RACE for 2030
RELIABLE AFFORDABLE CLEAN ENERGY

Final report

Ask The Energy System:

AI Assisted Energy Modelling

August 2025

Australian Government
Department of Industry, Science and Resources
Cooperative Research Centres Program

Final report

RACE for Networks

Research theme: NT6 Developing local grid solutions

Ask The Energy System: AI Assisted Energy Modelling

24.NT6.R.772

ISBN: 978-1-922746-76-4

August 2025

Project team

Monash University

- Markus Wagner
- Hao Wang
- Terrance Mak

RedGrid

- Simon Wilson
- Jason Clark

Project partners

Acknowledgements

The project team and partners would like to acknowledge the participation and contribution through the 'discovery phase' of this activity of many Retail, Utility, and Distributed Network Provider staff and management for the 27 individuals who provided insights to our process and provided feedback on the platform, software, direction, and 'jobs to be done' focus.

Your feedback has been crucial and immensely valuable to informing our work and direction and has provided a clear path and direction for the extension of our work to extend and enable the GridGuru platform and ecosystem.
The team would also like to acknowledge the Monash researchers and individuals who contributed to our findings and direction. We look forward to working more deeply with you all in Phase 2 of this project.

Acknowledgement of Country

The authors of this report would like to respectfully acknowledge the Traditional Owners of the ancestral lands throughout Australia and their connection to land, sea and community. We recognise their continuing connection to the land, waters, and culture and pay our respects to them, their cultures and to their Elders past, present, and emerging.

What is RACE for 2030?

Reliable, Affordable Clean Energy for 2030 (RACE for 2030) is an innovative cooperative research centre for energy and carbon transition. We were funded with $68.5 million of Commonwealth funds and commitments of $280 million of cash and in-kind contributions from our partners. Our aim is to deliver $3.8 billion of energy savings and 20 megatonnes of cumulative carbon emission savings by 2030.
racefor2030.com.au

Contents

1 Executive Summary

Introduction

1.1 Background

This project, AsktheEnergySystem, aimed to build a prototype AI energy platform called GridGuru. The RedGridGPT GridGuru platform is a flexible and innovative solution designed to model energy grids at various scales, from individual households to entire national grids. Leveraging agent-based data and machine learning algorithms, the platform offers detailed, real-time insights into energy consumption, generation, and distribution. It enables users to simulate different scenarios, forecast energy demands, and optimise grid operations under varying conditions, including the integration of renewable energy sources and the management of electric vehicle (EV) charging infrastructure under an interactive and immersive interface.

Based on a fully extensible platform with open integrations allows GridGuru to incorporate diverse data sources, such as smart meter data, weather information, and economic indicators and more through an open and growing ecosystem.

This capability is crucial for addressing complex energy management challenges with many devices and actors. It provides an aggregation platform for services such as visualising low voltage circuits, planning for community battery installations, and ensuring grid stability amid increasing renewable energy penetration where value is stacked and shared within and between models.

The platform's user-friendly interface and generative AI integration for conversational interaction also provides a powerful visualisation and configuration tool that is accessible and usable for many stakeholders, including utilities, policymakers, and community energy groups, and even non- technical teams and individuals enabling them to make data-driven decisions to enhance grid reliability and efficiency and plan their projects.

1.2 Mission and Vision Statement

Vision Statement

Leading Australia and the world to a net-zero future
through open-source energy solutions and global collaboration.

Through GridGuru, we envision a future where Australia leads the world in sustainable energy practices, achieving unparalleled progress toward net-zero emissions. Our vision is to create a global, open-source platform that not only addresses local grid challenges but also inspires international collaboration and democratises renewable planning and innovation.

Within the first three years, GridGuru will have piloted transformative programs, integrated critical data feeds, positioning ourselves as a key enabler in the global energy transition. Through our efforts, we strive to empower innovators and professionals to create a cleaner, smarter, and more resilient energy future for everyone.

Mission Statement

Open, extensible, intelligent energy system modelling tools for everyone

GridGuru is dedicated to accelerating Australia's transition to a net-zero future by providing an open and accessible platform for utilities, retailers, and new energy providers. Our mission is to empower businesses, governments, and communities with critical information on low voltage grid capacity, enabling informed decisions and equitable integration of renewable energy sources. Through innovative Gen-AI energy modelling and data-sharing capabilities, GridGuru will lower the barriers to energy transition, fostering a collaborative environment where data and insights are freely exchanged and utilised for the greater good.

Gen-AI and Large Language Model (LLM) integrated Energy Modelling

Integration of Generative AI (Gen-AI) and Large Language Model (LLM) features into the energy modelling software brings significant business benefits to enhance decision-making for various stakeholders. Through these integrated services we rapidly process and analyse vast amounts of local, proprietary, and global data that enables energy modelers to optimise their resources more efficiently, predict future energy demands, and identify trends that balance cost, efficiency, and sustainability according to their unique and specific needs and circumstances, project, and business objectives.

Scenario analysis

Through its LLM and GenAI integration features users can map and visualise their energy environment and then conversationally interact with them to evaluate and test complex scenarios. Through Machine Learning forecasting capabilities users are also able to predict potential outcomes. Stakeholders have expressed that they see significant value in this capability to make proactive and informed decisions, test concepts, ensure readiness for diverse possibilities, and improve overall strategic planning.

Real and Near-Real time monitoring

The platform's primary objective currently is for scenario analysis and planning and not specifically real time operational monitoring. However, the platform team have through architecture design and by virtue of the use of the Unity Open Integrations platform enabled a highly modular and 'open extensible' approach to device and data integrations.

The architecture allows the user to integrate and combine various data types and shapes including static data, mock sources, and real time API feeds.

Moreover, the open-source nature of the Unity platform provides a baseline for community developed new and updated integrations.

From a technical perspective our project has also found that Generative AI and LLM features provide powerful features in the service to deal with 'imperfect data' scenarios within projects as it can use Generative services and Machine Learning against the LLM to begin to fill certain gaps where appropriate.

This is one of the key value propositions for planning practitioners who reported consistently that they are dealing with varied and imperfect information.

Regulatory Compliance and Reporting

New strategic reporting standards are a key requirement for businesses and network providers and involve integrated and contextual planning to navigate the complex interplay between new technologies, market dynamics, and regulatory frameworks and proprietary objectives.

Gen-AI integration with modelling capability allows users to integrate their project's specific strategic and unique data and documentation with their environments and to use that information as context when interacting with energy models. This allows them to simulate different policies and decisions, propose strategies and plan long-term energy needs and has emerged from our discovery to be a key value proposition for the platform.

Democratise energy planning

Internally, AI-driven tools, such as chatbots powered by ChatGPT, can assist teams with self- service access to data insights, explain results, and provide personalised recommendations. This fosters better team collaboration, faster decision-making, and enhanced customer satisfaction through intuitive dashboards and interactive support, ultimately leading to a more efficient, reliable, and sustainable energy system for all stakeholders.

Through conversational integration features users can interact and ask questions on their own terms and based on their own level of expertise.

This also has emerged as a powerful value proposition for the platform in that it can begin to **'democratise' the energy planning function** and when combined with an open-source foundation can empower a whole new community of clean energy innovators and innovation.

2 Project

2.1 Overview

In January 2024 Redgrid commenced work on a four-month feasibility project with Monash University where the objective was to test and mature an energy modelling platform that RedgridGPT built on the world's leading 2D, 3D and VR Gaming Platform, Unity.

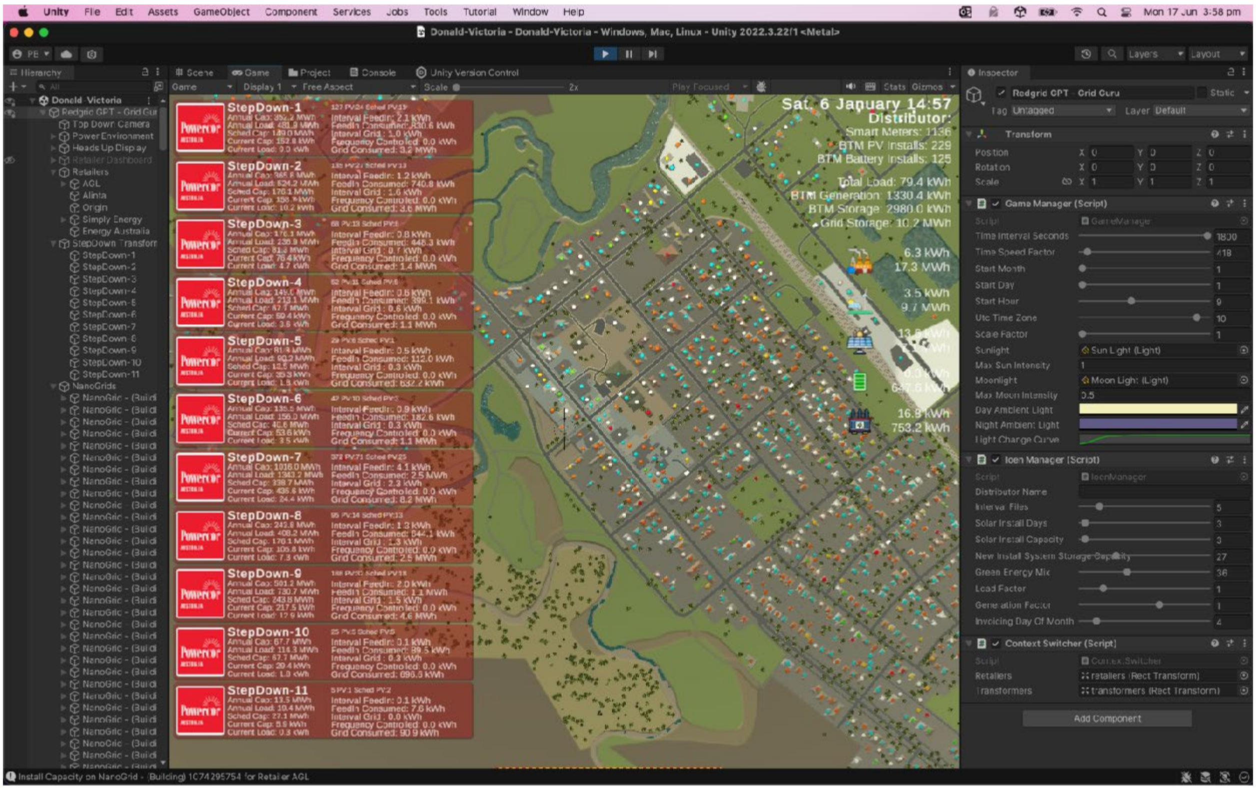

Figure 1: Screenshot of GridGuru on Unity platform

That platform, GridGuru, provides Energy Designers and Managers with a visual and intuitive interface and services on which they can see and measure energy flows and transactions, add change and intermingle real and simulated energy assets, and most importantly run and evaluate scenarios such as weather events, usage changes, or impacts of new or changing infrastructure.

The project focus was to integrate new and emerging Machine Learning, Generative AI, and Large Language Model (LLM) capabilities into the software to demonstrate the value of a fully interactive modelling tool that through Generative AI could provide users with the ability to ask questions and propose changes on their own terms and receive responses with local and relevant context to their needs and objectives.

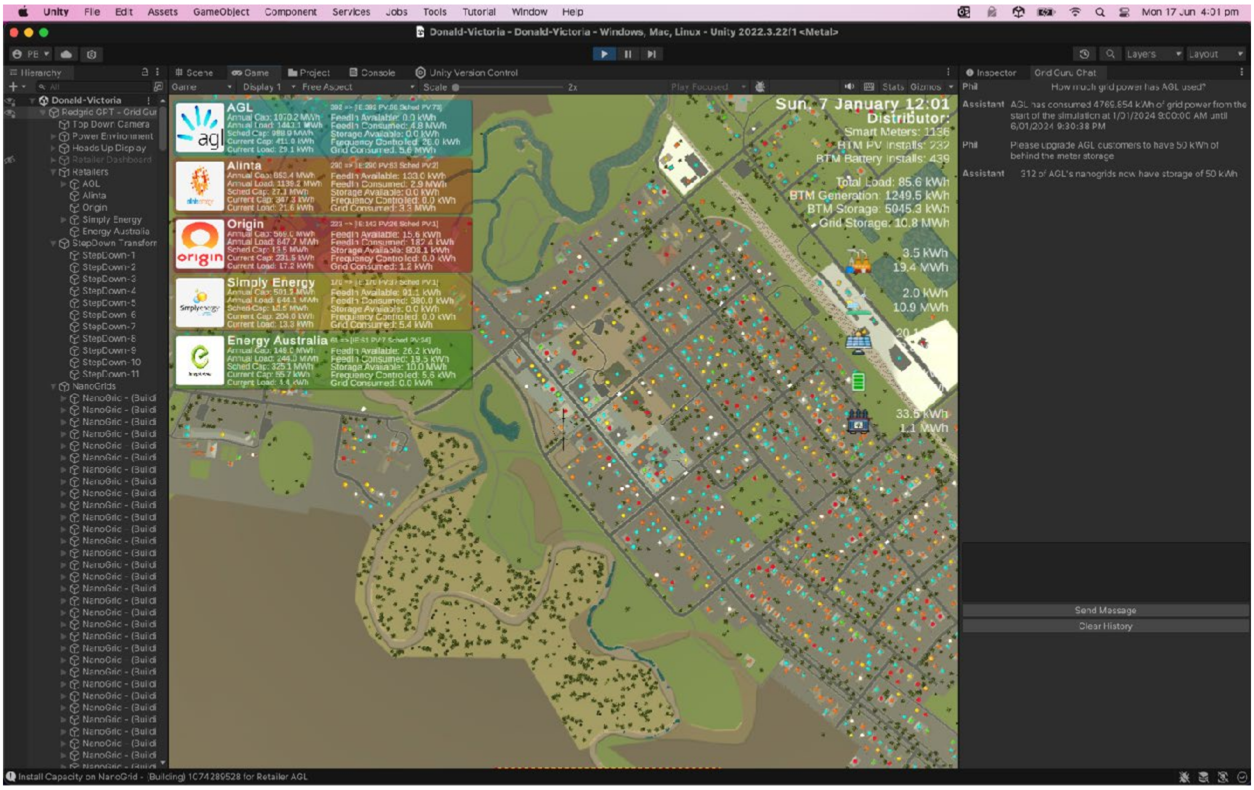

Figure 2: Screenshot of GridGuru on Unity platform

2.1.1 Methodology

The project was structured across four key work packages:

- Work Package 1:

  Product Discovery and Market Research Identify and understand potential users' needs and pain points for GridGuru.
- Work Package 2:

  Customer and User Validation.
- Work Package 3:

  MVP Prototype and Data Forecasting: Develop a prototype that demonstrates GridGuru's AI's core functionality with five MVP LLM questions. Improve the model data accuracy and representation, focusing on 5 min forecasting and benchmarking. Is the data accurate and is the AI answer what the user expected?
- Work Package 4:

  Platform Roadmap and Phase 2 Business Case: Develop a strategy that outlines the goals, funds, and roadmap to develop a fully serviced offering that provides an innovative and effective solution to help Australia meet its Net Zero Goals. Analyse funding pathways and routes and high target opportunities that align with RedGrid's goals.

