PROJECT PROPOSAL

WattWise-FL: Federated, Privacy-Preserving Energy Forecasts for Smart Buildings

Abstract:

WattWise-FL builds a classical ML pipeline to forecast hourly whole-building energy use while preserving privacy. Using the open BDG2 dataset (2016–2017) with weather and building metadata, we predict meter_reading for electricity, chilled water, hot water, and steam from calendar, building, and weather features plus leakage-safe lags (1/24/168 h) and rolling means (24/168 h). Models are Ridge/Linear, LightGBM, and Explainable Boosting Machine (EBM), trained under per-site, centralized, and federated (FedAvg over model updates—no raw data sharing) regimes. We evaluate with rolling time-series CV and an unseen-site test; RMSLE is primary and MAE secondary. Deliverables include predictions, error diagnostics, EBM shape plots, and optional day-ahead profiles with peak-risk alerts. We do not use any competition/leaderboard dataset; results are offline, reproducible, and aimed at peak shaving, smarter scheduling, and lower emissions.

Motivation:

- **Sustainability impact (CLO 2)**: Enhanced hour-ahead/day-ahead forecasts enable peak shaving, pre-cooling/heating, and smarter schedules lower emissions and bills.
- **Privacy by design**: We emulate federated learning in such a way that all sites can learn from others without sharing raw data—a feasible path for campuses.
- **Course fit (CLO 4)**: We use traditional ML (Ridge, LightGBM, EBM), time-series CV, and careful feature engineering—no deep nets—exactly what this course emphasizes.
- **Open, reusable data**: BDG2 is an open, non-PII weather and building metadata dataset, enabling us to build a reproducible pipeline and share derived artifacts.
- **Explainability for operators**: EBM and partial-dependence plots demonstrate the control that temperature, hour-of-day, and building usage have over load, giving confidence in recommendations.
- **Generalization focus**: We test on unseen locations, which approximates real deployments (a model that is trained on some buildings must perform on new ones).
- **Practical deliverables**: Day-ahead profiles and peak-risk notifications that facilities teams can act on; and a DOI-archived release (code + derived tables).

Literature Survey

Building energy forecasting & datasets.

Open datasets like Building Data Genome 2 (BDG2) provide two years of hourly whole-building meters (electric, chilled/hot water, steam) with building metadata and weather, making them suitable for reproducible studies on non-residential loads. Typical predictive signals are calendar (hour/day/season), weather (temperature, humidity, wind, pressure), and basic building attributes (use type, size, age).

Classical models and evaluation.
Large-scale evidence shows that regularized linear models and gradient-boosted trees (e.g., LightGBM) perform strongly on hourly building loads when paired with leakage-safe lags/rolling windows and careful preprocessing; evaluation commonly uses RMSLE (primary) alongside an absolute-error metric like MAE. To keep results understandable for operators, Explainable Boosting Machines (EBM) provide transparent global "shape" functions (e.g., temperature load) and readable local attributions. We follow this classical, interpretable toolkit—no deep learning.

Privacy-preserving training.
When sites cannot share raw meter logs, Federated Learning (FL) aggregates model updates (FedAvg) from local clients instead of centralizing data, a practical approach even with non-IID building populations. Our project combines these strands: classical models (Ridge/LightGBM/EBM) on BDG2, rigorous time-series CV with an unseen-site test, and a simulated federated setup to study accuracy vs. privacy—without using any competition/leaderboard dataset.

