



CHALMERS



Optimization Home Energy Management System

Examensarbete inom högskoleprogrammet datateknik

HUSSEIN AL-SAADI
ÖMER GÜNDOĞDU

INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK

CHALMERS TEKNISKA HÖGSKOLA

Göteborg 2025

www.chalmers.se

EXAMENSARBETE 2025

Optimization Home Energy Management System

Utveckling av en simulator för visualisering och kostnadsoptimering
av hushållens elanvändning

Hussein Al-Saadi

Ömer Gündogdu



CHALMERS

Institutionen för data- och informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2025

Optimization Home Energy Management System
Utveckling av en simulator för visualisering och kostnadsoptimering av hushållens
elanvändning
Hussein Al-Saadi & Ömer Gündogdu

© Hussein Al-Saadi & Ömer Gündogdu 2025.

Handledare: Xiuqi Zhang, Data- och informationsteknik
Handledare: Daniel Elfstrand, Grundare & VD för Grid Collective
Examinator: Nicholas Smallbone, Data- och informationsteknik

Examensarbete 2025
Institutionen för Data- och informationsteknik
Chalmers Tekniska Högskola
SE-412 96 Göteborg
Telefon +46 31 772 1000

Omslagsbild: Ett hem med solceller och batteri samt en simulator i himlen, genererad
med OpenAI DALL-E

Skriven i L^AT_EX
Göteborg 2025

Optimization Home Energy Management System

Utveckling av en simulator för visualisering och kostnadsoptimering av hushållens elanvändning

Hussein Al-Saadi & Ömer Gündogdu

Institutionen för Data- och informationsteknik

Chalmers Tekniska Högskola

Sammanfattning

Detta examensarbete har resulterat i en fullt fungerande simulator och optimeringsalgoritm som hjälper svenska hushåll att minska sina elkostnader genom att analysera energiförbrukning och ge konkreta rekommendationer för att jämna ut effekttoppar. Bakgrunden till projektet är det ökande antalet solpaneler, hembatterier och elbilar samt införandet av effektavgifter och dynamiska elpriser, vilket ställer nya krav på hushållens energihantering.

Projektet genomfördes i samarbete med Grid Collective och resulterade i ett användarvänligt verktyg där konsumenter kan importera sin energidata i CSV-format och få visualiseringar och rekommendationer för kostnadsoptimering. Simulatorens utveckling gjordes i React med TypeScript och använder PapaParse och Recharts för datainläsning och visualisering. Optimeringsalgoritmen bygger på analys av historisk data och fokuserar på att identifiera de tre högsta effekttopparna och omfördela dessa för att minska kostnader.

Effektiviteten utvärderades genom att jämföra energikostnader och effektvariation före och efter optimering. Resultaten visade att algoritmen kunde reducera toppar och ge användaren tydliga rekommendationer för hur energianvändningen kan optimeras i framtiden. Arbetet utgör ett underlag för vidare utveckling och potentiell implementering hos Grid Collective.

Abstract

This thesis resulted in the development of a working simulator and optimization algorithm designed to help Swedish households reduce their electricity costs by analyzing energy usage and identifying peak load periods.

In collaboration with Grid Collective, the simulator was implemented in React and allows users to upload CSV files to visualize consumption patterns and receive optimization suggestions. The optimization algorithm targets the three highest power peaks and redistributes consumption to reduce associated costs. Technologies like PapaParse and Recharts were used to handle data processing and visualization in the frontend.

The system was tested using real consumption data and evaluated by comparing energy costs and power peaks before and after optimization. The results demonstrated that the algorithm effectively reduces peaks and provides tangible recommendations for cost savings. The project provides a foundation for further development and possible deployment by Grid Collective.

Nyckelord: grön, hållbar, simulator, react, typescript, energi, el, effektagift, avgifter, elbolag

Förord

Detta arbete är resultatet av flera veckors engagemang, reflektion och lärande. Under arbetets gång har vi fått möjlighet att fördjupa oss i ämnet, utveckla våra analytiska färdigheter och förfinas i vår förmåga att strukturera och presentera information på ett tydligt sätt.

Vi vill rikta ett stort tack till Xiuqi Zhang och Daniel Elfstrand för värdefull vägledning och stöd genom hela processen. Förhoppningsvis kan detta arbete bidra till vidare förståelse och väcka intresse för ämnet hos läsaren.

Hussein Al-Saadi & Ömer Gündogdu , Göteborg, Maj 2025

Akronymer

CSV	Comma Separated Value
kW	Kilowatt
DALL-E	Image generation model by OpenAI
HTML	HyperText Markup Language
JSX	JavaScript XML
CSS	Cascading Style Sheets
API	Application Programming Interface
UML	Unified Modeling Language
CPU	Central Processing Unit
UI	User Interface

Nomenklatur

Index

d	Dagsindex i Daglig ombalansering
i	Tidsstegsindex
j	Granneindex i glidande fönster eller daglig ombalansering
k	Index för den k :te största toppen

Mängder

D	Mängd av kalenderdygn
G_d	Mängd av förbrukningsvärden dag d
W_i	Grannmängd kring tidssteg i

Parametrar

C_{\max}	Maximal kapacitet per period
M	Antal mätpunkter i tidsserien
N	Antal toppvärden som ska dämpas
α	Andel som omfördelas, $0 < \alpha < 1$
β	Tröskelfaktor för peakdetektion, $\beta > 1$
w	Fönsterstorlek

Variabler

c_i	Kvarvarande kapacitet vid i
δ	Överförd mängd energi i ett omfördelningssteg
Δ	Överföringskvantitet i Daglig ombalansering
p_k	Index för den k :te största toppen
Q_d	Dygnsmedelvärde vid dag d
r	Totalt omfördelad volym, gäller endast för Peak Shaving

r_k	Flyttbar volym från topp p_k
t_i	Tidsstämpel vid tidssteg i
u_i	Uppmätt förbrukning vid tidssteg i
u'_i	Justerat värde vid tidssteg i efter optimering
\bar{u}_i	Lokalt medelvärde vid i
u_{p_k}	Värdet av den k :te största toppen p_k
\tilde{u}_i	Utjämnat värde vid tidssteg i i Smoothing-sektionen

Innehåll

Akronymer	x
Nomenklatur	xiii
Figurer	1
1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	2
1.3 Grid Collective	2
1.4 Mål	2
1.5 Avgränsningar	3
1.6 Källhänvisning	3
2 Teknisk bakgrund	5
2.1 Teknisk stack	5
2.1.1 React	5
2.1.2 Vite	5
2.1.3 TypeScript	5
2.1.4 JavaScript	6
2.1.5 Tailwind CSS	6
2.2 Tredjepartsbibliotek	6
2.2.1 PapaParse	6
2.2.2 Recharts	6
2.3 Optimeringsalgoritm	6
2.3.1 Topp-N optimering	7
2.3.2 Peak Shaving	8
2.3.3 Daglig ombalansering	9
2.3.4 Smoothing	10
3 Metod	11
3.1 Övergripande arbetsgång	11
3.2 Utveckling av optimeringsalgoritm	12
3.2.1 Identifiering av kandidatmetoder	12
3.2.2 Prototypimplementering	12
3.2.3 Utvärdering	12
3.2.4 Val av slutlig metod	12

