimte	Binaire node	{0, 1}
Temp_oC(t)	Huidige temperatuur in de ruimte	Observabel	continue (°C)
OutT_oC(t)	Buitentemperatuur	Observabel	continue (°C)
SolarGain_oC(t+1)	Geschatte temperatuurstijging door zoninstraling	Afgeleid	continue (°C)
ComfortEffect(t+1)	Invloed zon op comfort in volgende tijdstap	Doelnode	{positief, neutraal, negatief}

13.1.2 Stap 2: Structuur van het netwerk

```
                Rad_W_m2(t)
                      |
                      v
Temp_oC(t) ---> SolarGain_oC(t+1) ---> ComfortEffect(t+1)
                      ^       ^       ^
Roof_Bit(t) --------+       |       |
SolarScreen_Bit(t)--+       |       |
OutT_oC(t) -----------------+       |
Presence_Bit(t) ------------------+
```

13.1.3 Stap 3: Rekenkundige relaties

13.1.3.1 1. Berekening zonnewarmte-effect (SolarGain_oC)

We modelleren de temperatuurtoename in de ruimte als:

SolarGain(t+1) = \beta_1 \cdot Rad_W_m2(t) \cdot \phi(t)

Waarbij:

- Rad_W_m2(t) = actuele zoninstraling
- φ(t) = reductiefactor (afhankelijk van open ramen, zonwering, aanwezigheid)

13.1.3.2 2. Reductiefactor ϕ(t)

Deze bepaalt hoe groot het effect van zon is in de ruimte:

$$\phi(t) = (1 - \gamma_1 \cdot Roof_Bit(t)) \cdot (1 - \gamma_2 \cdot SolarScreen_Bit(t)) \cdot (1 + \gamma_3 \cdot Presence_Bit(t))$$

Met voorbeeldwaarden:

- γ1=0.5\gamma_1 = 0.5γ1=0.5: open ramen halveren het effect
- γ2=0.7\gamma_2 = 0.7γ2=0.7: actieve zonwering schermt 70% af
- γ3=0.2\gamma_3 = 0.2γ3=0.2: aanwezigheid verhoogt interne warmtelast

13.1.3.3 3. Comfortimpact bepalen

We bepalen het verschil met de comfortdoelstelling:

$$\Delta T = Temp_oC(t) + SolarGain(t+1) - AirDeT_oC(t)$$

Op basis van de waarde van ΔT\Delta TΔT:

Waarde van ΔT	**ComfortEffect**
ΔT < -1.0°C	positief (gratis opwarming)
-1.0 ≤ ΔT ≤ 1.0	neutraal
ΔT > 1.0°C	negatief (dreigende oververhitting)

13.1.3.4 Voorbeeldberekening

Stel:

- Rad_W_m2 = 600
- Roof_Bit = 0 (ramen dicht)
- SolarScreen_Bit = 1 (zonwering aan)
- Presence_Bit = 1
- Temp_oC = 22.0
- AirDeT_oC = 23.0

Berekening reductiefactor:

$$\phi = (1 - 0.5 \cdot 0) \cdot (1 - 0.7 \cdot 1) \cdot (1 + 0.2 \cdot 1) = 1 \cdot 0.3 \cdot 1.2 = 0.36$$

$$SolarGain = 0.002 \cdot 600 \cdot 0.36 = 0.432^{\circ}C$$

$$\Delta T = 22.0 + 0.43 - 23.0 = -0.57^{\circ}C \rightarrow ComfortEffect = neutraal$$

Zonder zonwering (SolarScreen_Bit = 0):

13.1.4 Schematische werking van het Zoninstralingsinvloedmodel

13.1.5 Stap 1: Bereken de reductiefactor ϕ

Deze bepaalt hoeveel van de zonnewarmte daadwerkelijk invloed heeft op de ruimte:

Situatie	Berekening	Resultaat
Zonwering **niet actief**	1−0.7·0=1.01 - 0.7 \cdot 0 = 1.01−0.7·0=1.0	
Ramen **gesloten**	1−0.5·0=1.01 - 0.5 \cdot 0 = 1.01−0.5·0=1.0	
Aanwezigheid **aanwezig**	1+0.2·1=1.21 + 0.2 \cdot 1 = 1.21+0.2·1=1.2	
Totale φ	1.0·1.0·1.21.0 \cdot 1.0 \cdot 1.21.0·1.0·1.2	**1.2**

