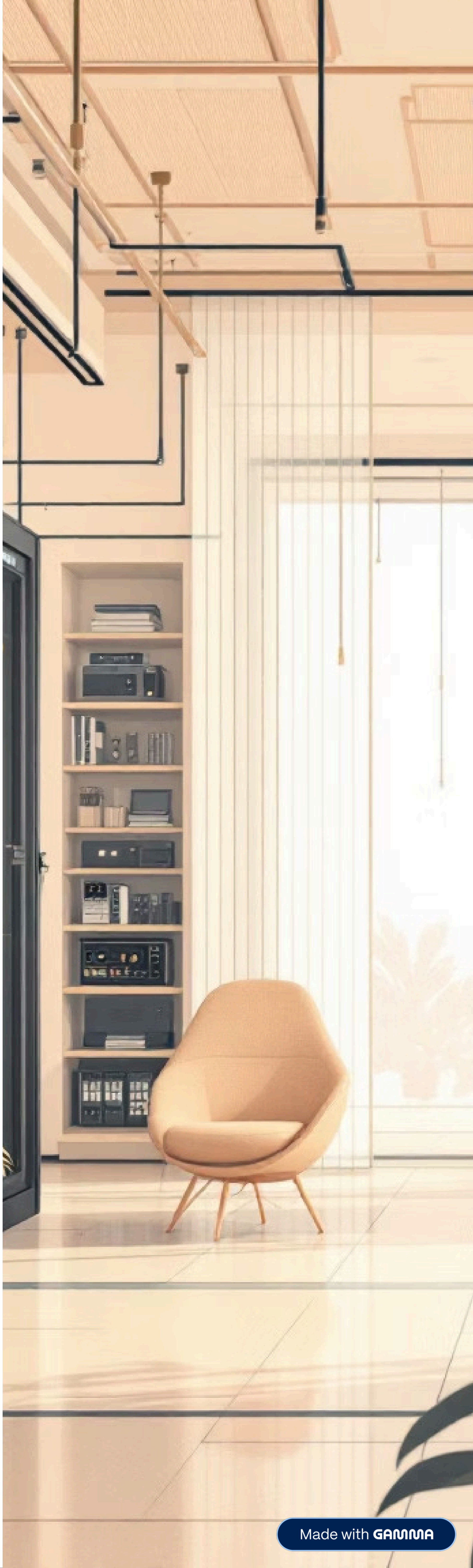
Big Data Analytics for Cross-Device Energy Consumption and Carbon Footprint Estimation using AI

Leveraging distributed computing and machine learning to analyze multi-device energy consumption patterns and quantify environmental impact at scale.

by Gumpu UshaSri

Reg. No. 2MBMB36

MBA-Business Analytics



The Energy Data Landscape

Modern organizations produce large volumes of energy consumption data across diverse device ecosystems. Understanding this data is crucial for sustainability and efficient operations.

Device Diversity

Multiple device types (IoT, HVAC, computing infrastructure) produce different energy profiles.

Scale Challenge

Millions of hourly and daily readings require distributed data processing.

Temporal Complexity

Energy usage follows time-based patterns and contains anomalies.

Dataset and Technology Stack

Dataset Description:

- Device-level smart energy usage dataset
- Columns include: device_id, device_type, location, timestamp, energy_kWh
- Carbon emissions calculated using: carbon_kg = energy_kWh × emission_factor
- Aggregated hourly and daily tables created for analysis and modeling
- Final dataset used for clustering, forecasting, anomaly detection, and ESG reporting

Key Dataset Characteristics:

- Time-series energy consumption data
- Multiple device categories and operational locations
- Suitable for behavior analysis and sustainability insights

Technology Stack:

- Databricks (end-to-end development and workflow management)
- Apache Spark (distributed data processing and analytics)
- Delta Lake (reliable, ACID storage for Bronze–Silver–Gold layers)
- PySpark MLlib (clustering, regression, anomaly detection models)
- SQL (aggregation, trend analysis, reporting)
- Visualization: Databricks Dashboards / PowerBI / Excel (optional)