3.2.5	Implementering	12
3.3	Utveckling av simulator	12
3.4	Testning och validering	13
3.4.1	Felhantering & användarfeedback	13
3.5	Etiska och tekniska överväganden	13
4	Genomförande	15
4.1	Systemarkitektur och teknisk plattform	15
4.2	Datamodell och databehandling	15
4.3	Algoritmutveckling	16
4.4	Simulatorns utveckling	16
4.5	Användargränssnitt	17
4.6	Implementation och systemtestning	18
4.7	Utmaningar och anpassningar under utvecklingen	18
5	Resultat	19
5.1	Utseende och användargränssnitt	19
5.1.1	Startsidan	19
5.1.2	Resultatssidan	19
5.2	Algoritmer och optimeringsresultat	22
5.3	Förbättringar och utvecklingspotential	22
5.3.1	Genomförda förbättringar	22
5.3.2	Utvecklingspotential	22
6	Diskussion och slutsats	25
6.1	Sammanfattning av resultat	25
6.2	Uppfyllelse av mål och frågeställningar	25
6.3	Begränsningar och felkällor	25
6.4	Praktiska och teoretiska implikationer	26
6.5	Förslag på framtida arbete	26
6.6	Reflektion kring teknikval	26
6.6.1	React och TypeScript	26
6.6.2	Vite	26
6.6.3	Tailwind CSS	27
6.6.4	Tredjepartsbibliotek	27
6.6.5	Hållbar utveckling	27
	Litteraturförteckning	29

Figurer

3.1	GANTT schema som beskriver de tre faserna	11
4.1	UML Diagram	17
5.1	Hemsida	20
5.2	Resultatsida	21

1

Inledning

Med den ökande installationen av solpaneler, hembatterier och elbilar i svenska hushåll har behovet av effektiva energihanteringssystem blivit allt viktigare. Effektagifter och timvisa variationer i elpriser har skapat en utmaning för hushåll som vill minimera sina kostnader. Syftet med detta examensarbete är att utveckla en simulator och en algoritm som hjälper hushåll med att visualisera användningen och ge rekommendationer för att sänka elkostnaden.

1.1 Bakgrund

Energisystemet befinner sig i en omfattande och snabb förändring, driven av teknologiska framsteg och den globala omställningen till hållbara energilösningar [1]. En viktig faktor bakom denna utveckling är den snabba ökningen av distribuerade energikällor som solceller, energilagring i hembatterier och användning av elektriska fordon [2]. Dessa teknologier ger hushållen möjlighet att kontrollera sin energiförbrukning mer effektivt, vilket främjar ökad egenproduktion och minskat beroende av traditionella energileverantörer [3].

Samtidigt medför spridningen av dessa teknologier nya utmaningar för elnäten, såsom ökad intermittens, förändrade belastningsmönster och större variationer i efterfrågan på el [4]. Införandet av dynamiska elpriser, som varierar baserat på efterfrågan och tillgång på energi, har förstärkt behovet av intelligenta energihanteringssystem. Dessa system analyserar och optimerar hushållens elanvändning för att utnyttja lågprisperioder och minimera effekttoppar som annars kan leda till höga kostnader [5].

Genom att tillämpa avancerade algoritmer och simuleringar är det möjligt för hushåll att bättre förstå sin energianvändning, planera konsumtionen strategiskt och därmed sänka sina energikostnader [6]. Grid Collective arbetar med att utveckla innovativa lösningar inom smart energihantering och har identifierat behovet av en simulator som kan ge hushåll insikter i deras energiförbrukning. En sådan simulator hjälper användare att fatta välgrundade beslut kring sin energianvändning, vilket både leder till ekonomiska besparingar och bidrar till ett mer hållbart energisystem.

1.2 Syfte

Syftet med detta examensarbete är att utveckla ett användarvänligt simuleringsverktyg och en optimeringsalgoritm som hjälper svenska hushåll att analysera och planera sin energiförbrukning i syfte att minska elkostnader, undvika effektavgifter och bidra till en mer hållbar elanvändning. Genom att visualisera förbrukningsdata och ge konkreta rekommendationer för hur effektuttag kan jämnas ut, ska lösningen göra det möjligt för användare att fatta informerade beslut om sin energianvändning.

1.3 Grid Collective

Grid Collective är ett svenskt företag med säte i Göteborg som verkar för att påskynda energiomställningen genom innovativa lösningar inom energihantering. Företagets övertygelse är att omställningen till förnybar energi inte bara handlar om att möta dagens behov, utan om att bygga en framtid där hållbara energilösningar är tillgängliga för alla.

Genom nära samarbete med företag och samhällen utvecklar Grid Collective tekniska lösningar som syftar till att optimera energiförbrukning och minska klimatavtryck. Företaget betonar vikten av hållbarhet, innovation och samarbete som kärnvärden i sitt arbete [7].

1.4 Mål

Projektets mål är att:

- Utveckla en användarvänlig simulator som riktar sig till svenska hushåll.
- Utveckla en optimeringsalgoritm som hjälper användare att minska sina elkostnader genom att analysera historisk energiförbrukning över en månad.
- Tillhandahålla konkreta rekommendationer för att jämna ut effekttoppar i energianvändningen.
- Visualisera energianvändningen på ett intuitivt och lättförståeligt sätt.
- Ge rekommendationer på hur användaren kan minska sin effektavgift baserat på historiska energiförbrukningsdata.
- Leverera en fullt fungerande prototyp av simulatoren och algoritmen till företaget Grid Collective för vidare användning och eventuell vidareutveckling.

1.5 Avgränsningar

Projektet är begränsad enligt följande punkter:

- Simulatorens behandlar enbart energianvändning i hushåll. Industriella och kommersiella byggnader ingår inte i tillämpningsområdet.
- Simulatorens är anpassad för svenska elprismodeller och effektavgifter.
- Ingen realtidsoptimering genomförs. Algoritmen baseras uteslutande på historiska konsumtionsdata som användaren manuellt tillhandahåller.
- Optimeringen fokuserar på att identifiera och reducera de tre högsta effekttopparna. Övriga aspekter av energianvändningen beaktas inte.
- De rekommendationer som genereras är begränsade till den data som tillhandahållits av användaren. Slutsatser och råd baseras därmed på det specifika datamaterialet och kan inte generaliseras utanför detta.
- Simulatorens riktar sig till hushåll som omfattas av effektavgifter. För hushåll utan sådan tariffstruktur är verktygets praktiska nytta begränsad.
- Den initiala implementationen stödjer enbart tariffstrukturen hos elnätsföretaget Ellevio. Stöd för andra elbolags modeller kan tillkomma i framtida versioner, men omfattas ej av detta projekts leverans.
- Optimering och analys sker på tim- och dagsnivå inom en månad. Varken kortare tidsupplösning (exempelvis minutnivå) eller längre tidsperioder behandlas.
- Endast data i form av tidsupplöst förbrukningsdata med en timmes upplösning, det vill säga ett värde per timme under en hel månad accepteras. Andra format eller tidsupplösningar hanteras inte.
- Simulatorens är utformad som ett klientbaserat verktyg utan separat backendlösning. All databehandling sker lokalt i webbläsaren. Serverlösningar, databaser eller molntjänster ingår inte i projektets omfattning.

1.6 Källhänvisning

För att säkerställa källornas äkthet och trovärdighet har urvalet genomförts med fokus på deras aktualitet, relevans, avsändare, noggrannhet och syfte. Källorna har valts ut i relation till projektets mål, med särskild vikt vid att använda primärkällor där det varit möjligt, såsom officiell dokumentation och publikationer från etablerade aktörer. Genom detta tillvägagångssätt har arbetet strävat efter att bygga på pålitlig och saklig information.

2

Teknisk bakgrund

För att utveckla simulatorn användes React tillsammans med Vite, TypeScript och Tailwind CSS.

2.1 Teknisk stack

Projektet bygger på moderna webbutvecklingsteknologier med målet att uppnå hög prestanda, en responsiv design samt en god utvecklarpoplevelse, vilket innebär att det ska vara enkelt för företaget att förstå, underhålla och vidareutveckla simulatorn. React användes som ramverk för att möjliggöra en komponentbaserad struktur i gränssnittet [8]. Vite valdes som byggverktyg och utvecklingsserver på grund av dess korta uppstartstid och effektiva hot-reloading, vilket underlättade arbetet under utvecklingsfasen [9]. Koden skrevs i TypeScript för att möjliggöra statisk typkontroll och därmed öka kodens robusthet och läsbarhet [12]. Tailwind CSS användes för att implementera ett konsekvent och responsivt gränssnitt med hjälp av utility-klasser direkt i komponenterna [14].