This project has successfully demonstrated the value of the platform/architecture and has shortlisted 3 Customer Use Cases and has mapped the direction for its further growth and expansion that we envisage will be realised in a phase 2 extended 12-month project.

3 Work Package 1: Produce Discovery and Market Research

3.1 Methodology

To comprehensively understand the market needs and validate GridGuru, we conducted a rapid literature review to identify current problems, challenges, and trends in energy modelling software. Based on these insights, we developed a Minimum Viable Product (MVP) demo showcasing GridGuru's capabilities in real-time 2D and 3D visualisation, scenario analysis, and data integration.

We engaged InnovateGPT, a consultancy, to assist in our product discovery phase by facilitating workshops and interviews with energy professionals, including utility managers and DNSPs. This helped us gather valuable feedback, identify specific use cases for GridGuru, and develop product hypotheses to test with the market.

Additionally, we performed market research focused on utilities and DNSPs, collecting data through market research and industry reports to understand their needs and pain points. We conducted a competitor analysis to position GridGuru effectively, identifying key players and understanding their strategies including the discovery interviews with 27 individuals.

This multi-faceted approach ensured that our product development was aligned with industry needs and market demand, providing a solid foundation for GridGuru's successful launch.

3.2 Findings

3.2.1 Market Research of the Low Voltage Network

Australia's low voltage (LV) network is managed by Distribution Network Service Providers (DNSPs) or Distribution Network Operators (DNOs), which deliver electricity from high voltage transmission networks to residential and small commercial consumers.

Components of the LV Network:

- Distribution Substations: Step down high voltage electricity to levels suitable for end users.
- Low Voltage Distribution Lines: Carry electricity from substations to homes and businesses.
- Meters and Customer Connections: Measure electricity usage and connect end users to the network.

Key Stakeholders:

- **Consumers:** Residential, commercial, and small industrial users.
- **Retailers:** Companies that sell electricity to end users and manage billing and customer service.
- **DNSPs/DNOs:** Manage the physical distribution network infrastructure.
- **Regulators and Market Operators:** Ensure compliance, manage market operations, and facilitate energy trading.
- **Community Groups:** Engage in local energy projects and initiatives.
- **New Energy Providers:** EV charging companies and other emerging energy providers.

Opportunities within the LV Network

The rapid uptake of consumer decentralised energy resources (DER) presents significant opportunities. Australia leads the world in per capita solar installation, enabling the use of advanced metering and smart grid technologies to provide real-time data on energy usage. This facilitates accurate demand forecasting, dynamic pricing, and improved load management, which can result in cost savings for both consumers and utilities.

Further opportunities include:

- **Demand Response Programs:** Balance supply and demand efficiently.
- **Energy Storage Solutions:** Store excess renewable energy for later use.
- **EV Charging Infrastructure:** Expand to support the growing number of electric vehicles.
- **Automated Systems:** Enhance operational efficiency and reliability through modernisation.
- **Data Analytics:** Improve decision-making and asset management by leveraging data insights.
- **Community Energy Projects:** Empower local communities to generate, store, and manage their own energy, increasing resilience and economic benefits.

These advancements support Australia's transition to a more resilient, efficient, and sustainable energy future by leveraging new technologies, regulatory frameworks, and market mechanisms to create a more flexible and robust LV network.

Current Challenges within the LV Network

The rise of rooftop solar panels, home battery storage systems, and electric vehicles (EVs) has increased distribution and decentralised generation within the LV network. This decentralisation challenges DNSPs with voltage management, grid stability, and integration of intermittent renewable energy sources. Smart meters, grid automation, and advanced monitoring technologies are being deployed to improve the efficiency, reliability, and responsiveness of the LV network. However, access to this data is often limited, making it difficult for new energy providers, such as EV companies and Virtual Power Plants (VPPs), to make informed grid management decisions.

The 2024 Draft AEMO ISP Report states significant challenges exist in understanding and optimising the low voltage network capacity to support the energy transition. Modernised distribution networks are needed to connect low-cost renewable resources to consumers, and readiness of market and power system operations for high renewable energy penetrations is crucial.

Data Accessibility and Sharing

The Grattan Report, "Keeping the Lights On," emphasises the need for better coordination and integration of DER, advocating for data sharing among various stakeholders, including utilities. Effective data sharing and coordination can mitigate unintended consequences of DER integration, ensuring resource adequacy and affordability in the National Electricity Market (NEM).

According to the EV Council report "EV Charging and the Grid: Impact to 2030 in Australia," data sharing between vehicle registration bodies and energy networks is essential for early detection of EV clustering. Capturing EV charging equipment installation data in a register accessible to DNSPs can help manage and predict grid impacts. Improved visibility within the LV network through monitoring hardware and software for transformers can assess and mitigate the impact of EV clusters on the grid.

AEMOs 2024 ISP Report, a notable document in informing the future strategy of the Australian grid, emphasises the need for coordination and appropriate operational standards for integrating DER. Whilst it does not explicitly state the need for better data sharing, it highlights challenges in understanding network capacity and recommends enhancing the LV network through storage solutions, CER integration, modernising distribution networks, and timely investments in transmission projects.

The AEMO 2024 Draft ISP Key actions and recommendations are:

- Enhancement of Storage Solutions: Increase in installed capacity and energy storage capacity is forecasted, requiring various storage solutions to firm renewables.
- Integration of Consumer Energy Resources (CER): Overcome technical complexities and build trust among CER owners to participate in VPPs.
- Modernising Distribution Networks: Essential for delivering electricity to homes and businesses and taking back surplus from consumer assets.
- Timely Investment in Transmission Projects: Urgent delivery of transmission projects to meet future demands and support the energy transition.

3.3 Product overview / what we're testing

To support our company mission and vision, Redgrid have built an initial version (or minimal viable product) of GridGuru, that captures the core features we believe the market will find valuable. These set of features will be tested during the discovery calls with industry specialists and target customers.

The full architecture and build of the MVP is documented within work package 3.

Key differentiators for V1:

- Open source to differentiate from competition and improve community accessibility, which are almost entirely closed, or enterprise focused.
- Highly user friendly by introducing LLM layer on-top of energy data to direct and ask questions of the model.
- Extensible and rapidly scalable by using Unity as the base platform.
- Accessible energy data for the low voltage network

Open-source approach

There is common consensus from all our discovery interviews that there is a core data quality and access problem that prevents energy innovation.
Various initiatives like AEMOs Data Interchange, CSIRO NEAR, CSIRO NationalMap, and others are working to enable a more open system for valuable energy data sets to be shared and accessible however most are organisation managed and maintained.

From our interviews we propose that there is a need for a true open-source, community curated service that would combine an aggregation point for these multiple existing services under a community-driven, community governed and community curated open- source foundation.

Our phase 2 project will further define such an architecture and approach that will balance security, privacy and open data to provide a useful base on which energy planners can pursue their goals.

Product value hypothesis

The product and value hypothesis that we tested with the market is:

For energy managers and infrastructure providers:

We believe that GridGuru helps Utilities and DNSPs who want to model and improve the low voltage network, by **reducing the time and complexity** it takes to do scenario modelling, **improve grid outcomes, and reduce costs,** through **faster and more robust** decision-making.

For beneficiaries of the open-source model:

We believe that GridGuru helps communities, energy providers and regulators, who want to understand grid capacity and test different Net Zero initiatives, by **reducing the barriers to acquire accurate energy data,** and **enables projects to achieve outcomes through improved knowledge sharing and quicker approval times.**

3.3.1 GridGuru v0.1: Overview and Features

During the build of V1, RedgridGPT prioritised the following MVP feature set.

GridGuru is an advanced energy grid modelling and optimisation platform that leverages agent-based energy data for a fully extensible solution built off Unity. It focuses on the low voltage network, enabling users to rapidly map and model any area using OpenStreetMap data. Key features are:

1. Agent Based Energy Data
   For different energy assets across the low voltage network:
   a. Stepdown Transformers, NEMI data, Solar, Battery, EV and Community Battery

2. Data Integration:
   Incorporate real-time data or representative mock data for accurate modelling.
   a. Real-time, historic, forecast.

3. Low Voltage Network Focused:
   a. Specialised tools for mapping and optimising low voltage networks.
   b. Quickly map and model any area using OpenStreetMap data.

4. Scenario Modelling:
   Interactive and intuitive scenario modelling enabled by Unity.
   a. Configure multiple parameters of the model
   b. Weather, time / day / month, tariff structure, renewable %, DER %.

5. Machine Learning Predictions:
   Predict future energy use with advanced machine learning models.
   a. 5min NEMI load forecast
   b. 5 min Solar load forecast