13.1.6 Stap 2: Bereken de verwachte temperatuurstijging (SolarGain)

$$\text{SolarGain} = 0.002 \cdot Rad_{W/m^2} \cdot \phi$$

Bijvoorbeeld:

- Zoninstraling = 600 W/m²
- φ = 1.2
- SolarGain = $0.002 \cdot 600 \cdot 1.2 = 1.44^\circ C$

13.1.7 Stap 3: Bepaal de comfortimpact (ΔT)

$$\Delta T = Temp_{oC} + SolarGain - AirDeT_{oC}$$

Bijvoorbeeld:

- Temp_oC = 22.0
- SolarGain = 1.44
- AirDeT_oC = 23.0
- $\Delta T = 22.0 + 1.44 - 23.0 = +0.44^\circ C$

13.1.8 Stap 4: Interpreteer de ComfortEffect

ΔT (verschil met gewenste temperatuur)	**ComfortEffect**
ΔT < -1.0°C	**Positief**
-1.0°C ≤ ΔT ≤ +1.0°C	**Neutraal**
ΔT > +1.0°C	**Negatief**

In bovenstaand voorbeeld: ΔT = +0.44°C → **Neutraal**

13.1.9 Alternatief scenario: zonder zonwering én ramen open

Input	**Waarde**
SolarScreen_Bit = 0	Geen afscherming
Roof_Bit = 1	Ramen open
φ-herberekening:	$(1 - 0.5 \cdot 1) \cdot (1 - 0.7 \cdot 0) \cdot (1 + 0.2 \cdot 1) = 0.5 \cdot 1.0 \cdot 1.2 = 0.6$
SolarGain	$0.002 \cdot 600 \cdot 0.6 = 0.72°C$

13.1.10 Praktische toepassingen

Detectie	**Actie**
ComfortEffect = Negatief	Activeer zonwering of koeling automatisch
ComfortEffect = Positief	Schakel verwarming tijdelijk uit (gratis warmte benutten)
ComfortEffect = Neutraal	Geen actie vereist, comfort binnen grenswaarden

Het model voorkomt zo **comfortklachten of onnodig energieverbruik** zonder extra meetpunten.

13.1.11 Adaptieve werking

- **Ruimteherkenning**: leert welke zones gevoelig zijn voor zoninstraling
- **Oriëntatiebewust**: stemt zonmodellen af op gevelrichting
- **Responsietijd**: past zich aan op hoe snel de temperatuur verandert

Feedback van gebruikers of gemeten afwijkingen wordt gebruikt om parameters zoals β\betaβ en γ\gammaγ automatisch bij te stellen.

13.1.12 Conclusie.

Het Zon instralings invloedmodel van CEO maakt:

- Comfort voorspellen en optimaliseren mogelijk
- Slim gebruik van gratis zonnewarmte
- Oververhitting voorkomen zonder koelen
- Schakelingen eenvoudiger en energiezuiniger

Een essentieel hulpmiddel voor duurzame en voorspellende klimaatbeheersing.

14 DBN-model 4 – Gebruikspatroonmodel

14.1 Doel van het model

Het **Gebruikspatroonmodel** voorspelt of een ruimte op een bepaald tijdstip **waarschijnlijk bezet is**, op basis van historische data, weekpatronen, kalenderinformatie en seizoensinvloeden. Hiermee kan het systeem vroegtijdig starten met verwarmen of ventileren, en energie besparen door tijdig af te schakelen.

14.1.1 Stap 1: Variabelen

Variabele	Omschrijving	Type	Domein/Eenheid
Presence_Bit(t)	Gemeten aanwezigheid in zone (0 = afwezig, 1 = aanwezig)	Observabel	{0, 1}
Weekday_Str(t)	Dag van de week (maandag t/m zondag)	Categorisch	{"Mon", ..., "Sun"}
TimeSlot(t)	Tijdvak (bijv. 08:00–08:15)	Categorisch	96 blokken/dag
Holiday_Bit(t)	Vakantie- of feestdag (0 = nee, 1 = ja)	Binaire node	{0, 1}
ManualOverride_Bit(t)	Handmatige bediening gemeten (bijv. verlichting aan/uit)	Binaire node	{0, 1}
ZoneType_Str(t)	Type ruimte (kantoor, vergaderzaal, open werkvloer)	Categorisch	{"office", ...}
ExpectedPresence(t+1)	Verwachte bezetting volgende tijdvak	Doelnode	{0, 1}