2.1.1 React

React är ett JavaScript-bibliotek som används för att skapa användargränssnitt med hjälp av komponentbaserad arkitektur [8]. Arkitekturen gör det möjligt att dynamiskt uppdatera innehåll i användargränssnittet baserat på applikationens tillstånd, vilket var centralt för att implementera en interaktiv simulator.

2.1.2 Vite

Vite [9] användes som bundler och utvecklingsserver. Jämfört med traditionella verktyg som Webpack [10] erbjuder Vite en snabbare utvecklingscykel och bättre hantering av moderna JavaScript-projekt [11]. Det bidrog till en mer effektiv utvecklingsprocess.

2.1.3 TypeScript

TypeScript är ett typat superset av JavaScript som transpileras till vanlig JavaScript [12]. Genom att införa statisk typning förbättrades kodens struktur, underhållbarhet

och läsbarhet. Den underlättar även samarbete mellan utvecklare och minskade risken för typrelaterade fel.

2.1.4 JavaScript

JavaScript [13] användes enbart i prototypfasen för att snabbt utvärdera algoriternas prestanda. Resultaten låg till grund för beslutet att porta den valda algoritmen till TypeScript.

2.1.5 Tailwind CSS

Tailwind CSS användes för att snabbt och effektivt kunna bygga ett stilrent och responsivt användargränssnitt [14]. Genom att använda utility-klasser direkt i HTML/JSX kunde man undvika extern CSS och istället hålla styling nära komponenterna. Detta underlättade prototypning och bidrog till en enhetlig design genom hela projektet.

2.2 Tredjepartsbibliotek

För att effektivisera utvecklingen integrerades flera tredjepartsbibliotek. Dessa bidrog till ökad kodkvalitet, bättre användarupplevelse och minskad utvecklingstid. Bland de mest centrala biblioteken märks PapaParse och Recharts [15, 16].

2.2.1 PapaParse

PapaParse är ett bibliotek för hantering av CSV-filer i JavaScript [15]. Det används för att importera användarens energidata från CSV. Biblioteket stödjer asynkron parsning, automatisk typdetektion och robust hantering av fel, vilket var användbart vid validering av data i projektet.

2.2.2 Recharts

Recharts är ett React-baserat bibliotek för datavisualisering [16]. I projektet användes det för att skapa linjediagram som visar elanvändning före och efter optimering. Diagrammen har stöd för verktygstips (tooltips), färgkodning samt markering av toppar i förbrukningen. Denna typ av visualisering underlättar analys av energimönster och optimeringsresultat.

2.3 Optimeringsalgoritm

Indata för följande optimeringsalgoritmer är en tidsserie $\{(t_i, u_i)\}_{i=1}^M$ där t_i är tidsstämplar ordnade i stigande följd och u_i är uppmätt förbrukning.

2.3.1 Topp-N optimering

Algoritmen för Topp-N optimering är en heuristisk metod för att jämna ut de högsta belastningstopparna i en tidsserie av energiförbrukning genom att omfördela en del av lasten till perioder med lägre efterfrågan.

Parameter och mål

Parametern N anger antalet toppvärden som ska dämpas. Målet är att reducera N största topparna, det vill säga minimera $\max_i u'_i$, under bibehållet total energiförbrukning.

Identifiering av toppar

Utifrån den ursprungliga serien väljs de N högsta värdena $\{u_p, \dots, u_{pN}\}$.

Beräkning av flyttvolym

För varje vald topp p_k beräknas en flyttbar volym $r_k = \alpha \cdot u_{p_k}$, där $\alpha = 0.5$ och valdes genom iterativa försök. Ett högre värde på α ledde till en orealistiskt aggressiv optimering, medan ett lägre värde inte skulle ge någon betydande förbättring.

Val av mottagare och kapacitetsbegränsning

Alla tidpunkter ordnas i stigande ordning efter sin förbrukning, så att de lägsta u_i prioriteras vid omfördelning. Varje period i kan maximalt ta emot

$$c_i = \max(C_{max} - u_i, 0) \quad (2.1)$$

enheter, där C_{max} är ett fördefinierat kapacitetstak.

Omfördelningsprocess

För varje topp p_k itereras mottagarlistan i stigande ordning. I varje steg överförs

$$\delta = \min(c_i, r_k) \quad (2.2)$$

från p_k till i . Efter överföringen uppdateras

$$u_i \leftarrow u_i + \delta, \quad u_{p_k} \leftarrow u_{p_k} - \delta, \quad c_i \leftarrow c_i - \delta, \quad r_k \leftarrow r_k - \delta \quad (2.3)$$

Resultat

Den resulterande serien $\{(t_i, u'_i)\}_{i=1}^M$ innehåller de utjämnade förbrukningsvärdena.

Begränsningar

Metoden är girig och garanterar inte global optimalitet; parametrarna α och C_{max} är fasta och inga övergångskostnader eller dynamiska begränsningar beaktas. Den fungerar dock snabbt och enkelt för praktisk belastningsutjämning i energisystem.

2.3.2 Peak Shaving

Peak Shaving-algoritmen [17] är en metod för att dämpa lokala förbrukningstoppar i en tidsserie genom att omfördela överskottsförbrukning till angränsande perioder med lägre belastning.

Parametrar och Mål

Parametrar är:

$$w \in \mathbb{N}, \quad \beta > 1, \quad \alpha \in (0, 1). \quad (2.4)$$

Målet är att för varje i där $u_i > \beta \bar{u}_i$ dämpa ökningen så att inga stora lokala toppar kvarstår.

Beräkning av lokalt medelvärde

För varje $i = \lfloor w/2 \rfloor + 1, \dots, M - \lfloor w/2 \rfloor$ beräknas

$$\bar{u}_i = \frac{1}{w-1} \sum_{\substack{j=i-\lfloor w/2 \rfloor \\ j \neq i}}^{i+\lfloor w/2 \rfloor} u_j. \quad (2.5)$$

Toppdetektion och reduktion

Om

$$u_i > \beta \bar{u}_i \quad (2.6)$$

beräknas överskottet

$$\Delta = u_i - \bar{u}_i, \quad r = \alpha \Delta \quad (2.7)$$

och

$$u_i \leftarrow u_i - r. \quad (2.8)$$

Val av mottagare

Mottagarindex utgörs av

$$W_i = \{j : j \neq i, i - \lfloor w/2 \rfloor \leq j \leq i + \lfloor w/2 \rfloor\}. \quad (2.9)$$

Eventuellt kan W_i begränsas till tider med låg efterfrågan.

Omfördelningsprocess

För varje $j \in W_i$ fördelas r antingen:

$$u_j \leftarrow u_j + \frac{r}{|W_i|} \quad (2.10)$$

eller, om kapacitetstak C_{\max} finns,

$$c_j = \max(C_{\max} - u_j, 0) \quad (2.11)$$

$$\delta_j = \min\left(c_j, \frac{r}{|W_i|}\right) \quad (2.12)$$

$$u_j \leftarrow u_j + \delta_j \quad (2.13)$$

$$r \leftarrow r - \sum_{j \in W_i} \delta_j \quad (2.14)$$

Resultat

Den justerade serien $\{(t_i, u'_i)\}_{i=1}^M$ returneras med dämpade lokala toppar.

Begränsningar

Metoden är girig och garanterar inte global optimalitet. Valen av w , β och α kräver fintrimning. Vid fasta C_{\max} kan kapacitetsbegränsningar göra att överskott återstår oredistribuerat.