6. Gen-AI LLM Models:
   Query and update the model using advanced Gen-AI large language models.

7. Proprietary Data Integration:
   Add company policies, products, and energy regulations to the LLM.

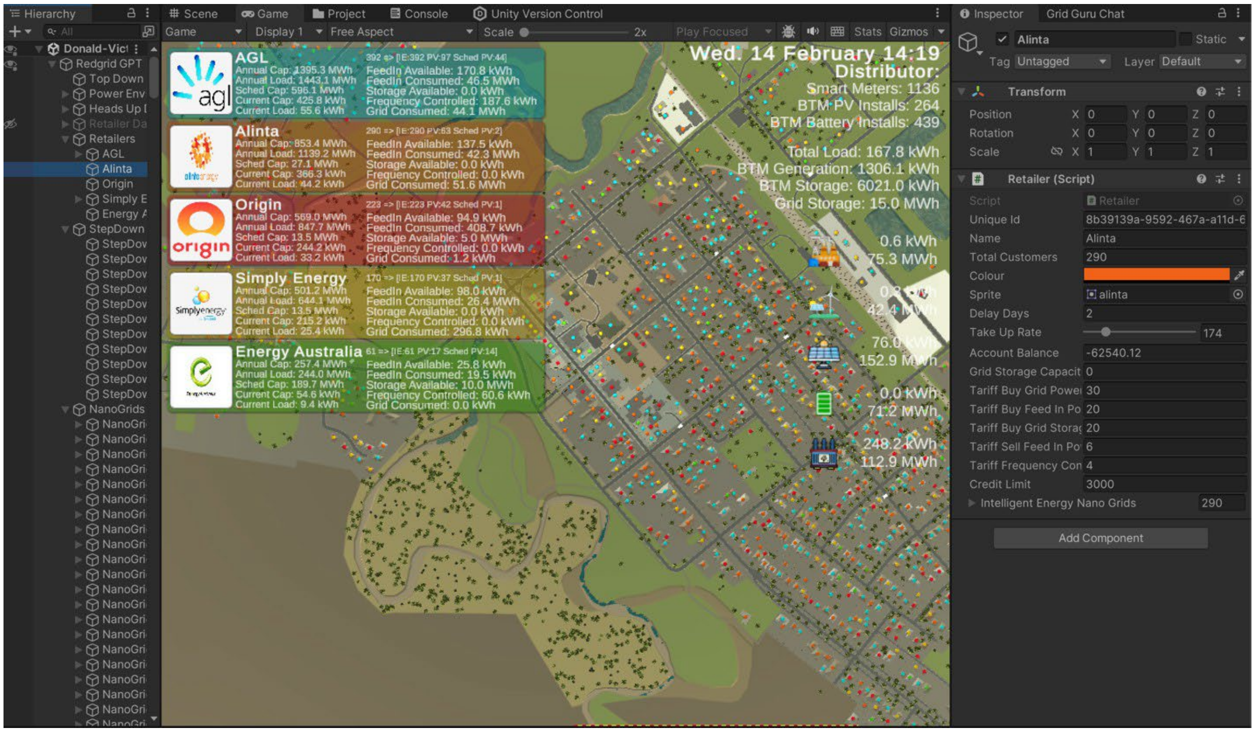

Figure 3: Screenshot of Retailer view

Figure 4: Screenshot of Low Voltage Networks view

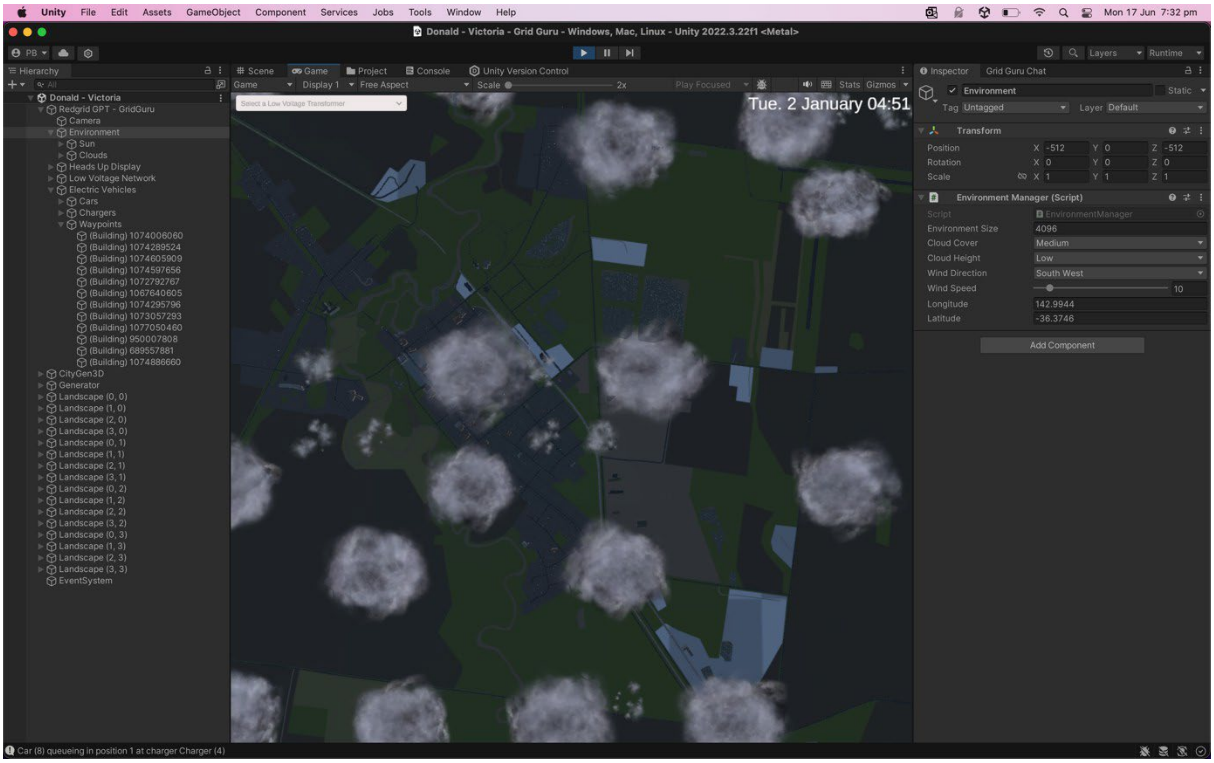

Figure 5: *Screenshot of Control Cloud Cover*

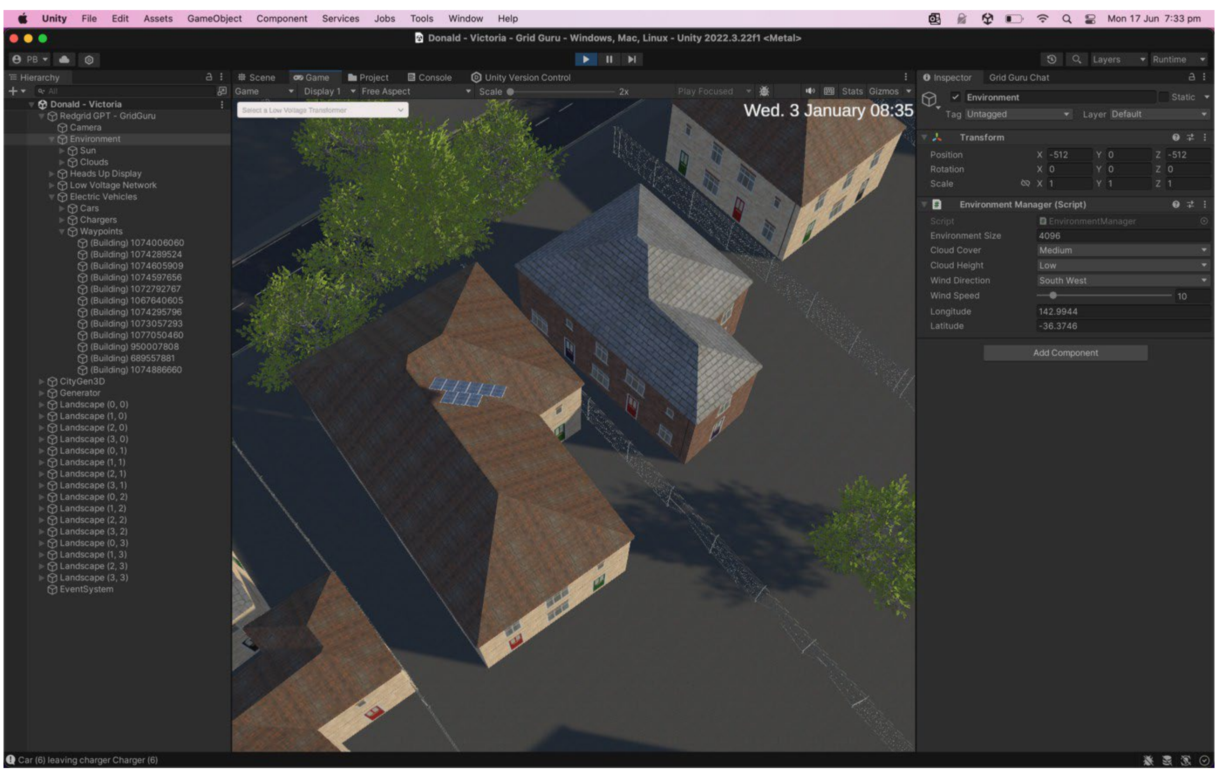

Figure 6: *Screenshot of Solar efficiency modelled with real shadows and cloud cover*

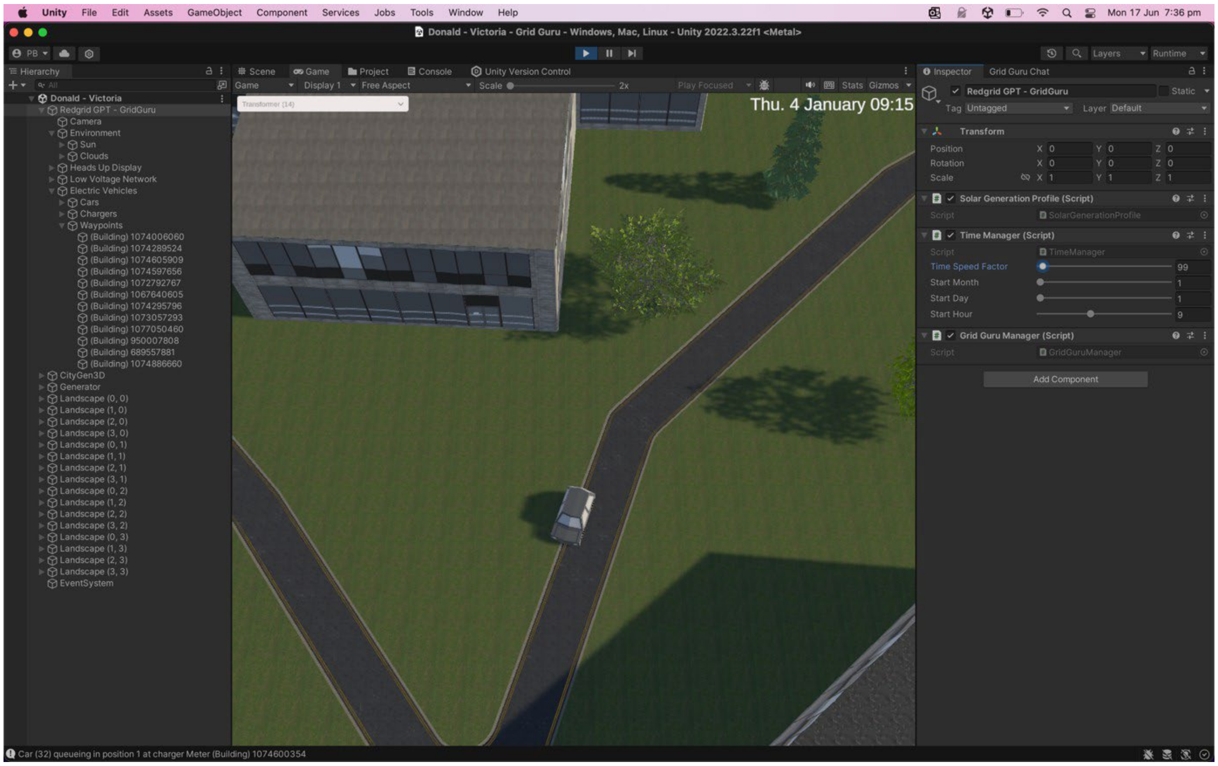

Figure 7: *Screenshot of Run EV Scenarios*

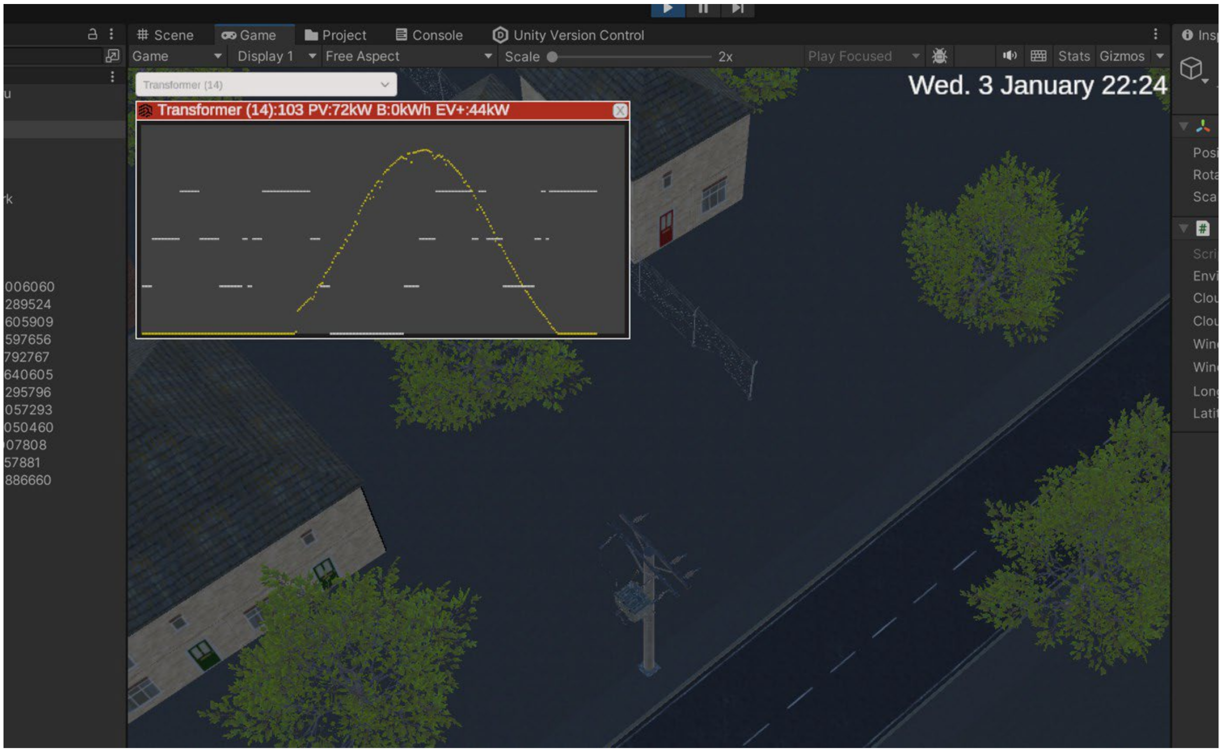

Figure 8: *Screenshot of Solar Generation and EV Charging Plot*

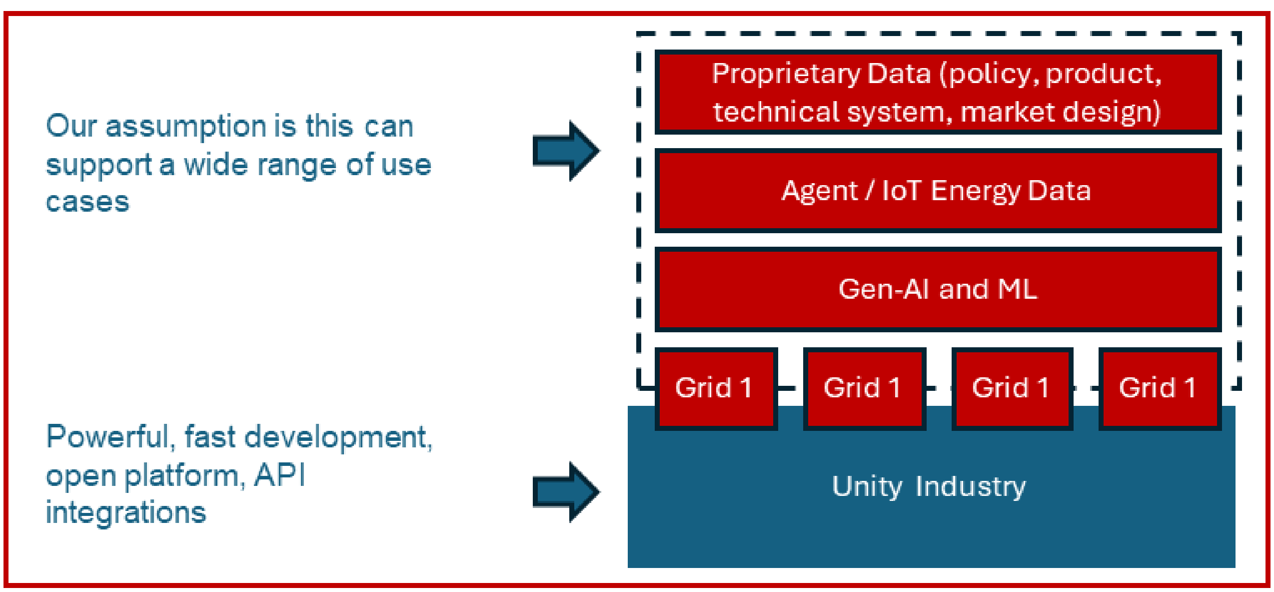

Figure 9: *Grid Guru Value Stack*

3.4 Product Discovery

Ask the energy system

During the feasibility study (detailed in WP3), we built a product demo that could be showcased to the market to test our assumptions and product hypothesis. We interviewed >27 professionals from various sectors within the energy industry. In each interview we presented the product demo and asked:

- What are the top problems the organisation is facing?
- What are their initial thoughts upon seeing the demo?
- Where would it add value?
- What customer and what use case?