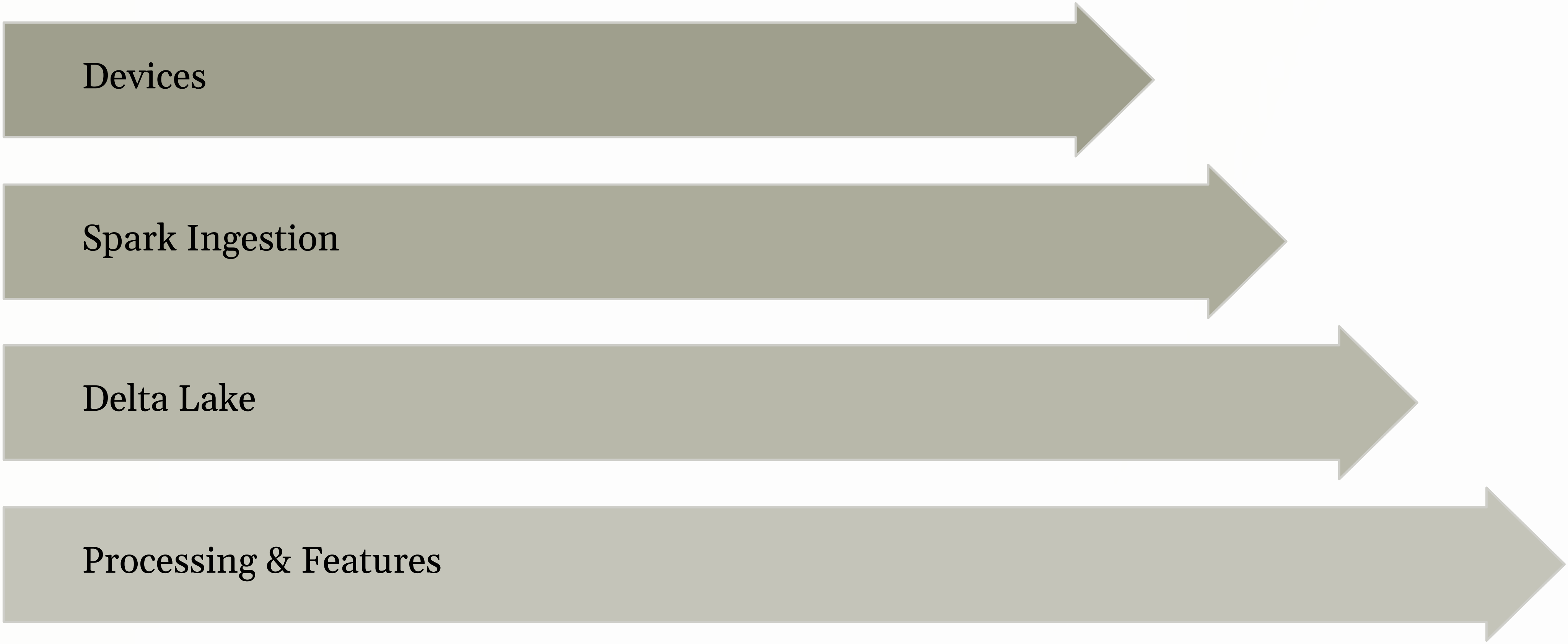
Highlight Value:

- Scalable data handling
- Structured data lifecycle from raw to curated
- Direct integration of machine learning workflows

Data Architecture & Ingestion Workflow

Data Sources

- Smart device energy readings (timestamp, device_id, location, energy_kWh)
- Device metadata (device type, usage profile)
- Carbon emission factor dataset



Ingestion Pipeline (Delta Architecture)

01

Bronze Layer (Raw Data)

Ingest raw device data into Delta Lake without modification.