2.3.3 Daglig ombalansering

Daglig ombalansering [18] är en heuristisk metod, och denna algoritm jämnar ut förbrukningen inom varje kalenderdygn genom att omfördela energi så att varje timvärde närmar sig dygnets medelvärde utan att överskrida ett kapacitetstak.

Mål

Målet är att för varje dag d minimera avvikelserna från dygnsmedelvärdet Q_d under bibehållen total energimängd.

Aggregering per dygn

Tidsseriens poster grupperas per kalenderdygn:

$$G_d = \{u_i : t_i \text{ tillhör dag } d\}, \quad d \in D. \quad (2.15)$$

Beräkning av dygnsmedelvärde

För varje dag d beräknas

$$Q_d = \frac{1}{|G_d|} \sum_{u \in G_d} u. \quad (2.16)$$

Iterativ omfördelning

Inom varje dag d upprepas följande tills inga ytterligare förbättringar kan göras:

$$\text{För alla par } (i, j) \text{ med } u_i > Q_d, u_j < Q_d : \quad (2.17)$$

$$\Delta = \min(u_i - Q_d, Q_d - u_j, C_{\max} - u_j), \quad (2.18)$$

$$u_i \leftarrow u_i - \Delta, \quad u_j \leftarrow u_j + \Delta. \quad (2.19)$$

Resultat

Den utjämnade serien $\{(t_i, u'_i)\}_{i=1}^M$ återlämnas där u'_i är värdet efter omfördelning.

Begränsningar

- Heuristisk metod utan garanti för global optimalitet.
- Iterativ konvergens kan vara långsam vid stora avvikelser eller höga C_{\max} .
- Parametern C_{\max} måste justeras efter tekniska och affärsmässiga krav.

2.3.4 Smoothing

Algoritmen för smoothing reducerar brus och högfrekventa variationer i en tidsserie genom glidande eller exponentiell medelvärdesberäkning.

Mål

Målet är att producera en utjämnad serie $\{(t_i, \tilde{u}_i)\}_{i=1}^M$.

Glidande medelvärde [19]

För ett fönster w beräknas för varje index

$$\tilde{u}_i = \frac{1}{w} \sum_{j=i-\lfloor w/2 \rfloor}^{i+\lfloor w/2 \rfloor} u_j, \quad \lfloor w/2 \rfloor < i \leq M - \lfloor w/2 \rfloor. \quad (2.20)$$

Vid kanter kan fönstret justeras eller användas symmetrisk förlängning.

Utjämning

Initieras med $\tilde{u}_1 = u_1$. För $i = 2, \dots, M$ gäller

$$\tilde{u}_i = \alpha u_i + (1 - \alpha) \tilde{u}_{i-1}, \quad \alpha \in (0, 1).$$

Resultat

Den utjämnade serien $\{(t_i, \tilde{u}_i)\}$ returneras, där högfrekvent brus är dämpat och underliggande trender tydligare.

Begränsningar

- Glidande medelvärde kan förskjuta signalen och sudda ut skarpa förändringar.
- Exponentiell utjämning introducerar fördröjning och kräver val av α .
- Metoderna beaktar inga kapacitets- eller regelbaserade begränsningar.

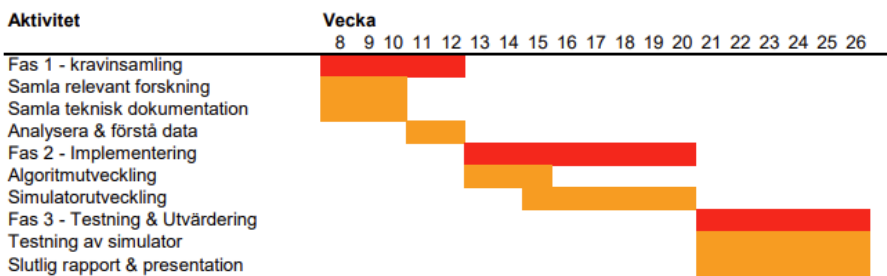
3

Metod

Detta kapitel beskriver det metodologiska tillvägagångssätt som kommer att användas för att utveckla simulatoren och optimeringsalgoritmen. Arbetet kommer att följa en iterativ utvecklingsprocess. Projektet kommer att bestå av tre faser (Se Fig 3.1): kravinsamling, implementering och testning & utvärdering.

GANTT-schema

Optimization Home Energy
Management System



Figur 3.1: GANTT schema som beskriver de tre faserna

3.1 Övergripande arbetsgång

Arbetet kommer att inledas med en genomgång av relevant forskning och teknisk dokumentation för att skapa en teoretisk grund för utvecklingen av simulatoren. Parallellt kommer en behovsanalys att genomföras i samråd med uppdragsgivaren Grid Collective, denna behovsanalys resulterade i ett antal centrala krav för systemet: visuell feedback samt funktionalitet för att identifiera och minimera effekttoppar i energiförbrukningen.

Utifrån dessa krav delades projektet in i två huvudsakliga utvecklingsspår:

- Utveckling av en optimeringsalgoritm för energiförbrukning.
- Implementation av en webbaserad simulator för visualisering och analys.

3.2 Utveckling av optimeringsalgoritm

Algoritmens mål är att sänka effektavgiften genom att jämna ut de tre högsta timtopparna i varje månadsprofil. Följande arbetssteg kommer att genomföras för att uppnå kriterierna:

3.2.1 Identifiering av kandidatmetoder

Genom en litteraturstudie identifierades fyra optimeringsmetoder - Peak Shaving, daglig ombalansering, smoothing och Topp-N, dessa bedömdes vara bäst lämpade för implementationen.

3.2.2 Prototypimplementering

Prototypen kommer implementeras så att varje utvald metod skrivs som en fristående JavaScript-funktion, och dess effektivitet som optimeringsalgoritm testas direkt i webbläsaren.

3.2.3 Utvärdering

Prototyperna kommer att köras mot ett urval verkliga månadsprofiler. Två mätvärden kommer att dokumenteras:

1. Optimeringsprestanda - procentuell reduktion av de tre högsta timtopparna och den därav beräknade sänkningen av effektavgiften.
2. Tålighet för mänskliga avvikelser - hur bra resultatet stod sig när de antagits följas av en användare som inte lyckas genomföra lastförskjutningen perfekt (t.ex. fördröjningar, uteblivna manuella åtgärder).

3.2.4 Val av slutlig metod

Topp-N med parametern $N = 3$ uppvisade kortast körtid och stabila resultat även vid avvikelser i indata. Metoden valdes därför för den färdiga simulatorm.

3.2.5 Implementering

Den slutliga algoritmen itererade över den uppladdade månadsprofilen, markerade de tre högsta timvärdena och flyttade en andel $\alpha = 0.5$ från varje topp till timmar med låg belastning, samtidigt som totalsumman hölls konstant. All logik skrevs i TypeScript och exekverades helt på klientsidan.

3.3 Utveckling av simulator

Simulatorm kommer att byggas helt i frontend med React och TypeScript. Biblioteket PapaParse kommer att användas för att läsa CSV-filer lokalt i webbläsaren; backendlösningar uteslöts för att förbättra integritet och minska komplexitet.

Gränssnittet kommer att utvecklas med fokus på enkelhet och responsivitet. Användaren kommer att ladda upp sin data, köra optimeringen och få direkt visuell återkoppling via Recharts-diagram. Visualiseringen innefattar förbrukningsprofil före och efter optimering, samt indikation av identifierade toppar.

3.4 Testning och validering

Simulatorn kommer att testas kontinuerligt med verklig energidata från en individ i Grid Collective. Testningen kommer att fokusera på:

- att algoritmen ger teoretiskt baserade rekommendationer som visar hur energianvändningen kunde ha optimerats om de hade följts, och som därmed illustrerar möjliga praktiska besparingar,
- att systemet kan hantera ogiltig eller ofullständig data korrekt,
- att användargränssnittet kommer vara responsiv och begriplig.