Their backgrounds and expertise have provided valuable insights into the potential use cases and value propositions for GridGuru. Here is an overview and summary of the different types of people we've spoken to and the contributions they've made:

Utility Executives and Managers:

- Insights:
  These professionals provided insights into the operational challenges and strategic priorities of utility companies. Their understanding of grid management, data analytics, and network planning highlighted the need for tools that improve network visualisation, expedite customer connections, and enhance data consistency. Their feedback emphasised the importance of real-time grid optimisation and the integration of distributed energy resources (DERs).

Consultants and Strategy Experts:

- Insights
  Their experience in business transformation and energy management solutions underscored the importance of customised, scalable solutions that can adapt to various business needs. They highlighted opportunities for GridGuru to streamline energy monitoring and reporting processes, particularly in light of emerging sustainability standards and regulatory requirements.

Renewable Energy and Policy Specialists:

- Insights:
  These specialists offered perspectives on the integration of renewable energy sources into existing grids and the evolving regulatory landscape. Their insights into energy market design and policy highlighted the potential for GridGuru to support scenario planning, compliance with renewable energy mandates, and the optimisation of renewable energy contributions to the grid.

Finance and Development Advisors:

- Insights

Their expertise in structuring and financing large-scale infrastructure projects provided a financial perspective on the deployment of grid solutions. They emphasised the importance of demonstrating clear ROI and aligning projects with sustainability goals to attract investment.

Technology and Data Experts:

- Contributions:
  These experts provided technical insights into the capabilities and limitations of current grid technologies. Their knowledge of smart grid solutions, advanced metering infrastructure, and data visualisation tools informed discussions on how GridGuru can leverage cutting-edge technologies to enhance grid management and data analytics.

Customer Connections and Request Managers:

- Contributions:

Their focus on managing customer connections and network data requests highlighted inefficiencies in current processes. They emphasised the need for tools that can streamline connection approvals, improve data accuracy, and provide self-service capabilities for customers and community energy groups.

Strategy and Finance Managers:

- Contributions:

Their expertise in renewable energy and forecasting models for distributed energy resources (DERs) provided insights into the financial and strategic considerations for grid upgrades. They highlighted the potential for GridGuru to support cost-benefit analyses, tariff modelling, and microgrid planning.

3.4.1 Insights

1. Flexibility and Extensibility:
   GridGuru is praised for its flexibility and the potential to integrate various data sources, providing comprehensive energy models down to the household level. This capability was valuable for utilities and energy companies looking to optimise network operations, manage distributed energy resources, and plan for future infrastructure needs. Similarly, the ability to perform detailed scenario analyses and forecasting was highlighted as a significant benefit, allowing utilities to plan for different times of the year and respond to dynamic changes in energy consumption and generation.

2. Potential Use Cases:
   Several potential use cases were identified across the discovery calls. These include using the platform for detailed capacity planning, optimising network spending, and managing community energy projects.

3. Strategic Opportunities:

GridGuru's potential to serve various segments within the energy sector was underscored. This includes vertically integrated utilities, standalone power systems (SAPS), and industry bodies focused on energy efficiency and consumer advocacy. The ability to provide high-fidelity planning and monitoring tools was seen as a strong value proposition for organisations managing complex energy systems and those involved in sustainability reporting and compliance with new regulations.

Key Market Segments Identified

1. Utilities (e.g., Ergon, Horizon Power, TasNetworks)
2. Distribution Network Service Providers (DNSPs)
3. Stand-Alone Power Systems (SAPS)
4. Industry Associations and Regulatory Bodies (e.g., AEMO, Grattan Institute)
5. Commercial & Industrial (C&I) Customers (e.g., Beacon Lighting)
6. Government and Public Sector Entities
7. Emerging Markets and International Projects

Limitations

1. **Data Integration and Real-Time Capabilities:**

   Successful implementation of GridGuru's features depends on accessing accurate and comprehensive data from existing systems. The platform should integrate seamlessly with current data sources and avoid duplicating functionalities already provided by other systems.

2. **Market Readiness and Adoption:**

   Despite its technical capabilities, GridGuru's market adoption faces hurdles. The platform's innovative features, such as 3D modelling and advanced scenario analysis, may not immediately resonate with all potential users. Additionally, the conservative nature of the Australian energy market and the challenge of convincing stakeholders to adopt new technologies were identified as potential barriers to widespread implementation.

3. **Competitive Landscape:**

   GridGuru needs to clearly differentiate itself from competitors like Neara and other edge technology platforms. Focusing on unique features such as the generative AI capabilities and the specific use cases where it can add the most value will be crucial.

4. **Collaborative Partnerships:**

   Forming strategic partnerships with utilities and leveraging existing data systems can help GridGuru provide a more comprehensive solution and improve its market positioning.

5. **Scalability and Cost Considerations:**

The scalability of the platform, while seen as a strength, also presents a limitation in terms of cost and resource requirements. Implementing and maintaining such a sophisticated system could be expensive, which might deter smaller utilities or organisations with limited budgets. For instance, the potential cost savings and efficiency gains for companies like Beacon Lighting were highlighted, but the initial investment and ongoing costs could still be a significant consideration.

3.4.2 Identified Use Cases

The feedback from the discovery interviews was synthesised and we have grouped the common recurring use cases and areas of value. These have been scored and prioritised by the level of problem or urgency the organisations were looking to solve.

1. Visualisation and Analysis of Low Voltage (LV) Circuits
    - **Problem:** Difficulty in visualising LV circuits and capacities. It is time consuming to respond to new requests and the current tools and process is outdated.
    - **Use Case:** Implement a tool that provides detailed visualisation and analysis of LV circuits, transformer capacities, and network constraints. This can expedite customer connections, reduce costs, and improve planning for new infrastructure like EV charging stations and community batteries.
    - **Business case:** Reduce the time and effort it takes to calculate and respond to new customer requests. Improve efficiency. Internal teams to focus on higher value tasks.

2. Compliance and Financial Impact Simulations
    - **Problem:** Compliance with new financial and climate-related regulations.
    - **Use Case:** Develop simulation models to assess the financial impacts of climate-related events on the network. This can aid in regulatory compliance and strategic planning.
    - **Business case:** Meet the new regulatory requirements and avoid reputational risk and potential financial penalties. Provides transparency of risk calculation, enhancing trust and confidence among investors, regulators, and customers.

3. Scenario Analysis and Planning for EVs and DER
    - **Problem:** Inability to effectively evaluate and monitor different scenarios to understand and react to network signals and constraints.
    - **Use Case:** Use agent-based analytics and large language models (LLMs) for scenario analysis and management. This would help in stress testing the network, planning for extreme events, and optimising investments in network infrastructure.

- **Business Case:** Optimising DER integration reduces grid instability risks and minimises the need for costly infrastructure upgrades, enhancing the resilience and efficiency of the electricity network.

4. **Community Battery Deployment**
- **Problem:** Determining the optimal locations and configurations for community batteries.
- **Use Case:** Provide a self-service tool that offers detailed insights into network capacities and constraints. This would help community groups and local governments make informed decisions on the deployment of community batteries, potentially as an alternative to traditional infrastructure upgrades.
- **Business Case:** Facilitating community battery deployment through a self-service tool can reduce reliance on traditional infrastructure investments, leading to lower capital expenditures and enhanced local energy resilience.

5. **Broader Sustainability Reporting and Compliance**
- **Problem:** Compliance with the Australian Sustainability Reporting Standards (ASRS).
- **Use Case:** Develop a solution that ingests energy usage and proprietary data to automate the creation of sustainability strategy documents. This would save significant time and resources for companies needing to maintain annual sustainability reports.
- **Business Case:** Automating sustainability reporting can save companies up to 70% of the costs associated with manual report preparation, ensuring compliance and freeing up resources for other strategic initiatives.

6. **Dynamic Operating Envelopes and Tariff Modelling**
- **Problem:** Setting and managing dynamic operating envelopes and tariffs in response to real-time network conditions.
- **Use Case:** Simulate different tariff scenarios and dynamic operating envelopes to manage network constraints and optimise load distribution. This would be particularly useful in areas with high DER penetration.
- **Business Case:** By optimising dynamic operating envelopes and tariff models, utilities can better manage load distribution and maximise revenue while maintaining network stability.

7. **Predictive Maintenance and Asset Management**
- **Problem:** Planning and forecasting maintenance needs and optimising asset management.

- **Use Case:** Use machine learning algorithms to predict future states of network assets based on historical data. This approach can help in proactive maintenance planning, reducing downtime, and extending the life of network assets.
- **Business Case:** Predictive maintenance can decrease unplanned outages and extend asset life, leading to significant cost savings and improved service reliability.

The final deliverable of Work package 1 has provided us with valuable perspectives on the potential use cases and value propositions for GridGuru. We can summarise that the output validated the need for the GridGuru platform, highlighting its significance in addressing current challenges within the energy sector. The feedback helped us understand how we differentiate from the competition by focusing on the low voltage network and leveraging advanced analytics. This distinction enables GridGuru to support a broad range of use cases, from detailed capacity planning to dynamic operating envelopes and predictive maintenance. Additionally, the interviews provided valuable feedback on potential limitations and necessary changes to the demo for better market alignment. Importantly, the sessions helped us shortlist a key customer and identify three top use cases, which will be presented in the next chapter.

4 Work Package 2: Customer and User Validation

Work Package 2 focuses on shortlisting the customer from market research and product discovery and obtaining user validation. The primary goal is to ensure that the proposed system meets the needs of potential users and has a viable market. This involves engaging with potential customers, identifying necessary partnerships, and developing a comprehensive project plan for pilot implementation.

Activities undertaken

- Customer and Use Case Shortlisting:
  A review of potential customers and use cases identified in Work Package 1 was conducted. This helped narrow down the most promising customer segment for further engagement, and to inform the business plan and Phase 2 planning.
- Partnership Identification:
  Essential partnerships with academic and industry stakeholders were identified for a Phase 2 project. This includes collaborations with research institutes like Monash University and potential industry partners who could benefit from or contribute to the "Ask the Energy" system.
- Pilot Project Planning:
  A detailed project plan for the pilot was developed. This plan includes clear objectives, timelines, required funding, and success metrics. The aim is to have a structured and measurable approach to pilot implementation.

4.1 Target customers

Based on the outputs from the product discovery and broader market research we can start to segment the target customers and beneficiaries of GridGuru.

Our initial target customers for a subsequent phase 2 project are:

- Distribution Network Service Providers (DNSPs)
- Vertically Integrated Utilities

By targeting DNSPs and vertically integrated utilities, we aim to leverage their financial strength, data availability, and critical operational needs. Our platform can address their pressing challenges related to low voltage network management, new customer connections, and regulatory compliance. This strategic focus will not only enhance the value proposition of GridGuru but also ensure its successful deployment and adoption in the energy sector.

Specific factors for selection:

Focus on Low Voltage Networks:

- These entities focus on managing low voltage networks, which are becoming increasingly complex and stressed due to the rapid rise in Consumer Energy Resources (CER) and Electric Vehicles (EVs). This segment of the grid requires innovative solutions to maintain stability and efficiency.

Data Utilisation and Support:

- GridGuru is an analytics rich and extensible platform. Private data from DNSPs and Utilities will help build out the capabilities of GridGuru whilst the open-source component of the platform is developed and scaled. If these entities lack specific meter data, we can provide robust representative data to fill in the gaps through other partnerships.

Target problem / product market fit:

- DNSPs and vertically integrated utilities face significant challenges with new customer connections. They need to efficiently identify and manage grid capacity within the low voltage network. The increasing penetration of CER and EVs adds to this complexity, creating a critical need for our platform.

Response to Network Data Requests:

- These entities are frequently required to respond to network data requests from government bodies, industry stakeholders, and consumers. Our platform can streamline this process by providing accurate, real-time data and analytics, improving their response efficiency and accuracy.

Financial Capability:

- Both DNSPs and vertically integrated utilities are well-funded and have resources to commit towards a Phase 2 project, subject to a compelling business case.