14.1.2 Stap 2: Netwerkstructuur

```
[Weekday_Str(t)] ------+
[TimeSlot(t)] ---------+----> ExpectedPresence(t+1)
[Holiday_Bit(t)] ------+
[ZoneType_Str(t)] -----+
[Presence_Bit(t)] -----+
[ManualOverride_Bit(t)]--+
```

14.1.3 Stap 3: Rekenkundige relaties

We gebruiken een **probabilistische kansmodel** op basis van conditional probabilities die voortkomen uit historische aanwezigheid:

$$P(ExpectedPresence = 1 | Weekday, TimeSlot, ZoneType, Holiday) = f_{\text{history}}$$

Deze wordt gecombineerd met recent gedrag:

$$Adjustment = \lambda_1 \cdot Presence_{t-1} + \lambda_2 \cdot ManualOverride_{t-1}$$

En het totaalresultaat:

$$FinalProbability = BaseProbability + Adjustment$$

14.1.4 Voorbeeld (empirisch bepaald):

- P(Presence op dinsdag 09:15 in kantoorzone) = 0.75
- Presence gisteren in zelfde tijdslot = 1 → +0.10
- Manual override gisteren = 1 → +0.05 → FinalProbability = 0.75 + 0.10 + 0.05 = **0.90**

Beslissing:

$$ExpectedPresence = \begin{cases} 1 & \text{als } FinalProbability \geq 0.6 \\ 0 & \text{anders} \end{cases}$$

14.1.5 Voorbeeldscenario

Dag	Tijdvak	ZoneType	Holiday	Aanwezig	Manual Override	Verwachte aanwezigheid
Dinsdag	09:15	Kantoor	Nee	Ja	Ja	Ja (90%)
Vrijdag	16:30	Vergader	Nee	Nee	Nee	Nee (20%)
Maandag	08:00	Werkvloer	Ja	Nee	Nee	Nee (vakantie)

14.1.6 Adaptiviteit

Het model wordt **voortdurend getraind op nieuwe data**:

- Elke 24 uur worden aanwezigheidspatronen geüpdatet;
- Het model leert unieke gedragspatronen per ruimte (bijv. 'vergaderzaal 3 wordt vooral gebruikt op woensdagmiddag');
- Feedback van gebruikers kan worden gebruikt om voorspellingen aan te scherpen (bijv. door kliks in de app: "ruimte bezet melden").

14.1.7 Toepassing in de praktijk

De voorspelde aanwezigheid stuurt onder andere:

Component	Actie bij verwachte aanwezigheid
Verwarming	Opstarten 30 minuten van tevoren
Koeling	Tijdig activeren bij warm weer
Ventilatie	Aanzetten i.v.m. CO_2-aanvoer
Verlichting	Aan bij begin werkdag

En bij **geen verwachte aanwezigheid**:

- Alles uitschakelen;
- Comfortbandbreedtes verruimen;
- Minder vaak meten/loggen (datareductie).

14.1.8 Privacy en autonomie

- Data wordt **per ruimte versleuteld opgeslagen**;
- Gebruikers zijn anoniem, er wordt alleen met **gedragsclusters** gewerkt;
- Het systeem kan draaien zonder gebruikersfeedback, maar verbetert met opt-in interactie.

14.2 Conclusie

Het gebruikspatroonmodel stelt Cloud Energy Optimizer in staat om:

- Verwarming, koeling en ventilatie **voorspellend en intelligent te sturen**
- Comfort te verhogen zonder verspilling
- De gebruiker centraal te stellen met minimale inspanning
- Privacy te waarborgen én energie te besparen

Het is een cruciale bouwsteen binnen een data gedreven, proactief en schaalbaar smart building systeem.

15 DBN-model 5 – Boilerbelasting en Verbruikspiekmodel

15.1 Doel van het model

Dit **Dynamic Bayesian Network (DBN)** voorspelt of en wanneer een **piekbelasting op de ketel** of warmtebron optreedt. Zo kunnen pieken proactief worden vermeden, energieverbruik verlaagd en wordt congestie op het energiesysteem voorkomen.

