National College of Ireland

A Comparative Analysis of Tree-Based, Deep Learning, and Ensemble Models for Building Energy Forecasting

MSc Research Project
MSCAITOPUP

Dan Alexandru Bujoreanu
Student ID: x23309903

School of Computing
National College of Ireland

Supervisor: Faithful Chiagoziem ONWUEGBUCHE

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Dan Alexandru Bujoreanu
Student ID:	x23309903
Programme:	MSCAITOPUP
Year:	2025
Module:	MSc Research Project
Supervisor:	Faithful Chiagoziem ONWUEGBUCHE
Submission Due Date:	11/08/2025
Project Title:	A Comparative Analysis of Tree-Based, Deep Learning, and Ensemble Models for Building Energy Forecasting
Word Count:	5079
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	11th August 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Comparative Analysis of Tree-Based, Deep Learning, and Ensemble Models for Building Energy Forecasting

Dan Alexandru Bujoreanu
x23309903

Abstract

Accurate short-term building energy load forecasting (STBELF) is essential for smart-grid operation, renewable energy sources flexible integration, and carbon-aware building management. This thesis develops a comprehensive STBELF framework, built around a four-stage pipeline that produces over 100 domain specific features, narrowed down to 35 core features/ predictors. On a heterogeneous, real-world dataset of 45 Norwegian public buildings, the study compares classical tree ensembles, deep learning architectures (LSTM, CNN-LSTM, TFT), and stacked ensembles.

Results show that, under rigorous feature engineering, tree-based models—Random Forest (MAE 3.30 kWh, $R^2 = 0.9819$) and XGBoost (MAE 3.42 kWh, $R^2 = 0.9817$)—significantly outperform deep learning models while training orders of magnitude faster. A stacking ensemble delivered competitive performance (MAE 3.58 kWh) but did not surpass the best single tree model. The findings support a *feature-engineering-first* methodology that prioritises data representation over model complexity as a pragmatic, robust strategy for real-world STBELF.

1 Introduction

Short-term building energy load forecasting (STBELF) underpins smart-grid operation (volatility, pricing, demand-response), flexible integration of variable renewables, and carbon-aware building management. In heterogeneous portfolios, forecasting is complicated by diverse building uses, occupancy schedules, control strategies, and weather sensitivity. This section outlines the problem, motivates the comparative study between feature-engineering-centric tree ensembles and deep learning (DL) architectures, and states the research questions, contributions, and limitations.

1.1 Background: forecasting in a volatile energy landscape

The global energy landscape in 2025 is characterized by unprecedented acceleration in demand while racing to net-zero to mitigate climate change. While the growth in global energy consumption moderating slightly in the early 2020s (due to Covid-19), it is projected to rise at an annual rate of 3.4% through 2026 (IEA, 2024). This surge is driven by the global economic expansion, the increasing electrification of transport and heating,

and the exponential growth of energy-intensive artificial intelligence developments and the data centers required to support it.

At the same time, meeting the 2030 and beyond climate targets (European Commission, 2020) requires a shift away from fossil fuels and towards variable renewable energy sources (VRES), like wind and solar power. Subsequently, countries grow increasingly reliant on these intermittent sources, as exemplified by Ireland, which achieved a record 54.5% of its electricity from renewables in February 2025 (EirGrid, 2025), with July 2025 the first coal-free month (EirGrid, 2025).

However, the integration of VRES introduces significant volatility and operational challenges into power grids. Periods of low renewable output during cold winter days (known as "Dunkelflaute" - dark wind lull) or in the hot summer evenings (known as "Hitzeflaute" - hot lull) (Jomaux, 2025a), can stress grid capacity and trigger sharp price spikes. Conversely, periods of excess generation can lead to negative electricity prices, forcing grid operators to implement costly curtailment measures as they race to invest heavily in Battery Energy Storage Systems (BESS) to better service peak demand periods - the "Duck curve" phenomenon explains these challenges (Jomaux, 2025b). Hence the ability to accurately predict near-term energy consumption is important for maintaining grid stability and optimising operational costs - ultimately, improvements in forecast accuracy will allow utilities to reduce reliance on expensive intraday market adjustments (Gandhi et al., 2024).

1.2 Problem Context: flexibility, heterogeneity, electrification

The importance of STBELF extends beyond grid balancing, market dynamics and climate transition - it is the lever of demand-side flexibility. By reliably predicting consumption patterns, building managers, grid operators and energy providers can proactively shift loads and optimise the dispatch of BESS as part of the fast growing "Behind the Meter" (BTM) technologies (Chesbrough, 2020).

Buildings are pivotal for achieving energy flexibility, as they account for a substantial share of total energy demand. In Ireland, for instance, this sector represented 41% of final energy use in 2022 (IEA, 2023a). A key challenge however, is the electrification of heat, especially as countries historically reliant on fossil fuels - such as Ireland, where 53% of heating came from these sources in 2021 - rapidly adopt electric heat pumps (IEA, 2023a). This transition fundamentally alters building load profiles and increases their weather sensitivity, demanding power systems operators to deploy more sophisticated forecasting techniques.

On the other hand, Norway provides a successful case study for this transition. With 81% of heating being electrified and 89% of its electricity generated by hydropower (IEA, 2023b), Norway's energy infrastructure represents a future state for grids transitioning away from fossil fuels. Analysing consumption patterns in a mature, electrified market like Norway provides critical insights for grids preparing for this transition, especially those managing the concurrent demands of large energy users like data centers.

Subsequently, this thesis utilises the real-world COFACTOR dataset (Lien et al., 2025), comprising hourly energy consumption and weather data from 45 public buildings in Drammen, Norway (2018-2022). The portfolio is intentionally heterogeneous (Table 1), making single-model generalisation non-trivial and motivating a careful comparison of modelling paradigms.