4.1.1 Additional beneficiaries / secondary customers

The primary focus of the feasibility study is to prioritise a primary customer and target market. Ongoing discovery is required to map and further explore the beneficiaries of the platform.

- Community groups
- Academics and research groups
- Regulators
- EV charging infrastructure

Partnerships required

The primary beneficiaries that have been identified and will be targeted for Phase 2 of our project are Distribution Network Providers. Specifically, these businesses identified potential benefits and immediate value scenarios in the platform's use for connections and network planning functions.

5 Work Package 3: GridGuru Prototype and Data Forecasting

Work Package 3 focuses on the development of a Minimum Viable Product (MVP) prototype and the integration of advanced data forecasting capabilities, forming the core technological foundation of the GridGuru platform. This phase is crucial as it transforms conceptual designs into tangible, functional components, demonstrating the platform's potential to stakeholders and end-users.

Key activities under this work package included:

1. Design and development of the platform's architecture
2. Implementation of machine learning models for energy forecasting
3. Testing and validation through discovery interviews and demos.

The successful execution of Work Package 3 will provide a solid proof-of-concept, paving the way for subsequent phases and larger-scale deployment. It will highlight the platform's capability to provide accurate, actionable insights into energy management, ultimately contributing to more efficient, sustainable energy systems.

5.1 Design and Development of GridGuru

There are lots of alternatives in the 'digital clone' space, which are almost entirely closed/engineering/enterprise focused.

Our project found through discovery that users saw GridGuru as a complementary capability to these other tools with GridGuru occupying a more integrated, and 'evaluation' function for planners, connections staff and energy project teams.

The open approach to the GridGuru architecture was also identified through customer interviews as a powerful feature in that integrations with other platforms and data sources could be simply integrated.

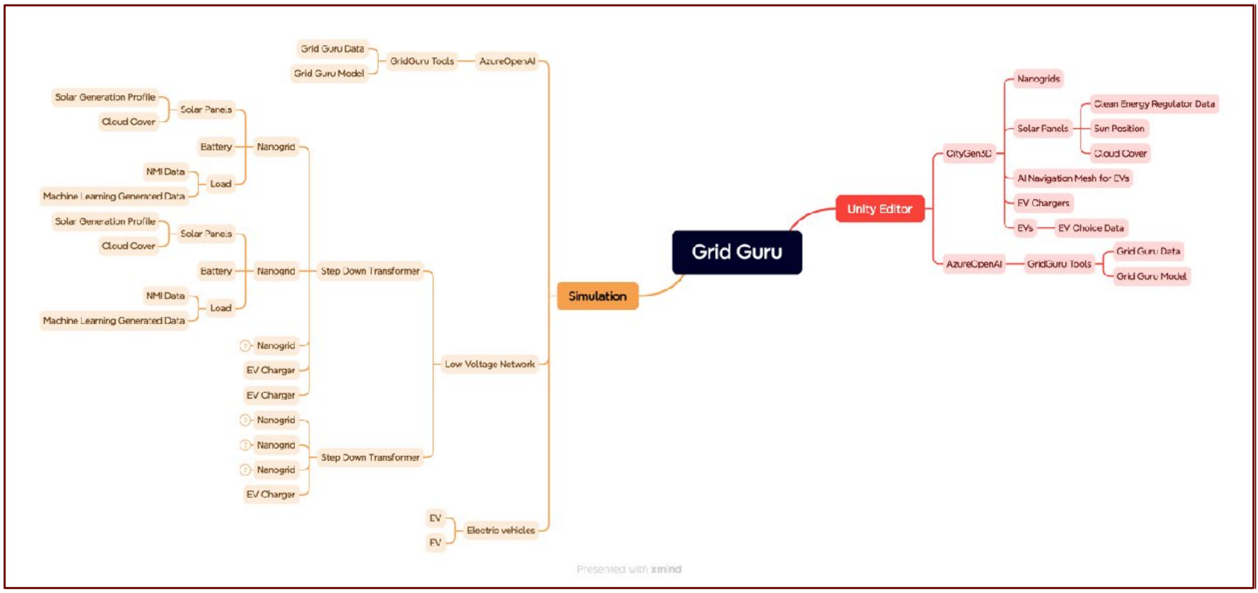

Figure 10: Architecture Overview

5.1.1 Unity

Unity is the world's leading 3D gaming software platform and engine with a variety of tools that can be utilised to meet specific needs of the highly distributed energy ecosystem. The Unity editor is intuitive and customisable allowing an open and extensible freedom to develop workflows and visualisations under an 'agent-based' model.

In terms of modelling, Unity built services start with the asset workflows, then build scenes and finally to publish and share builds.

The GridGuru service also leverages the 'Unity Industry' capability which is an open and extensible industry capability for enabling developers, and engineers to access integrations across and from industries like automotive, manufacturing, government, architecture, energy, and retail and integrate services from them to build and deliver custom real-time 3D experiences for augmented reality (AR), virtual reality (VR), mobile, desktop, and web.

5.1.2 Agent-based approach

Unity and the GridGuru platform have been architected to use what is called and 'Agent-Based Model'.

Under our service any device that consumes/stores/produces and/or monitors energy is defined as an agent.

Under the service agents are individual, sovereign entities that can create and participate in unique trading and energy transaction partnerships and can be configured with unique and customised policies and logic.

Use of this model allows the GridGuru system to be fully adaptable to simulate energy flows and transactions at all levels at the same time and within the same model that exist today and/or may emerge in the future (including p2p, user-retailer, and more).

5.1.3 LLM integration and design considerations

GPT4-turbo has been fine-tuned to detect when one or more functions should be called and respond with the inputs that should be passed to the function(s). In an API call, you can describe functions and have the model intelligently choose to output a JSON object containing arguments to call these functions. The goal of the OpenAI tool's APIs is to return valid and useful function calls more reliably than what can be done using a generic text completion or chat API.

GidGuru has an ever-expanding set of tools that use GPT4-turbo to extract the user's intent and then executes the relevant function. This feature enables GridGuru to present a chat interface for interrogating the data produced by the model, and for making changes to the model both when it is running and when it is in edit mode.

5.1.4 Scalability and extensibility

The LLM features and services are built using Microsoft Azure and Azure AI architecture solution.

These services provide a fully 'elastic' architecture that can be extended and expanded as the data set underpinning the services grows.

Other emerging LLM services, including open-source options such as Meta LLAMA and other alternatives, will be considered for integration in the Phase 2 service to further enhance accessibility of the service.

5.1.5 Capabilities

1. **Agent Based Energy Data**
   For different energy assets across the low voltage network:
   a. Stepdown Transformers, NEMI data, Solar, Battery, EV, and Community Battery

2. **Data Integration**:
   Incorporate real-time data or representative mock data for accurate modelling.
   a. Real-time, historic, forecast.

3. **Low Voltage Network Focused:**

a. Specialised tools for mapping and optimising low voltage networks.
b. Quickly map and model any area using OpenStreetMap data.

4. Scenario Modelling:
Interactive and intuitive scenario modelling enabled by Unity.
a. Configure multiple parameters of the model.
b. Weather, time / day / month, tariff structure, renewable %, DER %.

5. Machine Learning Predictions:
Predict future energy use with advanced machine learning models.
a. 5 min NEMI load forecast.
b. 5 min Solar load forecast.

6. Gen-AI LLM Models:
Query and update the model using advanced Gen-AI large language models.
a. Takes advantage of GPT4-turbo tools to extract the intent of the person chatting to the LLM and execute custom code against the running model.
b. Allows interrogation of the data produced by the model while it is running.
c. Changes to the model can be made using the chat interface.

7. Proprietary Data Integration:
Add company policies, products, and energy regulations to the LLM.
a. Using the Retrieval Augmented Generation (RAG) Pattern, GridGuru adds more insight to the LLM response by searching relevant documents and adding the result to the prompt sent to GPT4-Turbo.

5.2 Machine Learning Model Design and Implementation

In energy forecasting, accurate and reliable models for forecasting load and solar power generation are crucial. Due to the unavailability of direct benchmarks from private energy retail providers —their forecasting models and methodologies are not publicly disclosed —this document recommends the use of established benchmarks and forecast evaluation metrics to assess the performance of our forecasting models. This approach allows for a comprehensive comparison in terms of accuracy, reliability, and practical applicability.

We incorporate baseline methods — naive, seasonal naive, and mean — as foundational comparative tools. These methods often serve as fundamental benchmarks, providing a baseline to determine whether more sophisticated forecasting methods offer a significant improvement (Hyndman and Athanasopoulos 2021). Additionally, we incorporate standard statistical models like ETS (Error, Trend, Seasonality) and ARIMA (Autoregressive Integrated

Moving Average), which are well-established in the field of statistics. They are included as benchmarks to provide a broader context for assessing the accuracy of our forecasting models.

5.3 Data Source

The data originates from the Monash Time Series Forecasting Repository and is specifically the Residential Power and Battery dataset. This open-source dataset is available through Monash University on the Zenodo platform, includes anonymised, minute-by-minute data on load power, solar generation, and battery measurements from real-world customers.
For our data analysis and modelling work, we sample this minute-by-minute data every five minutes, using the resulting five-minute intervals to enhance computational efficiency while maintaining data granularity.

The weather inputs used in the model are sourced from Open-Meteo provides free weather data APIs, offering access to a wide range of meteorological data, including temperature, humidity, precipitation, and solar irradiance, which are essential for our forecasting model.

5.4 Data Cleaning and Pre-processing

Data cleaning and pre-processing were crucial steps to ensure the dataset's integrity and suitability for model training. The process involved several stages to address anomalies, handle missing values, and prepare the data for feature engineering.

The first critical issue we addressed was the occurrence of identical and near-identical solar and load power values over numerous instances. This issue, initially unnoticed, was discovered during an expanded review of the dataset, revealing problematic cases that challenged our initial classification logic. These near-identical values persisted over contiguous periods ranging from several hours to several weeks, posing a significant challenge to data integrity.

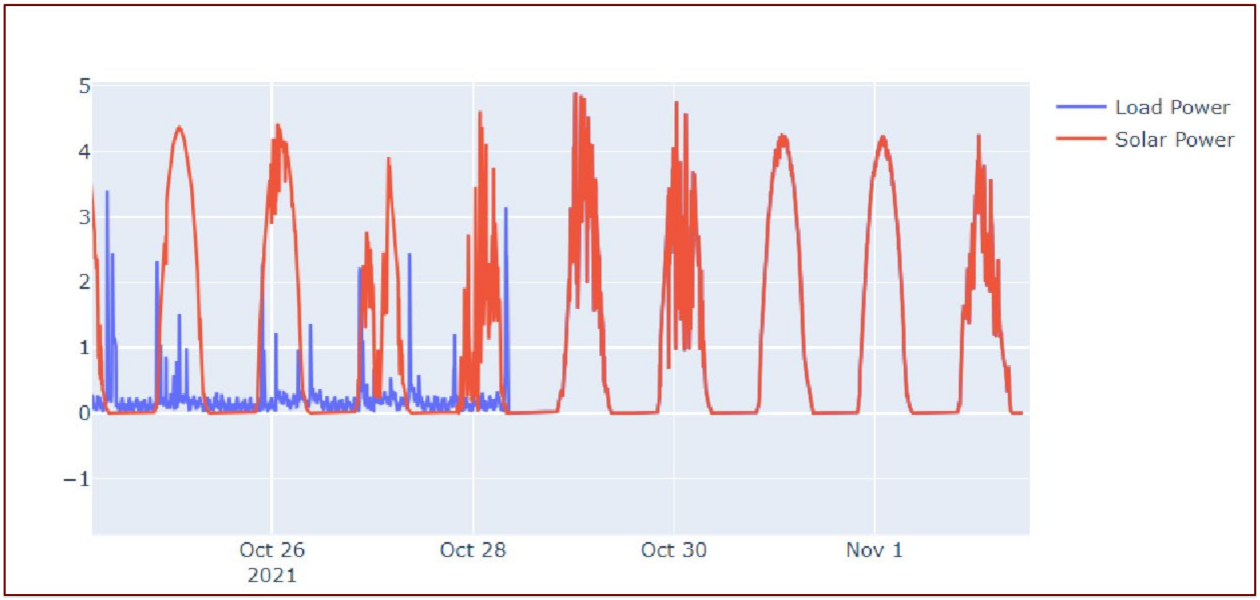

Figure 11: Exclude identical load and solar power from Oct 29th onwards

To resolve this, we developed a machine learning classification model, trained with the CatBoost algorithm, to distinguish between solar and load data even when their values were near-identical. This model was crucial for maintaining data integrity before using it to develop the load and solar forecasting models.

Our pre-processing steps included implementing logic to flag sequences where solar and load values were near-identical and replacing load data with NAs under specific conditions. This approach ensured that the dataset was accurate, reliable, and well-prepared for the subsequent feature engineering and modelling steps.

After addressing this critical issue, we excluded specific households identified as having unreliable or irrelevant data. Additionally, we applied filtering criteria to exclude data outside of defined date ranges for certain households. This ensured that only relevant and reliable data were used for modelling.

We then removed nearly constant sequences of data that lasted for a specified duration, including buffer periods before and after these sequences. Groups with a high proportion of negative values were also filtered out to ensure the data's accuracy.

To further refine the dataset, we identified and removed periods of high load that exceeded a certain threshold for a specified duration, including buffer periods around these high load periods. We applied a buffer around periods with negative values to exclude surrounding data points, ensuring that these anomalies did not affect the model's predictions.

Next, groups with a high proportion of missing values were filtered out, and data were segmented into new unique IDs based on significant time gaps within the original data. We also removed leading and trailing NaN values within each group to ensure that only valid data were included.