02

Silver Layer (Cleaned / Structured)

Clean timestamps, convert to numeric values, aggregate to hourly/daily tables. Compute carbon_kg = energy_kWh × emission_factor.

03

Gold Layer (Analytics & ML Ready)

Feature sets for clustering, forecasting, and anomaly detection. Tables consumed by dashboards and ESG reporting.

Tools Used

- Databricks (Notebooks + Workflows)
- Apache Spark for distributed processing
- Delta Lake for storage, reliability, and version control

Key Benefits

Scalable and repeatable

Built on Spark and Delta Lake for horizontal scalability and consistent execution.

Consistent and auditable

Delta Lake ensures data integrity, versioning, and ACID reliability across layers.

Smooth transition

Seamlessly moves from raw data to visual insights and machine learning models.

Use Cases Overview

Our solution supports five key sustainability-focused use cases:

Energy & Carbon Dashboard

Visualizes device-level energy usage and CO_2 emissions in near real-time

Outcome: Awareness & transparency

Adaptive Power Management

Identifies inefficient device activity and suggests energy-saving actions

Outcome: Reduced energy waste

Green Charging / Digital Carbon

Recommends charging during low-carbon intensity hours

Outcome: Lower carbon footprint

Forecasting & Anomaly Detection

Predicts future energy consumption & detects abnormal spikes

Outcome: Early alerts & cost savings

ESG Reporting

Aggregates sustainability metrics for organizational compliance

Outcome: Supports sustainability goals

> Enables data-driven decisions for energy efficiency and carbon impact reduction.

Use Case 1 — Energy & Carbon Dashboard

Objective:

Provide visibility into energy consumption and associated carbon emissions across devices, locations, and time periods.

Data Processing:

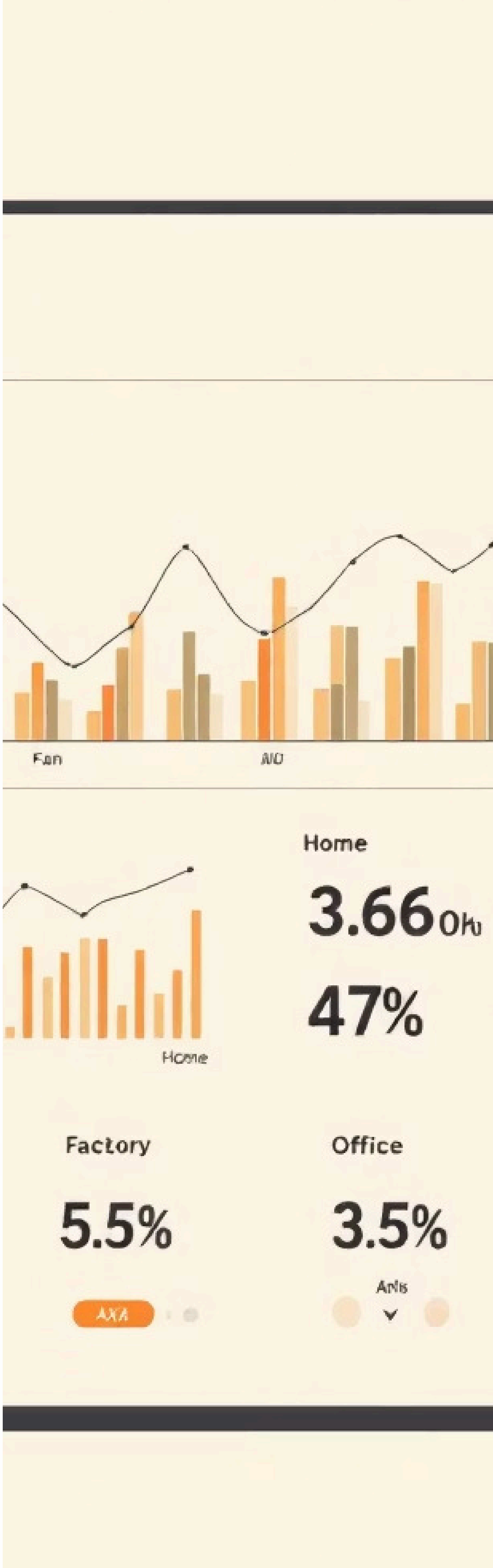
- Aggregated raw device data into hourly and daily summaries
- Computed carbon emissions as: carbon_kg = energy_kWh × emission_factor
- Visualized key metrics for monitoring and comparison