3.4.1 Felhantering & användarfeedback

Felhantering och användarfeedback kommer integreras kontinuerligt under utvecklingen för att öka systemets robusthet och användbarhet. Eftersom systemet kommer att utvecklas helt i frontend kommer all testning genomföras direkt i webbläsaren och på olika enheter och skärmstorlekar. Detta arbetssätt gör det möjligt att iterera snabbt, upptäcka fel tidigt och bekräfta funktionaliteten utan någon extern serveranslutning.

3.5 Etiska och tekniska överväganden

För att skydda användarnas integritet valdes en modell där all databehandling sker lokalt; ingen information lagras eller skickas externt. Systemet är byggt för att vara transparent, tillgängligt och resurseffektivt. Kodbasen hölls modulär och väldokumenterad för att stödja framtida vidareutveckling och eventuell integration med andra system.

4

Genomförande

Detta kapitel beskriver genomförandet av examensarbetet, med fokus på systemets huvudsakliga komponenter, valda tekniska lösningar samt det praktiska arbetet bakom utvecklingen av simulatoren och optimeringsalgoritmen.

4.1 Systemarkitektur och teknisk plattform

Systemet implementerades helt i webbläsaren med hjälp av React och TypeScript, vilket eliminerade behovet av en traditionell backend. Denna design gör simulatoren lätt att distribuera som en statisk webbapplikation, och användardata behandlades lokalt i webbläsaren, vilket också förbättrar integriteten.

Optimeringsalgoritmen skrevs i TypeScript och exekverades direkt i frontend. Algoritmen analyserade energiförbrukningsdata som användaren laddade upp i CSV-format. Ingen information sparades eller skickades till en server; all bearbetning skedde alltså på klientsidan i realtid.

Visualiseringen av data utfördes med biblioteket Recharts, som visade före- och efterprofiler, effekttoppar samt förändringar i kostnad och förbrukning. Arkitekturen byggdes komponentbaserad, vilket förenklade både utveckling och vidareutveckling.

4.2 Datamodell och databehandling

All data som hanterades i systemet importerades av användaren som en CSV-fil med en timupplöst förbrukningsprofil. Filen tolkades i webbläsaren med PapaParse. CSV-filen förväntades innehålla två kolumner: tidsstämpel och energiförbrukning (kW).

Vid inläsningen utfördes följande valideringssteg:

- Kommentarer (rader som började med #) och rubrikrader (t.ex. Date) filtrerades bort.
- Rader som inte innehöll minst två kolumner ignorerades.
- Förbrukningsvärden konverterades till flyttal, varvid kommatecken ersattes med punkt.

- Felaktiga eller ogiltiga värden utlöste ett tydligt felmeddelande till användaren.

När datan klarade samtliga kontroller normaliserades den och lagrades tillfälligt i minnet för vidare visualisering. Ingen backend eller permanent lagring användes. Hela processen skedde alltså lokalt i webbläsaren.

Dataflödet kunde sammanfattas som

Uppladdning → Parsning → Validering → Visualisering.

4.3 Algoritmveckling

Under algoritmvecklingen prövades fyra strategier för toppreducering: Topp-N, Peak Shaving, Daglig ombalansering och Smoothing. Topp-N valdes som huvudmetod; fördjupad information om jämförelsen finns i avsnitt 5.2.

Algoritmen får alltid in en enskild kalendermånad och följer tre steg:

1. **Identifiera toppar** - De N största timvärdena markeras.
2. **Välj mottagartimmar** - Hela månaden sorteras stigande så att lågbelastade timmar prioriteras.
3. **Flyttade last** - Energi fördes successivt från varje topp till den timme som hade lägst belastning.

Standardparametrarna som användes i simulatoren var $N = 3$ och $\alpha = 0.5$, där α anger andelen av varje topptimmes förbrukning som flyttas (dvs. 50% av toppen). Total energimängd bevarades eftersom all förbrukning drogs från en timme och adderades till en annan inom samma iteration.

Begränsning: Metoden var girig och tog inte hänsyn till restriktioner som maximal tidsförskjutning, men sådana krav kan införas genom att justera α eller lägga till extra villkor i det tredje steget.

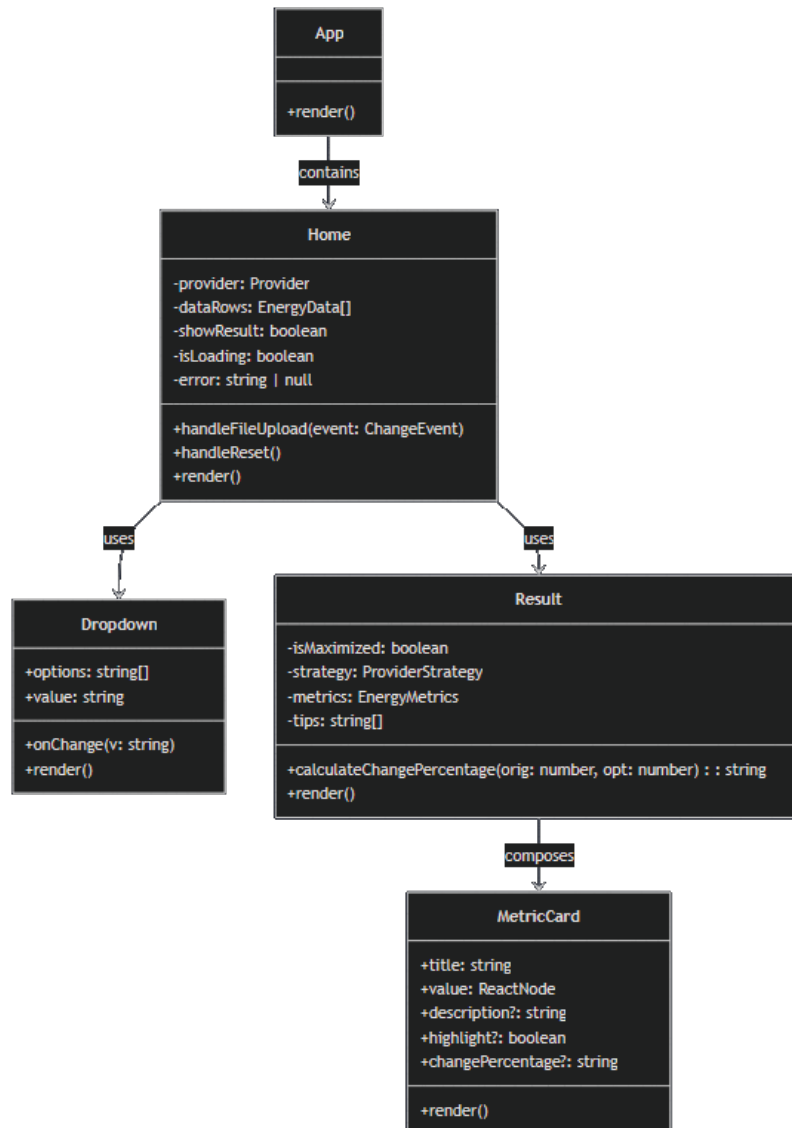
4.4 Simulators utveckling

Arbetet med simulatoren inleddes med att välja ett bibliotek för inläsning av CSV-filer i webbläsaren. PapaParse valdes eftersom det erbjöd stabil hantering av semikolonseparerad data. En valideringslogik implementerades för att sortera bort ogiltiga rader, kommentarer och rubriker samt för att konvertera strängvärden till numeriska data. Simulatoren utvecklades som en React-applikation med två huvudvyer:

- **Startsidan:** där användaren laddade upp sin fil.
- **Resultatsidan:** där optimeringen genomfördes och resultaten presenterades.