We then filled gaps in the data at a specified frequency, ensuring a continuous time series. Finally, we calculated and applied seasonal means to fill remaining NaN values, using both short-term rolling means and long-term time index-based means.

This comprehensive data cleaning and pre-processing approach ensured that the dataset was accurate, reliable, and well-prepared for the subsequent feature engineering and modelling steps.

5.5 Code and Descriptions

1. **Removing Specific IDs:**
   - We excluded data from specific households identified as having unreliable or irrelevant data.

2. **Filtering by Date Ranges:**
   - We applied specific date filters to exlude data outside of defined periods:

```
def filter_by_unique_id_and_date(df, filters):
    for condition in filters:
        unique_id = condition['unique_id']

        if 'start_cutoff_date' in condition:
            start_cutoff_date = pd.Timestamp(condition['start_cutoff_date'])
            df = df[~((df['unique_id'] == unique_id) & (df['ds'] <
start_cutoff_date))]

        if 'end_cutoff_date' in condition:
            end_cutoff_date = pd.Timestamp(condition['end_cutoff_date'])
            df = df[~((df['unique_id'] == unique_id) & (df['ds'] >
end_cutoff_date))]

        if 'start_date' in condition and 'end_date' in condition:
            start_date = pd.Timestamp(condition['start_date'])
```

```
            df = df[~((df['unique_id'] == unique_id) & (df['ds'] >= start_date) &
(df['ds'] <= end_date))]

    return df

# Example
filters = [
    {'unique_id': 18, 'end_cutoff_date': '2022-05-04'},
    {'unique_id': 34, 'start_cutoff_date': '2021-12-03'},
    {'unique_id': 34, 'start_date': '2022-03-15', 'end_date': '2022-05-02'},
    # additional filters...
]
df = filter_by_unique_id_and_date(df, filters)
```

3. **Removing Constant Sequences:**
    - We removed nearly constant sequences of data that lasted for a specified duration, including buffer periods before and after these sequences:

```
def remove_constant_sequences_with_buffer(group, min_length=288, buffer=24,
tolerance=1e-3):
    keep = np.ones(len(group), dtype=bool)
    differences = group['y'].diff().abs() > tolerance
    start_indices = np.where(differences)[0]
    if 0 not in start_indices:
        start_indices = np.insert(start_indices, 0, 0)
    start_indices = np.append(start_indices, len(group))
    for i in range(len(start_indices) - 1):
        start = start_indices[i]
        end = start_indices[i + 1]
        if end - start >= min_length:
            start_buffer = max(start - buffer, 0)
            end_buffer = min(end + buffer, len(group))
            keep[start_buffer:end_buffer] = False
    return group[keep]

df = df.groupby('unique_id',
group_keys=False).apply(remove_constant_sequences_with_buffer)
```

4. **Filtering by Negative Values:**
    - We filtered out groups where the proportion of negative values exceeded a specified threshold:

```
def filter_groups_by_negative_proportion(df, threshold=0.01):
    proportion_negatives = df.groupby('unique_id')['y'].transform(lambda x: (x <
0).mean())
    keep_rows = proportion_negatives <= threshold
    return df[keep_rows]

df = filter_groups_by_negative_proportion(df).reset_index(drop=True)
```

5. **Removing High Load Periods:**
    - We identified and removed periods of high load that exceeded a certain threshold for a specified duration, including buffer periods:

```
def remove_high_load_periods_with_buffer(data, threshold=1.5, duration=24,
buffer=24):
    data['ds'] = pd.to_datetime(data['ds'])
    def process_group(group):
        group_id = group['unique_id'].iloc[0]
        group['high_load'] = group['y'] > threshold
        group['block'] = (group['high_load'] !=
group['high_load'].shift()).cumsum()
        high_load_periods = group[group['high_load']].groupby('block').agg(
            start=('ds', 'min'),
            end=('ds', 'max'),
            duration=('ds', lambda x: (x.max() - x.min()).total_seconds() / 3600)
        )
        high_load_periods = high_load_periods[high_load_periods['duration'] >=
float(duration)]
        periods_to_exclude = [
            (row['start'] - pd.Timedelta(hours=buffer), row['end'] +
pd.Timedelta(hours=buffer))
```

```
            for _, row in high_load_periods.iterrows()
        ]
        for start, end in periods_to_exclude:
            group = group[(group['ds'] < start) | (group['ds'] > end)]
        group = group.drop(columns=['high_load', 'block'])
        return group
    result = data.groupby('unique_id', group_keys=False).apply(process_group)
    return result

df = remove_high_load_periods_with_buffer(df).reset_index(drop=True)
```

6. **Handling Buffering of Negative Values:**
    - We applied a buffer around periods with negative values to exclude surrounding data points:

```
negative_mask = df['y'] < 0

def apply_buffer(series, buffer_size=24):
    if series.any():
        convolution_size = buffer_size * 2 + 1
        kernel = np.ones(convolution_size, dtype=int)
        extended = np.convolve(series, kernel, 'same') > 0
        return extended
    return series.to_numpy()

def buffer_mask(df, unique_id_col, target_col, mask_series, buffer_size=24):
    def reindexed_apply_buffer(group):
        mask_subset = mask_series.reindex(group.index).fillna(False)
        return apply_buffer(mask_subset, buffer_size)
    return
df.groupby(unique_id_col)[target_col].transform(reindexed_apply_buffer)

buffered_mask = buffer_mask(df, 'unique_id', 'y', negative_mask, buffer_size=24)
df.loc[buffered_mask, 'y'] = np.nan
```

7. **Filtering High Missing Data Groups:**
    - We removed groups where the proportion of missing values exceeded a specified threshold:

```
def remove_high_missing(df, id_col='unique_id', target_col='y', threshold=0.25):
    missing_proportion = df.groupby(id_col)[target_col].apply(lambda x:
x.isna().mean())
    valid_ids = missing_proportion[missing_proportion < threshold].index
    filtered_df = df[df[id_col].isin(valid_ids)]
    return filtered_df

df = remove_high_missing(df)
```

8. **Filtering High Missing Data Groups:**
    - We segmented data into new unique IDs based on significant time gaps within the original data:

```
def segment_unique_ids(df, gap_days=7):
    df = df.sort_values(by=['unique_id', 'ds'])
    df['time_diff'] = df.groupby('unique_id')['ds'].diff()
    gap_threshold = pd.Timedelta(days=gap_days)
    df['is_gap'] = df['time_diff'] > gap_threshold
    df['segment'] = df.groupby('unique_id')['is_gap'].cumsum()
    df['new_unique_id'] = df['unique_id'].astype(str) + '-' + (df['segment'] +
1).astype(str)
    df.drop(columns=['time_diff', 'is_gap', 'segment'], inplace=True)
    df.drop(columns=['unique_id'], inplace=True)
    df.rename(columns={'new_unique_id': 'unique_id'}, inplace=True)
    return df

df = segment_unique_ids(df, gap_days=7)
```

9. **Trimming Lead and Trailing NaNs:**

- We removed leading and trailing NaN values within each group:

```
def trim_nans(group):
    non_nan_mask = group['y'].notna()
    if not non_nan_mask.any():
        return group.iloc[0:0]
    first_idx = non_nan_mask.idxmax()
    last_idx = non_nan_mask[::-1].idxmax()
    return group.loc[first_idx:last_idx]

df = df.groupby('unique_id').apply(trim_nans).reset_index(drop=True)
```

10. **Ensuring Minimum Data Span:**

- We filtered out groups that did not meet a minimum required number of days of data:

```
def filter_by_data_span(df, min_days=14):
    date_stats = df.groupby('unique_id')['ds'].agg([min, max])
    date_stats['data_span'] = (date_stats['max'] - date_stats['min']).dt.days + 1
    excluded_ids = date_stats[date_stats['data_span'] < min_days].index
    return df[~df['unique_id'].isin(excluded_ids)]

df = filter_by_data_span(df, min_days=14).reset_index(drop=True)
```

11. **Filling Gaps:**

- We filled gaps in the data at a specified frequency, ensuring a continuous time series:

```
from utilsforecast.preprocessing import fill_gaps
df = fill_gaps(df, freq='5min', end='per_serie')
```

12. **Applying Seasonal Means:**
    - We calculated and applied seasonal means to fill remaining NaN values, using both short-term rolling means and long-term time index-based means:

```
def seasonal_means(group):
    group['time_idx'] = group['ds'].dt.hour * 12 + group['ds'].dt.minute // 5
    seasonal_df = pd.DataFrame()
    for time_idx in range(288):
        temp = group[group['time_idx'] == time_idx]
        rolling_means = temp['y'].rolling(window=8, min_periods=1).mean().shift()
        temp['seasonal_mean'] = rolling_means
        seasonal_df = pd.concat([seasonal_df, temp])
    seasonal_df.sort_values('ds', inplace=True)
    seasonal_df['y'] = seasonal_df['y'].fillna(seasonal_df['seasonal_mean'])
    time_idx_means = seasonal_df.groupby('time_idx')['y'].transform('mean')
    seasonal_df['y'] = seasonal_df['y'].fillna(time_idx_means)
    return seasonal_df.drop(columns=['time_idx', 'seasonal_mean'])


df = df.groupby('unique_id').apply(seasonal_means).reset_index(drop=True)
```

5.6 Model Development

The development of the load power forecasting model followed a structured approach involving data preparation, feature engineering, model selection, and cross-validation to evaluate the performance of the model consistently. We used the MLForecast package from Nixtla and employed LightGBMCV for cross-validation to ensure the model's robustness and accuracy.

Modelling load power data, which exhibits a near-daily seasonal pattern but with some variation in the timing of peaks and troughs within and across households, is a common challenge in time-series forecasting. For the high frequency 5-minute load power data with non-exact seasonal patterns, we chose a modelling approach that can accommodate both the high granularity of the data and slight irregularities in the seasonal cycle. Below are some techniques that were effective for the modelling of load power.

5.6.1 STL Decomposition

STL decomposition is a robust method for decomposing a time series into seasonal, trend, and residual components. We use the MSTL decomposition technique to extract the seasonal and trend components, which are incorporated as features to directly inform the model of the underlying seasonal patterns and trends observed in the historical data.

Using seasonal and trend components as features in the training data and applying cross-validation requires careful consideration to ensure the validity and effectiveness of your cross-validation approach. To prevent leakage and ensure that the model can generalise well, the decomposition should be performed separately within each training fold used during the cross-validation. This way, the seasonal and trend components used in training are never derived from future data points that the model will be validated against. When moving to a new fold, we recompute the seasonal and trend features based on the extended training data set to reflect the most accurate and current patterns.

Machine Learning Models with Exogenous Variables

Advanced machine learning models are particularly useful due to their ability to capture complex patterns and interactions in the data. Gradient Boosting Machines like LightGBM, CatBoost, and XGBoost can handle non-linearities and complex interactions between predictors. They can use time indicators, previous load values, and other features to predict future load.

5.6.2 Feature Engineering

Feature engineering involved creating additional features from the original dataset to enhance the model's predictive power. The specific features created are:

Time-based Features:

- Hour of the Day:

Extracted from the timestamp to capture daily consumption patterns.

- **Day of the Week:**
  Extracted to account for weekly cycles in power usage.
- **Month:**
  Extracted to capture seasonal variations in power consumption.
- **Holiday Indicators:**
  Binary variables will indicate whether a given day is a public holiday, accounting for unusual consumption patterns. We will incorporate this in the next model iteration.

Lag Features:

- **Previous Load Values:**
  Power load values from previous time steps (e.g., 1 hour ago, 24 hours ago) were included to capture temporal dependencies.
- **Rolling Mean:**
  Average power load over a rolling window (e.g., past 7 days) to smooth out short-term fluctuations and highlight longer-term trends.
- **Rolling Standard Deviation:**
  Variability in power load over a rolling window, providing insight into fluctuations over time.
- **Seasonal Rolling Statistics:**
  Rolling mean and standard deviation over longer seasonal windows (e.g., past 7 days) to capture seasonal patterns.

Weather Features:

- **Temperature:**
  Actual temperature data was used as a proxy for weather forecasts due to the greater accessibility of historical actual weather data. Rolling statistics of temperature, such as rolling mean and rolling standard deviation, were included to capture recent trends and variability. In the future, when historical forecast data becomes available, we will replace the actual temperature data with forecasted data and retrain the model.
- **Humidity:**
  Actual humidity data was used as a proxy for weather forecasts because historical actual weather data was more accessible. Rolling statistics of humidity, such as rolling mean and rolling standard deviation, were included to account for trends. We plan to replace the actual humidity data with forecasted data and retrain the model when historical forecast data is available.
- **Precipitation:**
  Actual precipitation data was used as a proxy for weather forecasts due to the availability of historical actual weather data. Rolling statistics of precipitation, such as rolling mean and rolling standard

deviation, were included to capture trends. We will switch to using forecasted precipitation data and retrain the model when historical forecast data becomes accessible.

- Solar Irradiance:

  Solar irradiance data, including cloud cover and various radiation measures (shortwave, direct, diffuse, and terrestrial radiation), were used as they significantly impact solar power generation. Rolling statistics of these measures, such as rolling mean and rolling standard deviation, were included to account for recent trends and variability.

Each feature was carefully engineered to enhance the model's ability to capture intricate patterns in the data, ultimately leading to more accurate and reliable forecasts. This comprehensive feature set forms a robust foundation for the forecasting model, ensuring it is well-equipped to handle the complexities of household power load data.