Het model geeft input aan load shifting, stooklijnoptimalisatie en buffercapaciteitsturing.

15.1.1 Stap 1: Variabelen

Variabele	Omschrijving	Type	Domein/Eenheid
OutT_oC(t)	Buitentemperatuur	Observabel	continue (°C)
Rad_W_m2(t)	Zoninstraling	Observabel	continue (W/m²)
AirDeT_oC(t)	Gewenste ruimtetemperatuur (comfortdoel)	Observabel	continue (°C)
Temp_oC(t)	Gemeten ruimtetemperatuur	Observabel	continue (°C)
Presence_Bit(t)	Aanwezigheid in zone(s)	Binaire node	{0, 1}
Gas_m3(t)	Actueel gasverbruik per tijdseenheid	Observabel	continue (m³)
BoilerLoad_kW(t)	Vermogensvraag ketel (afgeleid)	Afgeleid	continue (kW)
ExpectedLoad_kW(t)	Voorspelde benodigde ketelbelasting op t+1	Afgeleid	continue (kW)
PeakRisk(t+1)	Kans op verbruikspiek of overbelasting	Doelnode	{laag, midden, hoog}

15.1.1.1 Stap 2: Netwerkstructuur

```
[OutT_oC(t)] ----+
[Rad_W_m2(t)] ---+
[Temp_oC(t)] ----+
[AirDeT_oC(t)] --+
[Presence_Bit(t)]-+---> ExpectedLoad_kW(t+1) ---> PeakRisk(t+1)
[Gas_m3(t)] -------------------------------> BoilerLoad_kW(t)
```

15.1.2 Stap 3: Rekenkundige relaties

15.1.2.1 1. Huidige vermogensvraag

$$BoilerLoad_k W(t) = c \cdot Gas_m 3(t)$$

Met bijv. c=9.5c = 9.5c=9.5 (energie-inhoud aardgas per m³ in kWh) → conversie naar vermogen bij bekend tijdsinterval.

15.1.2.2 2. Voorspelling toekomstige belasting

ExpectedLoad_{kW}(t+1) = \delta_1 \cdot (AirDeT_oC - Temp_oC) + \delta_2 \cdot (1 - \frac{OutT_oC}{20}) + \delta_3 \cdot Presence_Bit - \delta_4 \cdot \frac{Rad_W_m2}{800}

Voorbeeldwaarden (getuned via training):

- δ1=3.5\delta_1 = 3.5δ1=3.5
- δ2=4.0\delta_2 = 4.0δ2=4.0
- δ3=2.0\delta_3 = 2.0δ3=2.0
- δ4=3.0\delta_4 = 3.0δ4=3.0

Let op: formule zorgt dat zonlicht en buitenwarmte pieken dempen, terwijl comfortverschillen en aanwezigheid ze versterken.

15.1.3 Stap 4: Risico-inschatting piekbelasting

ExpectedLoad_kW(t+1)	PeakRisk
< 20 kW	Laag
20–50 kW	Midden
> 50 kW	Hoog

Daarnaast worden afwijkingen tussen huidige en verwachte belasting gelogd om foutdetectie te voeden.

15.1.4 Voorbeeldberekening

Stel:

Variabele	Waarde
AirDeT_oC	21°C
Temp_oC	18.5°C
OutT_oC	5°C
Rad_W_m2	200
Presence_Bit	1

Berekening:

$$ExpectedLoad = 3.5 \cdot (21 - 18.5) + 4.0 \cdot (1 - 5/20) + 2.0 \cdot 1 - 3.0 \cdot (200/800)$$

$$ExpectedLoad = 3.5 \cdot 2.5 + 4.0 \cdot 0.75 + 2.0 - 3.0 \cdot 0.25 = 8.75 + 3.0 + 2.0 - 0.75 = 13.0$$

→ Verwachte ketelbelasting: **13 kW** → **PeakRisk = Laag**

15.1.5 Adaptieve werking

- Modellen worden per gebouw getraind op historische verbruikspieken (gas of stadswarmte);
- Als pieken zich herhalen bij specifieke weeromstandigheden, worden drempelwaarden automatisch aangepast;
- Zones kunnen worden geclusterd: het model houdt rekening met gelijktijdigheid bij meerdere zones.