Optimeringskoden integrerades i frontend och kördes lokalt, vilket minskade behovet av backend och säkerställde att data stannade på användarens enhet. Resultaten visualiserades med Recharts. Funktioner för toppmarkering, responsiv layout och

strategispecifika rekommendationer lades till successivt. Utvecklingen bedrevs iterativt; funktionalitet och användarflöde förbättrades efter testning med realistiska data.



Figur 4.1: UML Diagram

4.5 Användargränssnitt

Användargränssnittet byggdes i React och designades med fokus på användarvänlighet. Det tillät användaren att enkelt ladda upp sina datafiler, starta optimeringen och ta del av de visualiserade resultaten. Principerna enkelhet, tydlighet och responsiv design följdes.

Användarflödet hölls linjärt: start → ladda upp data → se resultat. Alla interaktiva element markerades tydligt.

4.6 Implementation och systemtestning

Eftersom simulatoren var helt frontend-baserad fokuserade implementationen på att sammanföra användargränssnitt, datavalidering och optimeringslogik till en sammanhängande användarupplevelse. Varje delkomponent - inläsning av data, optimeringsalgoritm och visualisering - utvecklades och testades stegvis innan de integrerades i den färdiga simulatoren.

Testningen genomfördes manuellt genom att olika versioner av energidata laddades upp för att kontrollera att systemet hanterade både ogiltig och korrekt indata på ett tillförlitligt sätt. Särskild vikt lades vid att verifiera att optimeringsalgoritmen gav logiska och konsekventa resultat samt att visualiseringen återgav skillnaden mellan förbrukning före och efter optimering på ett korrekt sätt.

Responsivitet och användarflöde utvärderades genom tester på olika skärmstorlekar för att säkerställa en god användarupplevelse även på mindre enheter. Eftersom ingen backend eller databas användes skedde all testning lokalt i webbläsaren, vilket gjorde det möjligt att snabbt iterera och justera under utvecklingsprocessen.

4.7 Utmaningar och anpassningar under utvecklingen

Under utvecklingen uppstod flera utmaningar som hanterades genom tekniska och designmässiga anpassningar. En begränsning var den tillgång till testdata som fanns tillgänglig, vilket påverkade möjligheterna att utvärdera och kalibrera algoritmens prestanda under olika förbrukningsmönster.

Användargränssnittet justerades vid flera tillfällen för att uppfylla specifika krav från uppdragsgivaren. Detta inkluderade anpassningar av layout, informationsstruktur och presentationen av resultat, med målet att förbättra tydlighet och användarupplevelse.

Eftersom systemet är helt frontend-baserat och saknar backend, behövde all validering, bearbetning och visualisering genomföras lokalt i webbläsaren. Detta ställde krav på optimerad kod och en noggrann hantering av resurser för att upprätthålla god prestanda.

5

Resultat

I detta kapitel presenteras resultaten från implementeringen och testerna av simulatoren samt de algoritmer som utvecklats. Resultaten redovisas utifrån både funktionella och icke-funktionella krav, med fokus på användargränssnittets utseende, systemets prestanda, samt optimeringsalgoritmernas effektivitet.

5.1 Utseende och användargränssnitt

Användargränssnittet har utformats med fokus på enkelhet, tydlighet och tillgänglighet. Gränssnittet är uppdelat i två huvudsidor: en startsida för filuppladdning och leverantörsväl, samt en resultatsida där användaren kan visualisera och analysera sin energiförbrukning.

Designen följer följande principer för användarcentrerad utveckling:

- **Responsiv design:** Gränssnittet anpassar sig till olika skärmstorlekar och enheter.
- **Visuell hierarki:** Viktig information, som val av leverantör och filuppladdning, lyfts fram med tydliga rubriker och avstånd mellan sektioner.
- **Återkoppling:** Användaren får tydlig återkoppling vid felaktig filstruktur, under laddning och vid framgångsrik bearbetning.
- **Konsekvent färgschema:** Blå och grå färgtoner används för att vägleda användaren utan att störa informationsinnehållet.

5.1.1 Startsidan

På startsidan (Se Fig 5.1) presenteras en rubrik med instruktioner följt av ett nedfällbart val för elnätsleverantör. Därefter följer ett område där användaren laddar upp sin CSV-fil. Uppladdningsrutan har en ikon och text som beskriver formatkrav och exempel på hur man hämtar data från sin leverantör.

5.1.2 Resultatssidan

Vid en fungerande uppladdning genereras en resultatsida (Se Fig 5.2). Här visas nyckeltal såsom totalförbrukning, effektagift, toppvärden samt grafer som jämför

data före och efter optimering. Topp 3-värden markeras med ikoner (sol/måne) för att illustrera om topperna före- och efter skedde under natten eller dagen.

Ett växlande vy-läge för grafen möjliggör att användaren kan maximera och fördjupa sin analys. Slutligen visas även textbaserade tips och exempel på energiförflyttningar mellan tidpunkter, vilket förstärker förståelsen för algoritmens effekt.

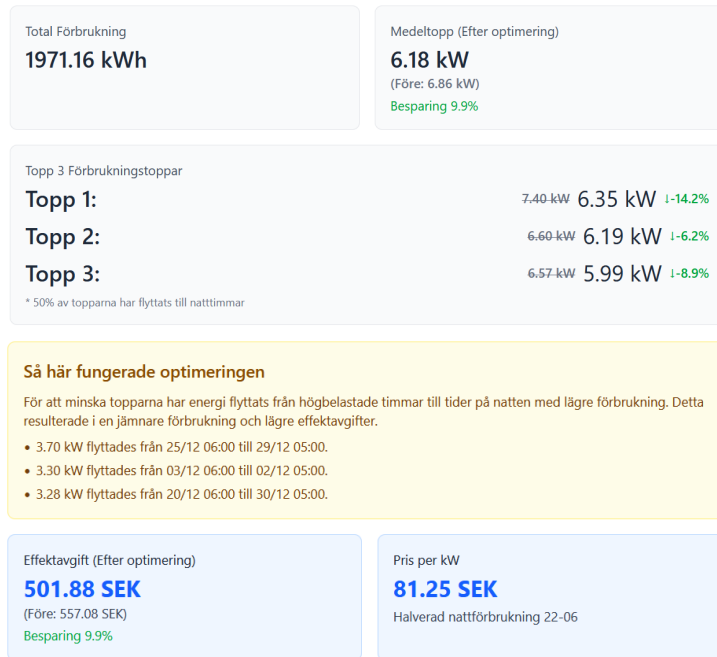
Syftet med gränssnittet är att på ett pedagogiskt och tillgängligt sätt möjliggöra insikt i hushållets energianvändning och därigenom underlätta för konsumenten att fatta informerade beslut.



Figur 5.1: Hemsida

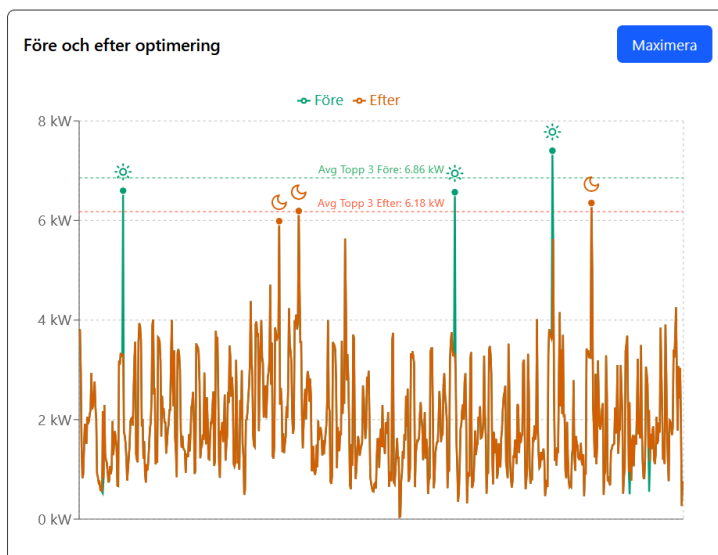
Optimeringsresultat

Optimeringsmetod : Top-N-optimering: Minska toppar genom att fördela energi från de högsta N posterna.