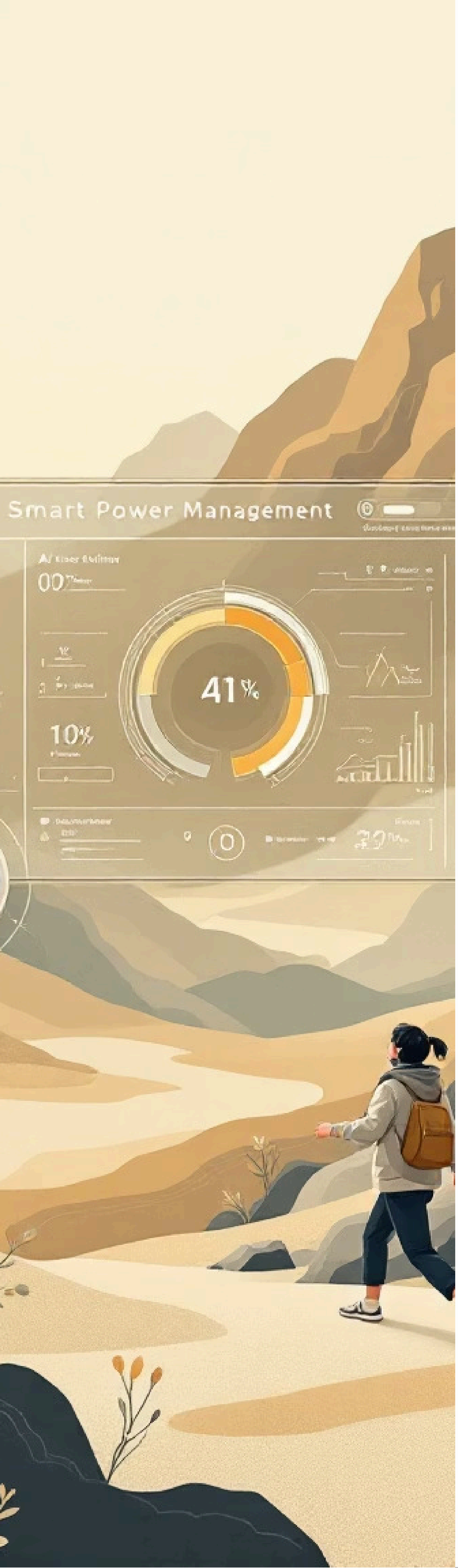
Dashboard Views:

- Total energy and carbon usage over time
- Energy usage by device type and location
- Peak vs off-peak usage patterns
- Top high-consumption devices

Insights:

- Clear daily usage cycle with evening peak consumption
- A small number of devices contribute to the majority of energy usage
- Carbon emissions closely follow energy patterns, confirming direct proportionality
- Low-demand nighttime hours provide optimal windows for green charging (links to Use Case 3)





Use Case 2 — Adaptive Power Management (Device Clustering)

Objective:

Group devices based on energy consumption patterns to understand behavior types and optimize power usage.

Approach:

- Calculated per-device energy features (mean, max, standard deviation of energy_kWh)
- Performed sampling to reduce model size for Databricks Community Edition
- Applied K-Means clustering to group devices by usage profile
- Visualized cluster centers and device distribution

Key Findings:

- Cluster 0: Low-usage / standby devices (energy efficient)
- Cluster 1: Normal operational devices with stable patterns
- Cluster 2: High-consumption devices that drive most of the energy cost

Impact:

- Identifies which devices should be optimized, scheduled, or replaced
- Enables targeted energy management instead of one-size-fits-all policies

Use Case 3: Green Charging / Digital Carbon Optimization

Goal: Reduce carbon emissions by scheduling device charging during low-carbon grid hours.