Methodology:

Data Acquisition & Integration
BDG2 hourly meters (2016–2017) + weather + building metadata from the official GitHub repo. Freeze raw snapshots with checksums and UTC retrieval time. Map weather to each site, and join building facts (primary_use, square_feet, year_built).
Pre-processing
Normalize to UTC; enforce hourly regularity and (meter_id, timestamp) uniqueness. Standardize units, remove impossible values (e.g., negative where not allowed). Handle feature missingness via short forward-fill/bounded interpolation with missingness flags. Keep targe
Feature Engineering
Calendar (hour, dayofweek, month, holiday, season, sin/cos hour).
Weather (dry-bulb temp, dew point, wind speed, pressure; optional HDD/CDD).
Building (primary_use one-hot/target-encode, log(square_feet), building age).
History: lags 1/24/168h**,** rolling means 24/168h, rolling std 24h.
Model Building (Classical ML only)
Baselines: "same hour last week", 24h rolling mean, Ridge/Linear.
Main models: LightGBM (GBM) and EBM (Explainable Boosting Machine).
Three regimes: Per-site, Centralized, and Federated (simulated).
Bounded tuning grids; fixed seeds; config-driven training.
Tech Stack
Python (pandas, numpy, scikit-learn, lightgbm, interpret), plot libs (matplotlib), config via YAML, environment lockfile; version control on GitHub.

Deliverables:

Code repository (MIT) — ingestion, cleaning, feature pipeline, time-series CV, classical models (Ridge, LightGBM, EBM), and the federated (FedAvg) loop; README with reproduce steps)
Derived data package (CC BY 4.0) — leak-safe feature tables, split indices, datapackage.json, data dictionary, and provenance log (URLs/DOIs, checksums, retrieval times). (We do not re-host BDG2 raw files.)
Results bundle — metrics tables (RMSLE/MAE overall + per meter type/season/temp band), unseen-site test, calibration plots, and centralized vs per-site vs federated comparison.
Interpretability & diagnostics — EBM global shape plots, LightGBM feature importances/partial dependence, hour x temperature error heatmaps, and peak-risk analysis.
Report & slides — short technical report + presentation deck summarizing methods, results, and sustainability impact.
Data Management Plan (DMP) — final PDF exported from DMPTool, plus SAID certificate.
Archival release — DOI deposit (Zenodo/OSF) containing code, docs, derived tables, model cards, and figures; GitHub release tag.
Dashboard/Visualization : For better understanding

Milestones:
W1–W2 · Data ready (ingestion & QA)
Freeze BDG2 + weather; normalize timezone/units; build processed Parquet; generate validation report; repo/docs skeleton started.
W3–W4 · Features & baselines
Leak-safe features (lags **1/24/168h**, rolling **24/168h**, temp×hour); save split indices; train naïve + **Ridge**; initial metrics.
W5–W6 · Models & explainability
LightGBM (bounded grid) and **EBM**; error slices (season/temp); calibration; EBM global shapes + a few local explanations.
W7–W8 · Federated & robustness
Simulate **FedAvg** (sites as clients) for Ridge/EBM; for trees use ensembling or global-then-local; **unseen-site** holdout (~20%); ablations (no-weather, no-lags); compare regimes.
W9–W12 · Results & delivery
Final metrics tables, calibration plots, error heatmaps, model cards; finalize DMP; write **report + slides**; archive code + derived tables with DOI (Zenodo/OSF); GitHub release tag; include non-competition & sustainability impact notes.
Publication (optional)
Public DOI (Zenodo/OSF) for code + derived data ensures reusability and citation.

Team members and their roles

Arya Mehta — Lead & Data Integration (25%)
Owns DMP updates, provenance (URLs/DOIs, UTC retrieval, checksums), licensing
Sets access & backup policy (encrypted device + SJSU Drive/OneDrive).
Coordinates milestones, merges PRs (after one review), creates GitHub release tag
Prajwal Dambalkar — Data Ingestion & QA (25%)
Pulls BDG2 + weather; freezes raw snapshots; normalizes timezone; enforces)uniqueness.

Builds processed Parquet; generates automated validation_report (schema/units/missingness/outliers)

Keerthika Loganathan — Feature Engineering & Modeling (25%)

Creates leak-safe features (hour/day/month/holidays, weather vars, lags 1/24/168h,rolling
Saves time-series CV split indices; trains baselines (naïve, Ridge) LightGBM EBM.
Implements federated loop (clients = sites; FedAvg for linear/EBM; tree ensembling/fine-tune)

Devanshee Vyas — Documentation, Explainability & Release (25%)

Writes data dictionary + datapackage.json; keeps README and processing.md current.
Produces EBM global shapes, partial-dependence/feature importance, hour×temperature error heatmaps, model cards
Drafts report & slides; assembles the release package with Arya.