Tips för att minska effekttoppar

- Flytta energiförbrukning till nattetid enligt optimeringen för att minska din effektavgift.
- Identifiera och minimera perioder med hög förbrukning.
- Schemalägg tunga laster (tvätt, disk) till lågprisperioder.
- Investera i energieffektiva vitvaror och LED-belysning.
- Sänk värmen med 1-2 °C och ventiler kort men effektivt.
- Koppla ur standby-apparater och använd smarta timers.
- Följ förbrukningen i realtid med en energimonitor.
- <https://www.ellevio.se/abonnemang/ny-prismodell-baserad-pa-effekt/#h-sa-kan-du-sanka-dina-effekttoppar>



Figur 5.2: Resultatsida

5.2 Algoritmer och optimeringsresultat

Under algoritmutvecklingen prövades fyra huvudmetoder för toppreducering: Topp-N, Peak Shaving, Daglig ombalansering och Smoothing. Teorin om dessa metoder är i avsnitt 2.3:

- Topp-N markerade de N största timmarna och förde över en andel av varje topp direkt till de lägsta timmarna, vilket gav snabb dämpning av de högsta effekttopparna.
- Peak Shaving beräknade ett lokalt medelvärde i ett glidande fönster och fördelade överskottet från topparna till intilliggande timmar, vilket var effektivt för kortvariga peakar.
- Daglig ombalansering jämnade ut timvärdena inom varje kalenderdygn genom att flytta energi från timmar över dygnsmedelvärdet till timmar under, vilket gav en jämn dygnskurva.
- Smoothing använde glidande eller exponentiell utjämning för att dämpa högfrekvent brus i hela profilen, vilket resulterade i en slätare kurva men inte direkt riktade in sig på de största topparna.

Topp-N, med parametern $N = 3$ valdes som huvudmetod och parameter eftersom den kombinerade snabb bearbetning, hög grad av toppreduktion och enkel kommunikation av resultaten till användaren. Metoden krävde inga beräkningar av glidande medelvärden eller komplexa iterativa procedurer, vilket förenklade implementeringen och minskade risken för fel vid manuella avvikelser i indata. För framtida arbete finns möjlighet att utveckla adaptiva varianter där parametrarna N eller α justeras automatiskt utifrån förbrukningsmönster, samt att integrera prediktiva modeller för att förbättra metodens träffsäkerhet vid topphantering.

5.3 Förbättringar och utvecklingspotential

5.3.1 Genomförda förbättringar

Under utvecklingsprocessen har flera förbättringar genomförts för att öka systemets användbarhet och precision. Gränssnittet har iterativt förbättrats i enlighet med de krav och önskemål som identifierats i dialog med Grid Collective, vilket har lett till en tydligare och mer användarvänlig upplevelse. Optimeringsalgoritmen har även justerats för att ge mer realistiska optimeringsrekommendationer som bygger på teoretiska scenarier av hur energianvändningen kunde ha sett ut vid en optimerad fördelning.

5.3.2 Utvecklingspotential

Trots dessa förbättringar finns det flera områden med fortsatt utvecklingspotential. Ett centralt förbättringsområde är automatisering av datainhämtning. För närvarande krävs manuell import av konsumtionsdata i form av CSV-filer, vilket kan

förenklas genom att implementera automatiserad insamling av data direkt från elbolagens hemsidor via webscraping. Detta skulle ge användaren en smidigare upplevelse och säkerställa att systemet alltid arbetar med uppdaterad information.

Vidare kan prisstrukturerna i systemet förbättras genom att ersätta hårdkodade värden med dynamiskt hämtad prisinformation från elbolagens webbplatser. Detta skulle öka systemets flexibilitet och möjliggöra stöd för fler leverantörer utan behov av manuell uppdatering av koden.

Ytterligare förbättringspotential finns i att på sikt införa ett backend-lager för att möjliggöra lagring av historisk data och användarspecifika inställningar. Detta skulle möjliggöra återkommande användning, förbättrad personalisering och mer avancerad analys över längre tidsperioder.

6

Diskussion och slutsats

I detta kapitel diskuteras projektet i relation till uppsatta mål, begränsningar och användningsområden. Dessutom presenteras reflektioner kring genomförandet och möjliga framtida utvecklingsvägar.

6.1 Sammanfattning av resultat

Projektet har resulterat i en fungerande simulator som visualiserar och optimerar energiförbrukning på hushållsnivå. Gränssnittet har utformats med fokus på tydlighet och tillgänglighet, och systemet uppfyller kraven på användarvänlighet. Prestandatester visar att topp-n algoritmen kan hantera 720-744 datapunkter i realtid utan att påverka användarupplevelsen negativt. Resultaten visualiseras med grafer och nyckeltal, vilket underlättar förståelsen för energiförflyttning och kostnadsbesparing.

6.2 Uppfyllelse av mål och frågeställningar

Projektet uppfyller de huvudsakliga målen enligt projektplanen. En prototyp av en optimeringsalgoritm har utvecklats och integrerats i ett fungerande användargränssnitt. Simulatorens klarar att analysera timvis energidata och ge realistiska rekommendationer baserade på teoretiska scenarier utifrån användarens specifika förbrukningsmönster. Filinläsning, leverantörsväl och datavisualisering fungerar i enlighet med de funktionella kraven. Optimeringen baseras på ett konkret problem: att minska toppar i effektförbrukning, och resultaten visar att detta kan uppnås på ett begripligt och användbart sätt för användaren.

6.3 Begränsningar och felkällor

Trots goda resultat finns vissa begränsningar. Simulatorens är beroende av korrekt formaterad input från användaren. Vid val av olika optimeringsmetoder är det viktigt att beakta hur realistiska dessa är i praktiken. En betydande felkälla uppstår i och med att man inte kan förutsäga hur väl användaren faktiskt kommer att kunna följa de rekommenderade optimeringsstrategierna. Det är därför viktigt att betona att rekommendationerna baseras på teoretiskt möjlig, men inte perfekt, användning - där viss flexibilitet i elanvändningen antas. Detta innebär att algoritmens effekti-

vitet i verkliga tillämpningar kan påverkas av individuella variationer i följsamhet och beteende.

6.4 Praktiska och teoretiska implikationer

Ur ett praktiskt perspektiv kan simulatorn hjälpa hushåll att identifiera energibe- teenden som leder till onödigt höga avgifter och därmed bidra till både ekonomisk besparing och avlastning av elnätet. Den användarcentrerade designen och det fak- tum att all beräkning sker lokalt i webbläsaren gör systemet tillgängligt och säkert att använda utan integritetsrisker. Teoretiskt visar projektet hur enkla optimerings- principer kan tillämpas i en verklig energikontext, och det ger en grund för forskning bl.a om lokal energihantering och användarstyrda styrsystem.

6.5 Förslag på framtida arbete

För att öka systemets tillförlitlighet och användbarhet finns flera möjliga utveck- lingsspår. Ett naturligt nästa steg är att automatisera datainsamlingen genom att integrera systemet med elbolagens API:er eller via webbscraping. Det vore även fördelaktigt att ersätta hårdkodade prisstrukturer med dynamiskt inhämtade priser från externa källor. På längre sikt bör ett backend-lager övervägas för att lagra histo- rik och möjliggöra mer avancerad användaranalys, personalisering och kontinuerlig förbättring av algoritmens träffsäkerhet.