15.1.6 Praktische toepassingen

Detectie	Actie
Hoog piekrisico	Start ketel/boiler eerder op, verhoog bufferlaadniveau
Midden risico	Schakel dynamisch tussen ketel en WKO/warmtepomp
Laag risico	Sturing op comfort met optimalisatie

Daarnaast:

- Maakt het model deel uit van het **congestiemodel** (portfolio-aansturing);
- Wordt data gedeeld met energiemanagementsystemen (EMS);
- Voorkomt comfortklachten tijdens koude starts.

15.2 Conclusie

Met het Boilerbelastingmodel kan Cloud Energy Optimizer:

Voorspellend sturen op warmteproductie
Piekbelasting voorkomen zonder overcapaciteit
Dynamisch schakelen tussen bronnen
Verbruikskosten verlagen én installatie beschermen

Het model is daarmee onmisbaar in slimme warmtesturing, vooral in gebouwen met gecombineerde opwekking of netbeperkingen.

16 Eindconclusie: Van Technologie naar Samenwerking – De Weg naar Slimme Gebouwen

16.1 1. Inleiding

De ontwikkeling van slimme gebouwen vereist meer dan alleen geavanceerde technologie. Het succes van dergelijke projecten hangt af van de samenwerking tussen verschillende belanghebbenden, de vereenvoudiging van processen en de inzet van adaptieve systemen. De vijf uitgewerkte Dynamic Bayesian Network (DBN) modellen binnen het Cloud Energy Optimizer-platform illustreren deze benadering. Daarnaast onderstreept het artikel van Niels de Jong het belang van samenwerking boven technologie.

16.2 2. Samenvatting van de DBN-modellen

Model 1: Zone Comfort Model

Dit model voorspelt of een ruimte binnen de comfortbandbreedte blijft, rekening houdend met variabelen zoals gemeten en gewenste temperatuur, aanwezigheid, buitentemperatuur en zoninstraling. Het stelt het systeem in staat om proactief te reageren op comfortbehoeften.

Model 2: Warmtepomp Efficiëntiemodel

Dit model bepaalt of de warmtepomp binnen het efficiënte werkgebied opereert, op basis van aanvoer- en retourtemperaturen, buitentemperatuur, elektriciteitsverbruik en modulatieniveau. Het helpt bij het optimaliseren van de warmtepompwerking en het verminderen van energieverbruik.

Model 3: Zoninstralingsinvloedmodel

Dit model schat de temperatuurimpact van zonlicht op een ruimte in, met inachtneming van zoninstraling, zonwering, raamstatus, aanwezigheid en gemeten temperatuur. Het ondersteunt het beheer van comfort en energieverbruik door zonlicht effectief te benutten of te mitigeren.

Model 4: Gebruikspatroonmodel

Dit model voorspelt de aanwezigheid in ruimtes voor voorspellende sturing, gebaseerd op dag van de week, tijdslot, vakantiedagen, ruimtetype en eerdere aanwezigheid. Het maakt het mogelijk om systemen zoals verwarming en verlichting efficiënt te beheren.

Model 5: Boilerbelasting & Verbruikspiekmodel

Dit model voorspelt piekbelasting op ketels om congestie te vermijden, door analyse van buitentemperatuur, gemeten en gewenste temperatuur, aanwezigheid, zoninstraling en gasverbruik. Het draagt bij aan het voorkomen van overbelasting en het optimaliseren van energiegebruik.

16.3 3. Belang van Samenwerking

Het artikel van Niels de Jong benadrukt dat samenwerking de sleutel is tot succesvolle implementatie van slimme gebouwtechnologieën. Technologie alleen is onvoldoende; het vereist een gezamenlijke inspanning van alle betrokken partijen, waaronder gebouwbeheerders, technici, gebruikers en beleidsmakers. Door samen te werken kunnen systemen beter worden afgestemd op de behoeften van gebruikers en kunnen implementatie-uitdagingen effectiever worden aangepakt.