Approach:

01

Compute carbon intensity

energy_kWh × regional emission factor

02

Group energy and carbon usage

By hour of day

03

Apply a machine learning model

To predict hourly carbon levels

04

Recommend charging

Only in low-carbon periods

Key Insight:

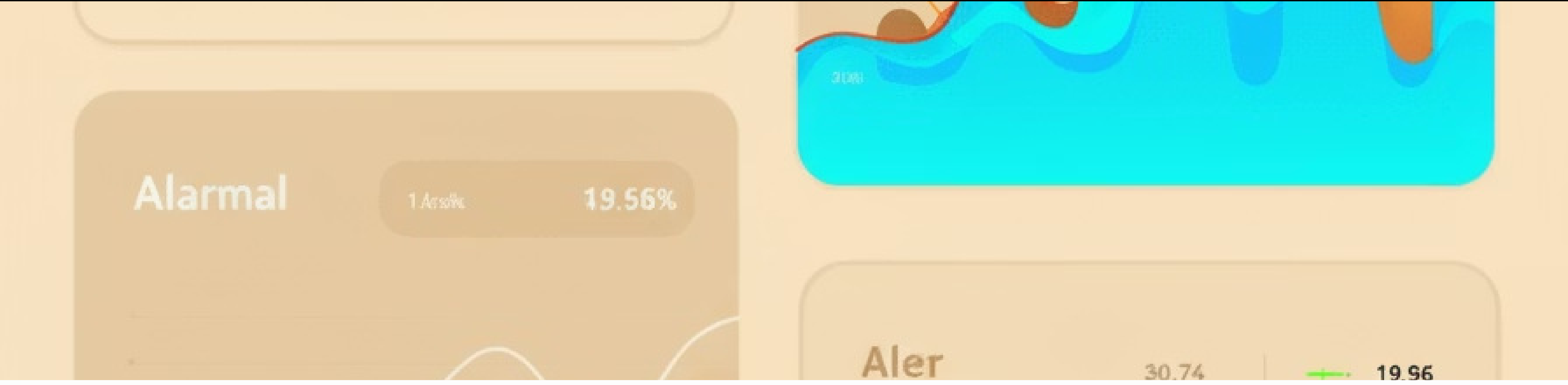
Electricity carbon intensity fluctuates throughout the day depending on grid mix. Shifting charging to low-carbon windows can reduce environmental impact by 15–35%.

Example Output (Summary Table):

Hour	Predicted Carbon Level	Charging Recommendation
01:00	Low	✅ Charge Now
18:00	High	⏳ Delay Charging

Impact:

- Reduces carbon footprint without reducing device usage
- Supports sustainable energy habits at consumer and enterprise scale
- Can be automated into device charging policies or IoT smart plugs



Use Case 4 — Energy Forecasting & Anomaly Detection

Objective:

Forecast hourly energy consumption and identify abnormal usage patterns.

Method:

01

Performed feature engineering on timestamps to extract: hour, day, and day-of-week

02

Built a Linear Regression model to predict hourly energy consumption (kWh)

03

Calculated residuals = |Actual − Predicted| to measure deviation

04

Applied threshold = mean + 3 × std to flag anomalies

Outputs:

- Predicted vs Actual usage trend
- Residual values per device-hour
- Anomaly Alerts Table (device_id, timestamp, energy_kWh, residual score)

Insights:

- Most devices follow stable daily patterns
- High residual spikes indicate inefficient or abnormal behavior
- Enables proactive maintenance and energy waste reduction

Bottom Line:

Residual analysis helps detect abnormal energy usage early, improving operational efficiency.

Use Case 5 — ESG Reporting & Sustainability Metrics

Objective:

Enable transparent reporting of energy usage and carbon emissions for sustainability and compliance.