5.6.3 Model Selection

Several models were considered for the task, including ETS, ARIMA, and various machine learning algorithms like XGBoost. After a comprehensive evaluation, we chose LightGBM due to its efficiency, scalability, and strong performance on tabular data. LightGBM's ability to handle large datasets and its support for various objective functions made it an ideal choice for our forecasting needs.

Training Process

The training process involved multiple steps:

- Data Preparation:

  The dataset was pre-processed to handle missing values, filter irrelevant data, and engineer features as described in the Data Cleaning and Pre-processing section. Missing values were imputed using forward filling, and irrelevant data such as redundant columns were removed.
- Defining Parameters:

  We defined the parameters for the LightGBMCV model, including lags and lag transformations, to capture temporal dependencies and patterns in the data.
- Cross-Validation:

  We used LightGBM with cross-validation to ensure robust performance assessment. This involved splitting the data into training and validation sets across multiple windows, allowing us to evaluate the model's performance over different periods. The parameters for cross-validation included the forecast horizon (h), step size, and the number of windows (n_windows). Specifically, we used a rolling-origin evaluation approach with a forecast horizon of 24 hours, a step size of 1 hour, and 10 windows.

5.6.4 Early Stopping

Early stopping was employed to prevent overfitting and to determine the optimal number of iterations for the model. During cross-validation, the model's performance was evaluated at regular intervals (every 5 iterations), and training was halted if the performance did not improve by 1% over 10 evaluations. This approach ensured that the model did not continue training beyond the point of diminishing returns, thus enhancing its generalisation capability.

5.6.5 Model Validation

The model was validated using root mean squared error (RMSE) as the primary evaluation metric due to its sensitivity to large errors, which is crucial for forecasting applications. We also considered mean absolute error (MAE) to provide additional perspectives on model performance. The cross-validation results provided insights into the model's performance across different time windows, ensuring its robustness and generalisability. The best iteration identified during cross-validation was used to train the final model. Additional out-of-sample validation was performed using a hold-out dataset to further ensure model robustness.

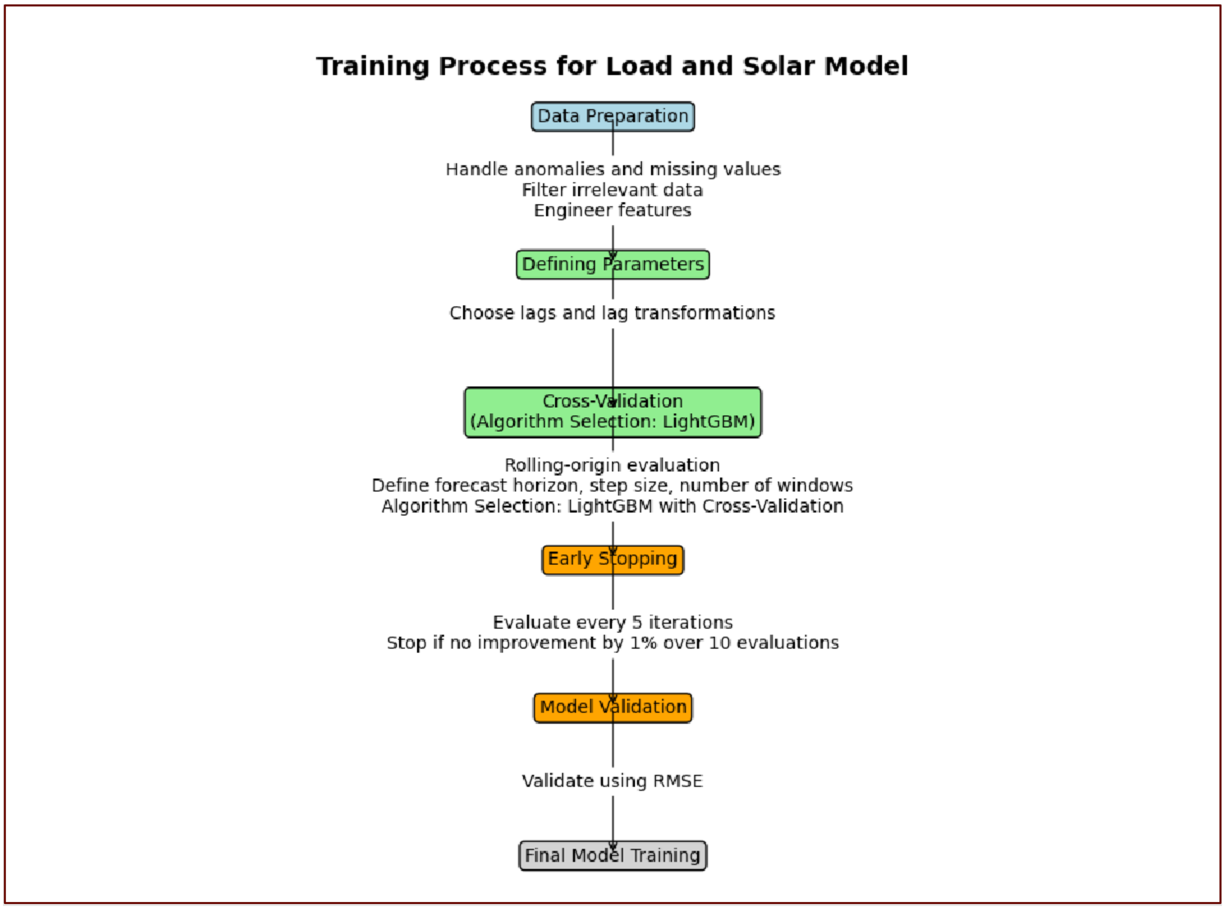

Figure 12: *Training Process for Load and Solar Model*

This flowchart illustrates the sequential steps involved in the training process of the load and solar forecasting model. It begins with data preparation, including handling anomalies, filtering irrelevant data, and engineering features. The next step is defining parameters, followed by cross-validation, which includes algorithm selection (LightGBM) and defining the forecast horizon, step size, and number of windows. Early stopping is employed to evaluate the model every 5 iterations and stop if there is no improvement by 1% over 10 evaluations. The model is then validated using RMSE before final model training.

5.6.6 Benchmarking Process

In assessing our forecasting models, we used a set of established benchmarking models, comparing them with our methods through a structured process. This process included selecting baseline forecasting methods such as Naive, Seasonal Naive, Historical Average, and traditional statistical forecasting methods like Seasonal Exponential Smoothing. We prepared consistent test datasets and applied these models in parallel with our proposed LightGBM model. Using cross-validation, we conducted a comparative analysis of each forecast under the test dataset, employing standard performance evaluation metrics. This approach ensured a robust comparison across different time windows and periods.

5.6.7 Evaluation Metrics

The performance of our forecasting models, as well as the benchmarks, is assessed using the following universally recognised evaluation metrics:

Mean Absolute Error (MAE):
MAE measures the average magnitude of the errors in a set of forecasts. It is calculated as the average of the absolute differences between the forecasted and actual values.

$$\boxed{MAE = \frac{1}{h}\sum_{1}^{h} |y_{T+h} - \hat{y}_{T+h|T}|}$$

Root Mean Square Error (RMSE):
RMSE measures the square root of the average squared differences between forecasted and actual values. RMSE is sensitive to large errors, making it useful for identifying major discrepancies.

$$\boxed{RMSE = \sqrt{\frac{1}{h}\sum_{1}^{h} (y_{T+h} - \hat{y}_{T+h|T})^2}}$$

These metrics are computed for each fold in the cross-validation process, providing a comprehensive view of model performance across different periods. The results from the LightGBM model's cross-validation are compared against those from the Naive, Seasonal Naive, Mean, and Seasonal ETS models to ensure robust benchmarking and evaluation.

5.7 Results

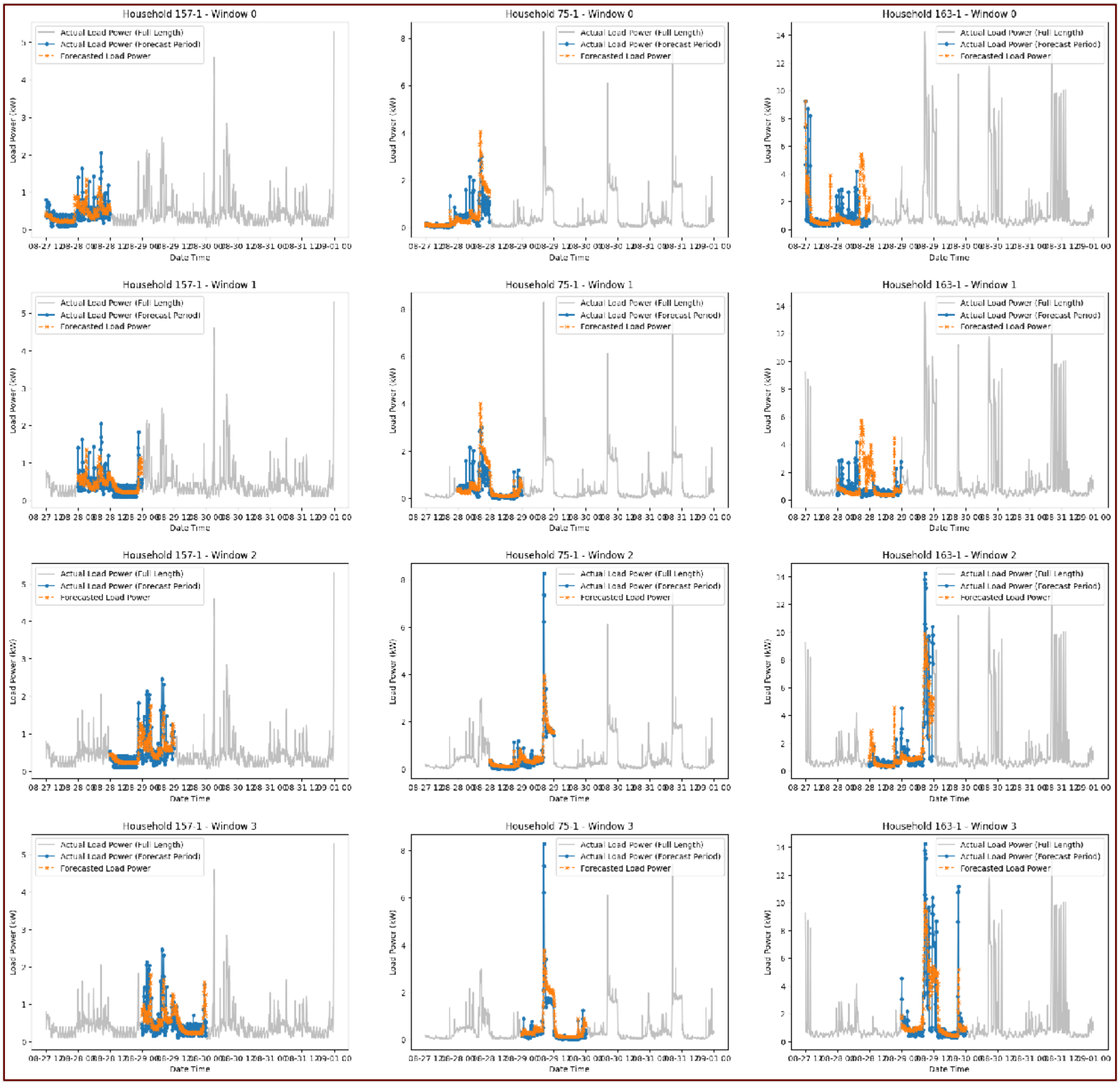

Figure 13: Forecasts of household load power using cross-validation with LightGBM trained models

represents a different household and validation window. The grey line depicts the actual load power over the full length of the time series, the blue dots represent the actual load power during the forecast period, and the orange dots indicate the forecasted load power. The model's performance is illustrated across multiple validation windows for a sample of 3 households.

Model	RMSE	MAE
Naive	1.267114	0.944484
Seasonal Naive	1.117578	0.684848
Historical Average	1.040551	0.773943
Seasonal Exponential Smoothing	0.937039	0.609569
LightGBM	0.662013	0.453449

Table 1: *Cross Validation Performance Metrics for Load Forecasting Models*

The cross-validation results for the load forecasting model, based on 1-day ahead forecasts and a step size of 12 hours, indicate that the LightGBM model outperforms the various statistical methods in terms of both RMSE and MAE. The LightGBM model achieves the lowest RMSE of 0.662 and the lowest MAE of 0.453, suggesting it provides the most accurate and reliable predictions among the models tested.

Among the statistical models, the Seasonal Exponential Smoothing performs the best, with an RMSE of 0.937 and an MAE of 0.610. This indicates that incorporating seasonality into the weighted average improves the forecasting accuracy compared to simpler models such as Naive, Seasonal Naive, and Historical Average.

The Naive model, which serves as a baseline, has the highest RMSE of 1.267 and the highest MAE of 0.944, demonstrating the least accuracy and highest prediction error. This underscores the value of using more sophisticated models that account for seasonality and historical patterns in the data.

Overall, these results highlight the effectiveness of the LightGBM model for load forecasting tasks, providing improvements over traditional statistical models in predicting household load power with reduced error.

5.8 Deployment

Deploying the load power forecasting model to Azure ML Studio involved several critical steps, including local deployment for testing, developing a custom scoring script, and handling challenges specific to the Azure environment. This section provides a detailed account of the deployment process.

5.8.1 Local Deployment for Testing

Before deploying the model to Azure ML Studio, extensive local testing was conducted to ensure the model's functionality and performance. This step was crucial to identify and resolve issues early in the deployment process.

5.8.2 Setting Up the Local Environment

A local environment was configured to mirror the Azure ML environment as closely as possible. This included setting up dependencies and ensuring compatibility with the MLForecast package and LightGBM.