6.6 Reflektion kring teknikval

6.6.1 React och TypeScript

Teknikvalen i detta projekt gjordes huvudsakligen utifrån rekommendationer från företaget. React med TypeScript användes efter en tydlig rekommendation från fö- retaget, främst eftersom deras befintliga projekt och interna rutiner var anpassade efter dessa tekniker. Att använda React säkerställde också att projektet enkelt kunde integreras i företagets befintliga kodbas och arbetsflöden.

Att vi valde TypeScript tillsammans med React hade ytterligare fördelar, framförallt vad gäller förbättrad kodkvalitet och underhållbarhet. Den statiska typningen bidrog till färre fel under utvecklingsfasen och gav en tydligare struktur som underlättade samarbete mellan utvecklare.

6.6.2 Vite

Vite valdes framför andra byggverktyg som Webpack främst på grund av dess pre- standan och snabba utvecklingscykel. Detta beslut visade sig vara effektivt, då Vites snabba uppstart och effektiva hot-reloading avsevärt förbättrade utvecklarupplevel- sen och minskade tiden som gick åt till konfigurering och felsökning.

6.6.3 Tailwind CSS

Användningen av Tailwind CSS motiverades av behovet av ett snabbt och effektivt sätt att utveckla ett stilrent och responsivt gränssnitt. Genom att använda Tailwinds utility-klasser kunde vi direkt visualisera och prototypa designidéer utan att lägga tid på att skapa och underhålla separata CSS-filer. Detta gjorde att vi kunde fokusera mer på applikationens funktionalitet och användarupplevelse.

6.6.4 Tredjepartsbibliotek

Integrationen av tredjepartsbibliotek som PapaParse och Recharts var ett strategiskt beslut för att snabbt kunna implementera avancerad funktionalitet, exempelvis dataimport och visualisering, utan att behöva utveckla dessa delar från grunden. Dessa bibliotek var välbeprövade, vilket gav trygghet i att de skulle fungera stabilt även när projektet skalades upp.

6.6.5 Hållbar utveckling

Projektet har en tydlig koppling till hållbar utveckling genom sitt fokus på att effektivisera energianvändningen i svenska hushåll. Genom att minska toppbelastningar och optimera elanvändningen på individnivå kan man i förlängningen bidra till ett mer stabilt och motståndskraftigt elsystem. Simulatorens gör det möjligt för användare att förstå hur deras konsumtionsmönster påverkar både elkostnaden och belastningen på elnätet, vilket främjar medvetna, långsiktigt hållbara beslut.

Utöver miljönyttan främjar projektet även ekonomisk hållbarhet genom att ge hushållen verktyg för att sänka sina elkostnader. Samtidigt stärker det den sociala hållbarheten genom att öka individens delaktighet i energiomställningen - något som är avgörande för en rättvis och inkluderande utveckling.

Det faktum att all databehandling sker lokalt i webbläsaren innebär dessutom att systemet har låg påverkan på servrar och infrastruktur, vilket minskar dess klimatavtryck.

Litteraturförteckning

- [1] Y. Numata, L. Speelman, and M. Gantman, “The Energy Transition Is a Technological Revolution - with a Deadline,” Rocky Mountain Institute (RMI), 2023. [Online]. Available: <https://rmi.org/the-energy-transition-is-a-technological-revolution-with-a-deadline/>. [Accessed: May 25, 2025].
- [2] International Energy Agency, “Rapid progress of key clean energy technologies shows the new energy economy is emerging faster than many think,” IEA News, Jul. 2023. [Online]. Available: <https://www.iea.org/news/rapid-progress-of-key-clean-energy-technologies-shows-the-new-energy-economy-i>. [Accessed: May 25, 2025].
- [3] D. Keiner, M. Ram, L. D. S. N. Barbosa, D. Bogdanov, and C. Breyer, “Cost optimal self-consumption of PV prosumers with stationary batteries, heat pumps, thermal energy storage and electric vehicles across the world up to 2050,” *Solar Energy*, vol. 185, pp. 406-423, 2019. doi: 10.1016/j.solener.2019.04.081.
- [4] A. Tavakoli, S. Saha, M. T. Arif, M. E. Haque, N. Mendis, and A. M. T. Oo, “Impacts of grid integration of solar PV and electric vehicle on grid stability, power quality and energy economics: a review,” *IET Energy Systems Integration*, vol. 2, no. 3, pp. 243-260, 2020. doi: 10.1049/iet-esi.2019.0047.
- [5] H. Shareef, M. S. Ahmed, A. Mohamed, and E. S. Al Hassan, “Review on Home Energy Management System Considering Demand Responses, Smart Technologies, and Intelligent Controllers,” *IEEE Access*, vol. 6, pp. 24498-24509, 2018. doi: 10.1109/ACCESS.2018.2831917.
- [6] D. Giofrè, G. Manzolini, S. Leva, R. Jaboeuf, P. Tosco, and E. Martelli, “Quantifying the Economic Advantages of Energy Management Systems for Domestic Prosumers with Electric Vehicles,” *Energies*, vol. 18, no. 7, article no. 1774, 2025. doi: 10.3390/en18071774.
- [7] Grid Collective, “Om oss,” Grid Collective, 2025. [Online]. Available: <https://www.gridcollective.se/om-oss> [Accessed: May 24, 2025].
- [8] Meta, “React documentation,” React, 2025. [Online]. Available: <https://react.dev> [Accessed: May 24, 2025].
- [9] Evan You and Contributors, “Vite documentation,” Vite, 2025. [Online]. Available: <https://vite.dev> [Accessed: May 24, 2025].

- [10] Webpack Contributors, “Webpack documentation,” Webpack, 2025. [Online]. Available: <https://webpack.js.org/> [Accessed: May 29, 2025].
- [11] Kinsta, “Vite vs Webpack: A detailed comparison,” *Kinsta Blog*, Apr. 2024. [Online]. Available: <https://kinsta.com/blog/vite-vs-webpack/#5-build-time-and-bundle-size>. [Accessed: Jun. 15, 2025].
- [12] Microsoft, “TypeScript documentation,” TypeScript, 2025. [Online]. Available: <https://www.typescriptlang.org> [Accessed: May 24, 2025].
- [13] MDN Web Docs, “JavaScript documentation,” MDN, 2025. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [Accessed: May 25, 2025].
- [14] Tailwind Labs, “Tailwind CSS documentation,” Tailwind CSS, 2025. [Online]. Available: <https://tailwindcss.com> [Accessed: May 24, 2025].
- [15] M. Holt, “PapaParse GitHub repository,” GitHub, 2025. [Online]. Available: <https://github.com/mholt/PapaParse> [Accessed: May 24, 2025].
- [16] Recharts Authors, “Recharts GitHub repository,” GitHub, 2025. [Online]. Available: <https://github.com/recharts/recharts> [Accessed: May 24, 2025].
- [17] R. Dharani, M. Balasubramonian, T. S. Babu and B. Nastasi, “Load Shifting and Peak Clipping for Reducing Energy Consumption in an Indian University Campus,” *Energies*, vol. 14, no. 3, article 558, 2021. [Online]. Available: <https://doi.org/10.3390/en14030558> [Accessed: May 24, 2025].
- [18] S. V. Oprea, A. Bâra and G. Ifrim, “Flattening the electricity consumption peak and reducing the electricity payment for residential consumers in the context of smart grid by means of shifting optimization algorithm,” *Computers & Industrial Engineering*, vol. 122, pp. 125-139, 2018. [Online]. Available: <https://doi.org/10.1016/j.cie.2018.05.053> [Accessed: May 25, 2025].
- [19] R. J. Hyndman and G. Athanasopoulos, “Moving averages,” *Forecasting: Principles and Practice*, 2021. [Online]. Available: <https://otexts.com/fpp3/moving-averages.html> [Accessed: May 25, 2025].

INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK
CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige
www.chalmers.se



CHALMERS