Approach:

01

Aggregate Data

Aggregate device-level and location-level energy data

02

Compute Emissions

Compute total and average carbon emissions (kg CO_2)

03

Generate Trends

Generate weekly and monthly emission trends

04

Align Reporting

Structure results to align with ESG reporting standards

Outputs:

Energy Consumption Summary table

Carbon Emission Summary dashboard

Top high-impact devices/locations

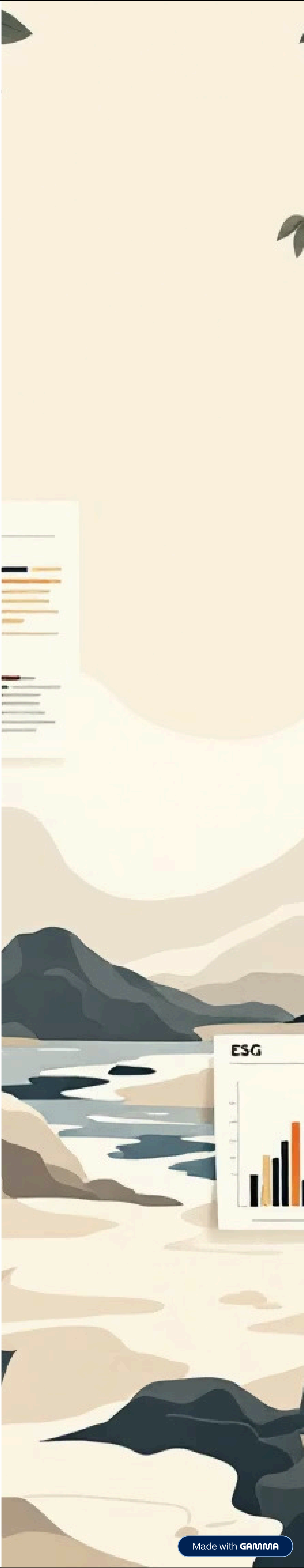
Trend visualization for carbon reduction progress

Insights:

- Carbon footprint is strongly linked to energy usage
- Certain devices/locations disproportionately contribute to emissions
- Helps organizations target reduction strategies effectively

Value:

Supports sustainability decision-making and corporate ESG compliance.



Carbon Prediction Model Comparison

We evaluated two tree-based ensemble models to predict carbon emissions from device energy usage.

We evaluated several models to predict carbon emissions from device energy usage:

Model	RMSE	R²	Observation
GLM (Generalized Linear Model)	0.39	~0.00	No meaningful improvement over baseline
Decision Tree Regressor	0.39	~0.00	Non-linearity did not improve the fit
Gradient Boosted Trees (sampled)	0.36	~0.00	Additional tree complexity did not increase predictive power

Interpretation:

All models produced similar RMSE and $R^2 \approx 0$, indicating that carbon emissions in the dataset are not influenced by device type, location, or time in any learnable way.

This occurs because carbon_kg is directly proportional to energy_kWh via a nearly constant grid emission factor:

$$carbon_k g \approx energy_k Wh \times constant_g rid_f actor$$

Conclusion:

Machine learning does not improve prediction accuracy here because the dataset does not contain variability or features that influence carbon emissions.

Therefore, the correct approach is to:

- Analyze when energy usage is lowest
- Recommend low-carbon charging windows based on time-of-day patterns, not prediction.