5.8.3 Testing the Model Locally

The model was run locally, generating forecasts and validating results against expected outcomes. This step involved using the same datasets and transformations as planned for the Azure deployment.

5.8.4 Custom Scoring Script

Developing a custom scoring script was a critical part of the deployment process. This script was responsible for handling incoming data, pre-processing it, generating forecasts, and returning the results.

The custom scoring script was designed to pre-process incoming data to match the format used during model training. It also applied the necessary transformations and feature engineering steps before generating the forecast.

Handling large datasets and ensuring consistency in pre-processing steps presented notable challenges when developing the custom scoring script. Efficient handling of large datasets required optimisations in data loading and pre-processing to maintain performance. Matching the pre-processing steps exactly as used during training was challenging, necessitating thorough testing and validation to ensure the script performed accurately and consistently with the trained model.

5.8.5 Creating the Azure ML Pipeline

An Azure ML pipeline was created to streamline the data preparation and model training processes. The pipeline integrates multiple steps including data ingestion, pre-processing, and model training, ensuring a seamless workflow within Azure ML Studio.

The pipeline begins with two primary data sources: a parquet file containing load data (load.parquet) and a CSV file with weather data (open-meteo-34.905138.54E51m.csv). These datasets are processed in the "Data preparation for load power model" step, producing multiple output datasets: transformed, future, training, and testing data. The processed data is then used in the "Train Load Power Model" step to create and validate the forecasting model, leading to the generation of a final model output. The various pipeline endpoints, labelled as pipeline_transformed_data, pipeline_future_data, pipeline_train_data, and pipeline_test_data, represent the different stages of data used in training and evaluating the model.

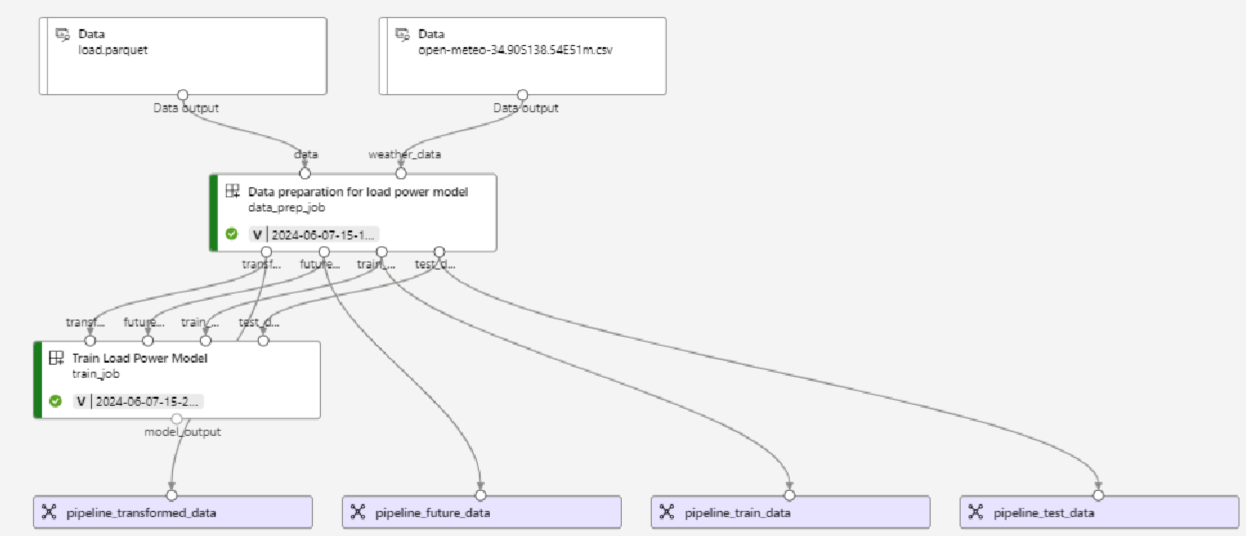

Figure 14: *Azure ML Pipeline for Load Power Forecasting Model.*

5.8.6 Deployment to Azure ML Studio

Deploying the model to Azure ML Studio involved several steps, from setting up the environment to registering and deploying the model.

- Setting Up the Azure Environment:
  A workspace and resource group were created in Azure ML Studio, configuring the necessary compute resources and dependencies.
- Registering the Model:
  The trained model was registered in the Azure ML workspace, making it available for deployment.
- Creating an Endpoint:
  An endpoint for the model was created, configuring it for scaling and setting up security measures.
- Deploying the Model to the Endpoint:
  Once the endpoint was created, the model was deployed with the entry script. Each endpoint can have multiple deployments, and traffic can be directed to these deployments using specified rules.

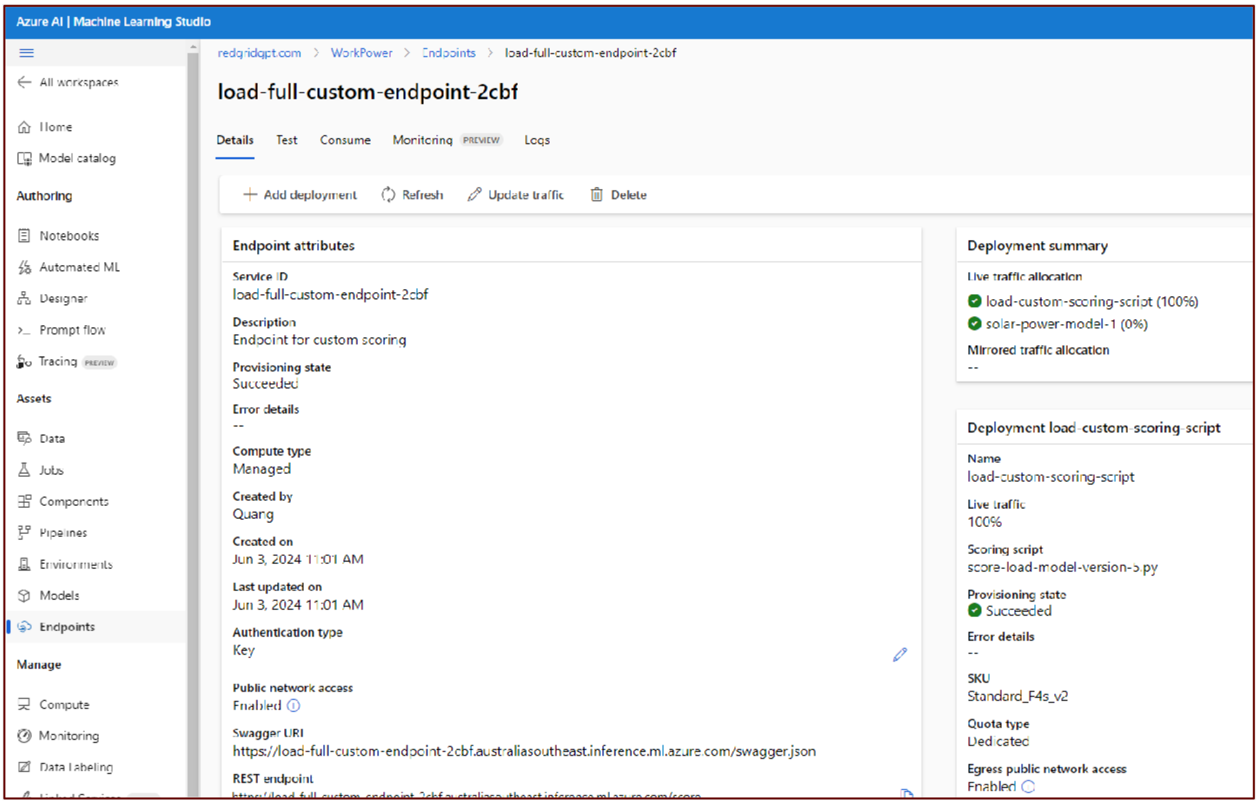

Figure 15: *Azure ML Endpoint for Load and Solar Power Forecasting Model Deployment.*

The Azure ML Studio interface showcasing the endpoint details for the deployment of load and solar power forecasting models. The endpoint is designated for custom scoring. The deployment includes two models: load-custom-scoring-script model, and the secondary solar-power-model-1 model, details of which are not visible in this screenshot but are also part of the deployment.

5.9 Challenges and Solutions

Throughout the development and deployment of the load power forecasting model, we faced several challenges that needed careful consideration and effective solutions. This section outlines the key obstacles encountered and the strategies implemented to overcome them.

5.9.1 Data Preparation Challenges

Handling missing values and outliers in the dataset was a significant challenge. The dataset contained substantial amounts of missing values and outliers, which could negatively impact the model's performance. To address this, we implemented a comprehensive data cleaning and pre-processing pipeline, as detailed in the Data Preparation section. This pipeline effectively handled missing values, removed outliers, and ensured data consistency.

A critical challenge was managing data integrity, particularly due to identical or near-identical load and solar data, which led to numerous large gaps in the dataset. These gaps posed a risk of distorting the model's understanding of typical consumption patterns. To address this, we built a machine learning classification model using the CatBoost algorithm. This model was designed to identify whether the values where load and solar were identical or near identical over contiguous periods were actually load or solar. By segmenting data based on time gaps and employing more stringent filtering criteria, we ensured that only high-quality data was used for model training.

Managing the data volume and processing time was another substantial hurdle. Processing a large dataset efficiently was crucial to avoid excessive computational time. We optimised data pre-processing steps, such as segmenting data based on time gaps and removing unnecessary data points, to reduce the volume and improve processing time. This optimisation was essential to maintain the balance between computational efficiency and data integrity.

5.9.2 Deployment Challenges

Developing and integrating a custom scoring script with Azure ML's deployment framework was one of the most time-consuming tasks. This required extensive debugging and iterative testing to ensure the script performed accurately in the cloud environment. Collaboration with the Nixtla development team was crucial in refining the scoring script, ensuring it functioned correctly and efficiently. The custom scoring script had to handle incoming data, apply necessary pre-processing steps, generate forecasts, and return results seamlessly.

Ensuring that the local testing environment matched the cloud deployment environment to avoid discrepancies was another challenge. We meticulously replicated the cloud environment locally, using the same dependencies and configurations, to ensure consistent performance across both environments. This involved configuring a local environment to mirror Azure ML's setup, including dependencies like the MLForecast package and LightGBM.

The deployment process to Azure ML Studio presented further challenges, particularly related to dependency management and integration with Azure's deployment framework. Ensuring all dependencies were correctly configured in both local and Azure environments required careful management of the conda environment and consistent package versions. The iterative testing process was essential to debug and refine the custom scoring script, ensuring it could handle real-time data and generate forecasts without issues.

Moreover, handling large datasets efficiently during deployment required additional optimisations in data loading and pre-processing. Matching the pre-processing steps exactly as used during training was crucial, necessitating thorough testing and validation. The custom scoring script needed to maintain performance while processing substantial volumes of data, which was addressed by optimising data handling routines and leveraging Azure's robust computational resources.

By addressing these challenges through innovative solutions and extensive testing, we ensured the successful deployment and operation of the load power forecasting model in Azure ML Studio.

5.9.3 Future Improvements

Looking ahead, there are several areas for potential improvement that can enhance the performance and utility of the load and solar power forecasting model.

5.9.4 Model Consumption, Dashboard Development, and Advanced Visualisation

To effectively consume and interact with the forecast data, we plan to develop a comprehensive dashboard that visualises the forecast when a payload of input data is passed into the endpoint. This dashboard will serve as a testbed to observe the model's performance in a simulated real-time environment using historical weather data as inputs. By integrating forecast data into advanced visualisation tools, such as a Unity-based simulation environment, we can provide real-time interactive visualisations. This integration will help users understand and interact with the forecast data in a more intuitive and immersive manner. Eventually, we aim to transition to a real-time production environment that utilises live weather forecast data, enhancing the model's utility and effectiveness.

5.9.5 Hyperparameter Tuning and Modelling Methodology

Hyperparameter tuning will be a focus, as optimising hyperparameters can lead to better model performance. An upgrade in the modelling methodology is also planned. Currently, we use a single model with recursive forecasting. To enhance the accuracy and robustness of our forecasts, we plan to train one model per forecasting step. This direct forecasting approach will eliminate the need for recursive forecasting, reducing error propagation and potentially improving forecast accuracy.

5.9.6 Real-Time Production Environment

Testing the model in a real-time production environment is crucial for validating its performance. This will involve setting up a system to ingest live weather forecast data, as these are key inputs to the model. For now, we can run demos of a simulated real-time environment using historical actual weather data as inputs.

This will help us identify any potential issues and refine the model before deploying it in a fully live environment.

59.7 Model Selection and Clustering

To further enhance the forecasting accuracy, we could compare the error evaluation of each model within each household and select the best model for each household rather than using a single machine learning model for all households. By evaluating the forecasting results on a per-household basis, we can effectively determine and select the most accurate model for each household. In cases where there is limited data, we could use a rolling average window to forecast in the beginning while more historical data is gathered, before switching to the machine learning model for forecasting.

Exploring clustering methods to group households by their historical load power could also be beneficial. By clustering households and building models based on the groups determined by the clustering algorithm, we can create more tailored and accurate models for each cluster, thereby improving the overall forecasting performance.

By addressing these areas, we aim to enhance the robustness, accuracy, and reliability of the load and solar power forecasting model, ensuring it performs effectively in a real-time production environment.

A. Tables and figures

Figures:

Tables:

RACE for 2030

www.racefor2030.com.au

Australian Government
Department of Industry, Science and Resources

Cooperative Research Centres Program