Explain how the criteria in the rubric are met.

Abstract (5 pts)

Clear goal (hourly meter_reading forecasting), dataset (BDG2 + weather + metadata), approach (classical ML + simulated federated learning), metrics (RMSLE/MAE), and expected outcomes (day-ahead profile, peak flag, open release).

Motivation (10 pts)

Direct sustainability link: better forecasts enable peak shaving, load shifting, and retrofit targeting, cutting cost and carbon for campuses/non-residential buildings. Also addresses privacy by avoiding raw data sharing.

Literature Survey (10 pts)

Cites building-energy forecasting with boosted trees / regularized linear models, FL (FedAvg) as a training topology, and explainable models (EBM). Positions our work as privacy-preserving, interpretable, and classical-ML focused.

Methodology (10 pts)

Experiment design includes: data freeze + provenance; leak-safe features (lags/rolling), time-series CV (rolling folds), models (Naïve/Ridge LightGBM/EBM), federated vs centralized vs per-site comparison, and evaluation with RMSLE + MAE and seasonal/temperature slices.

Deliverables & Milestones (10 pts)

Week-by-week plan to ship: processed Parquet, feature matrices & split indices, metrics tables, explainability plots, model cards, report & slides, and a Zenodo/OSF DOI release (code + derived/aggregated data). Option to draft a short workshop/ student-venue paper.

Team members & roles (10 pts)

4-way split with backups:
A Lead/Data Steward; B Ingestion & QA; C Features & Modeling; D Docs/Explainability/Release. Responsibilities, RACI, and milestones show uniform workload.

Relevance to the course (10 pts)

No deep learning. Uses classical ML (Ridge, LightGBM, EBM), rigorous validation,

documentation, and communication—exactly within CLO-4. Social-purpose framing satisfies CLO-2.

Technical Difficulty (10 pts)
Non-IID sites, leakage prevention, time-series CV, cross-site generalization, and a practical FL training loop (FedAvg for linear/EBM; documented tree workaround) provide appropriate challenge.

Novelty / Uniqueness (10 pts)
Most student projects centralize data; ours simulates federated learning with explainable classical models, adds unseen-site evaluation, and delivers operational extras (day-ahead profile, peak alert).

Impact (10 pts)
Direct utility for facilities: better operations and emissions reduction. Public, citable release (DOI) + reproducible pipeline reusability and publication potential.
Scale: Buildings use 40% of U.S. energy (and a large share of electricity), so even small forecasting gains translate to big kWh/$ savings and CO_2 cuts across campuses and portfolios.

Heilmeier Catechism — WattWise-FL
What are you trying to do?
Predict each building's hourly energy use (plus day-ahead profile/peak alert) without moving raw

How is it done today, and limits?
Centralized models need data sharing (privacy/legal issues). Per-site models are weak and don't generalize. Many papers use random splits (leakage) and give little explanation for operators.

What's new, and why will it work?
Federated setup with classical ML (Ridge, LightGBM, EBM), strict time-series validation, and explainable effects (EBM). This keeps data local, avoids leakage, and stays lightweight.

Who cares / what difference?
Campus and facility teams, utilities, sustainability staff. Better short-term forecasts peak shaving, load shifting, retrofit planning, lower cost and carbon.

Risks?
Buildings differ (non-IID), tree models don't average neatly, data quality gaps.
Mitigation: FedAvg for Ridge/EBM, tree ensembling/fine-tuning, strong QA, start with electric

How much will it cost?
$0 expected (open tools, SJSU storage);
How long will it take?
12 weeks (W1–2 ingest/QA; W3 dictionary; W4–5 features/baselines; W6–7 LightGBM/EBM; W8 FL; W9 diagnostics + unseen-site; W10–12 package delivery.