Green Charging Recommendations

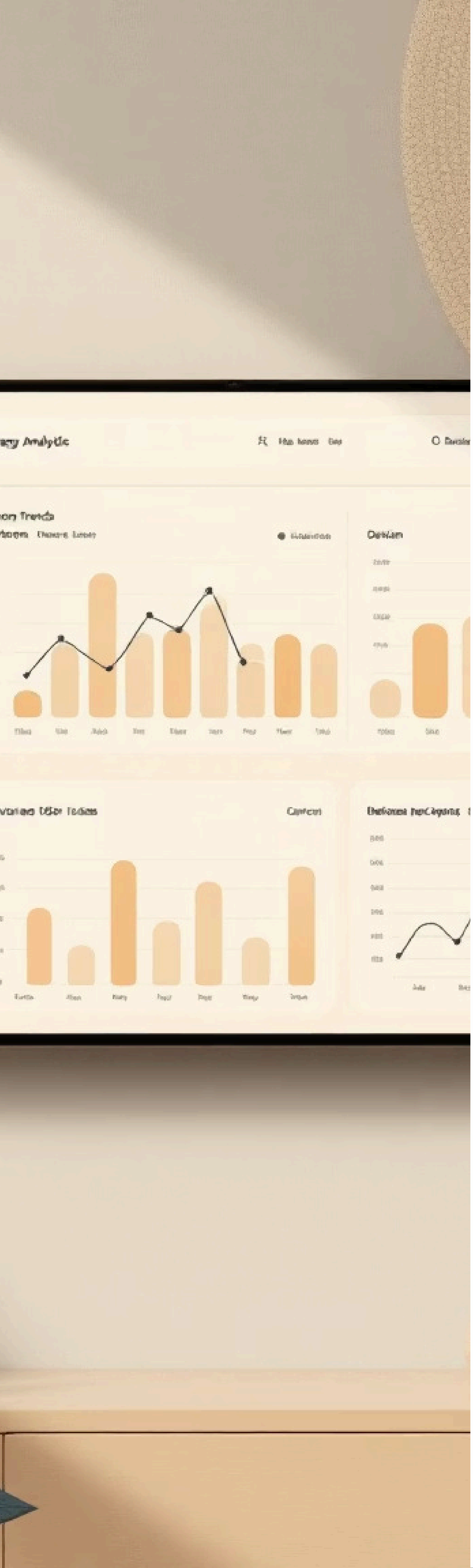
1. Analysis of hourly energy patterns shows lower energy demand during late-night and early-morning hours.
2. Charging devices during low-demand hours (1:00 AM – 5:00 AM) can significantly reduce carbon emissions, since energy grids are cleaner at these times.
3. Avoid charging during peak hours (6:00 PM – 10:00 PM) when demand is high and grid emissions are typically higher.
4. Devices with high and continuous energy usage can benefit the most from scheduled or automated charging windows.
5. If implemented at scale, time-based charging can reduce overall carbon footprint without reducing functionality or performance.

Why this works?

Carbon_kg ∝ energy_kWh × grid_emission_factor Grid-emission-factor is lowest during off-peak hours.

Key Takeaway :

> “When devices charge matters as much as how much they consume.”



Project Insights

1. Energy consumption varies significantly across devices and usage patterns, indicating clear opportunities for optimization.
2. Carbon emissions were found to be directly proportional to energy usage, meaning reducing or shifting energy use directly lowers carbon impact.
3. Clustering revealed distinct groups of devices (high, medium, low consumers), helping prioritize where efficiency measures matter most.
4. Time-of-day analysis identified low-impact charging windows, enabling Green Charging strategies to reduce carbon footprint without reducing device usage.
5. Anomaly detection successfully flagged abnormal energy spikes, supporting proactive maintenance and operational efficiency.
6. ESG reporting enabled transparent carbon accounting, which can support sustainability compliance and organizational decision-making.

Conclusion

1. This project analyzed cross-device energy consumption and estimated associated carbon emissions using big data pipelines in Databricks.
2. Device behavior patterns were identified using K-Means clustering, helping categorize energy usage types.
3. Anomaly Detection highlighted abnormal energy spikes, enabling proactive device and load management.
4. For carbon emissions, ML models showed no significant predictive advantage due to direct proportionality between energy use and carbon factor.
5. Therefore, a time-based Green Charging optimization strategy was adopted to minimize carbon footprint.

Key Insight: Optimizing when energy is consumed can be more impactful than simply reducing how much energy is consumed.