16.4 4. Vereenvoudiging en Adaptiviteit

De implementatie van de DBN-modellen toont aan dat vereenvoudiging van processen en adaptieve technologie cruciaal zijn voor schaalbaarheid en effectiviteit. Door het gebruik van een wizard die op basis van eenvoudige vragenlijsten koppelpagina's genereert, wordt de foutmarge geminimaliseerd en kan een gebouw binnen één dag worden aangesloten. Bovendien is de implementatietijd gereduceerd van gemiddeld 240 uur naar 16 uur, met een significante verlaging van de foutmarge.

16.5 5. Conclusie

De transitie naar slimme, duurzame gebouwen is niet louter een technologische uitdaging, maar vooral een organisatorische en samenwerkingsgerichte opgave. De combinatie van adaptieve DBN-modellen, vereenvoudigde implementatieprocessen en een focus op samenwerking vormt de basis voor succesvolle en schaalbare oplossingen. De echte kracht ligt in de samenwerking tussen mensen.

Door deze benadering kunnen gebouwen niet alleen energie-efficiënter worden beheerd, maar ook beter inspelen op de behoeften van gebruikers, wat leidt tot verhoogd comfort en tevredenheid. De toekomst van slimme gebouwen ligt in de synergie tussen technologie en samenwerking.

17 Slot conclusie

17.1 Succes zit in Samenwerking, Niet in Technologie Alleen

Het project rond de *Cloud Energy Optimizer (CEO)* binnen het *Brains for Buildings (B4B)* programma laat zien dat technologische innovaties zoals foutdetectie, voorspellende sturing en adaptieve modellen veel potentieel hebben voor energiebesparing en comfortoptimalisatie in gebouwen. Toch blijkt dat de sleutel tot succes **niet in technologie alleen ligt**, maar vooral in de **samenwerking tussen mensen**: gebouwbeheerders, technici, gebruikers, leveranciers en beleidsmakers.

17.2 Techniek is er – maar de data ontbreekt vaak

- Het 4S3F-diagnosemodel en andere geavanceerde algoritmes werken goed in laboratoria, maar falen vaak in de praktijk door ontbrekende of inconsistente data.
- Belangrijke randvoorwaarden, zoals volledige en gestandaardiseerde gebouwdata, zijn zelden aanwezig.
- Gebruikersdata (aanwezigheid, gedrag, voorkeuren) zijn veelbelovend, maar moeilijk te verzamelen, valideren en interpreteren.

17.3 Stapsgewijze benadering als succesformule

- CEO adviseert een gefaseerde aanpak: **beginnen bij datarijke gebouwen**, eerst eenvoudiger diagnosemodellen inzetten en daarna uitbreiden.
- Door te focussen op 20% van de gebouwen die 80% van de benodigde data al hebben, ontstaan realistische toepassingen.
- Er wordt gewerkt aan “light-versies” van algoritmes en DBN-modellen die minder afhankelijk zijn van complexe input.

17.4 Voorbeeld van een succesvol project:

De use-cases bij **Vrije Universiteit Brussel** en **Besi Duiven** tonen hoe foutdetectie, gebaseerd op afwijkingen tussen voorspeld en gemeten gedrag, concrete besparingen én comfortverbeteringen oplevert:

- Bij VUB werd een slecht functionerende warmtepomp ontdekt, die zonder deze analyse onopgemerkt zou blijven.
- Bij Besi werden structurele regelproblemen blootgelegd, zoals het gelijktijdig verwarmen en koelen, zonder dat gebruikers dit doorhadden.

17.5 Samenwerking maakt het verschil

- In deze projecten bleek samenwerking essentieel: tussen softwareontwikkelaars, installatiebedrijven en gebouwbeheerders.
- Alleen door open communicatie, gezamenlijke doelen en feedbackloops konden de algoritmes effectief worden toegepast.
- CEO toont aan dat het betrekken van eindgebruikers via dashboards en visualisaties het draagvlak en het effect van energiesturing versterkt.
- **Conclusie**

De echte innovatie zit niet in de technologie, maar in **hoe mensen samenwerken** om die technologie slim toe te passen. Zonder samenwerking blijft slimme techniek dom.

De Cloud Energy Optimizer is technisch klaar voor de toekomst, maar **de weg naar slimme gebouwen loopt via mensen, niet via machines**. Door te kiezen voor een stapsgewijze, samenwerkingsgerichte aanpak ontstaat een schaalbare, robuuste en gedragen oplossing voor energieoptimalisatie in de praktijk.