Furthermore, the machine learning landscape for STBELF is characterised by a fun-

Table 1: Diversity of the COFACTOR Drammen Building Portfolio.

Characteristic	**Range / Composition**
Building Types	20 Kindergartens, 16 Schools, 7 Nursing Homes, 2 Offices
Scale (Floor Area)	188 m^2 to 8,513 m^2
Age (Construction Year)	1800 to 2015
Occupancy (Users)	16 to over 800

damental tension between two competing paradigms. The choice highlights the trade-off between the significant domain-aware human effort required for manual feature engineering in efficient tree-based models (such as XGBoost and Random Forest) that excel on tabular data, and the high computational cost and complexity of Deep Learning (DL) architectures (like LSTMs and Transformers), which promise automatic feature learning from raw sequences (Kim and Cho, 2019; Grinsztajn et al., 2022).

1.3 Research questions and objectives

This study addresses the current research gap by reviewing a set of research questions (RQ) that support the following primary hypothesis:

A systematic and intensive domain-specific feature engineering pipeline can enable computationally efficient tree ensembles to match or exceed the performance of complex, resource-intensive DL models for multi-building STBELF.

- **RQ1**: Can a comprehensive feature-engineering pipeline enable Random Forest, XGBoost, and LightGBM to outperform LSTM, CNN–LSTM, and TFT for multi-building hourly load forecasting?

- **RQ2**: Do stacking or weighted ensembles deliver material accuracy gains over the best single model, and at what computational cost?

- **RQ3**: Which engineered features (lags, rolling statistics, cyclical encodings, weather interactions) contribute most to accuracy across an electrified public-building portfolio?

The objectives of this research were: (i) to design a leakage-safe, multi-building feature-engineering pipeline (100+ candidates distilled to 35 predictors); (ii) to implement a unified evaluation across tree-based, DL, and ensemble models; and (iii) to quantify accuracy–compute trade-offs for practitioner deployment.

1.4 Contributions and Limitations

This thesis makes three primary contributions to energy load forecasting. First, it establishes a rigorous and transparent comparative benchmark by developing a comprehensive feature engineering first pipeline and evaluating leading tree-based, deep learning, and ensemble models on the COFACTOR dataset. The central contribution, however, is providing clear empirical evidence on the trade-off between feature engineering and model complexity. The findings demonstrate that sophisticated domain-aware feature engineering paired with efficient tree-based models, such as Random Forest (MAE 3.30

kWh), offers a superior balance of accuracy and practicality that significantly outperforms complex deep learning architectures, which are far less efficient, as highlighted by the training time of XGBoost (2.62 seconds) compared to an LSTM (3.75 hours).

Limitations include external validity (Nordic, heating-dominated context) and conservative tuning for DL models to respect compute budgets; stronger tuning may narrow gaps. For ensembling, the meta-learner used a fixed validation set (rather than out-of-fold) due to the long training times of TFT and sequence models (see §4.2.4). A minor implementation detail in early notebooks used non-standard cyclic denominators (23/6); the final analyses retained raw `DayOfWeek` and strong lag/ rolling features, and the codebase now uses standard 24/7 encodings (discussed in §6).

1.5 Thesis outline

Section 2 reviews the evolution of methods and synthesises research gaps and section 3 details the dataset and the feature-engineering pipeline. Section 4 specifies the experimental design and model configurations, while section 5 presents results, compute analyses, and interpretability. And lastly, section 6 discusses implications, limitations, and future work.

2 Related Work

This section reviews the evolution of short-term building energy load forecasting (STBELF) from statistical and time-series baselines to tree-based ensembles, sequence-learning with deep neural networks, and ensembling. The focus is on what has worked in practice, why, and where the literature still falls short.

2.1 From statistical baselines to early machine learning

Historically, forecasting in buildings evolved from physics-based *white-box* models to data-driven *black-box* approaches. Early data-driven work relied on Multiple Linear Regression (MLR) and Autoregressive Integrated Moving Average ARIMA (including its seasonal variant SARIMA), which performed well where seasonality was strong and responses to temperature were near-linear; accuracy improved when social rhythms (weekday/holiday) were encoded explicitly (Fang and Lahdelma, 2016; Hyndman and Athanasopoulos, 2018). As sub-hourly metering, public weather APIs, and shareable datasets became common, the problem shifted from pure time-series analysis to supervised learning. Systematic reviews identified Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and Decision Trees as the most frequently applied advanced models (Yildiz et al., 2017; Amasyali and El-Gohary, 2018; Wei et al., 2018; Zhao and Magoulès, 2012; Deb et al., 2017). These same reviews cautioned that many studies were single-building case reports with heterogeneous splits and metrics, and limited treatment of weather-forecast uncertainty, reducing external validity (Hirvonen, 2024). This motivated more standardised comparisons on multi-building data.

2.2 Tree-based ensembles and feature engineering

The ASHRAE Great Energy Predictor III competition, featuring data from more than 1400 buildings, provided a compelling insight: Gradient Boosted Decision Trees (GB-

DTs) were difficult to outperform (LightGBM/XGBoost ensembles in particular), with participants overwhelmingly reporting that feature engineering and careful preprocessing were the most critical aspects of their approach (Miller et al., 2022). This aligns with the inductive biases of tree-based ensembles (Chen and Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018) and with broad tabular benchmarks (Shwartz-Ziv and Armon, 2022; Grinsztajn et al., 2022). GBDTs learn by additively combining many shallow trees, which matches building physics and operations: heating/cooling often trigger at thresholds, and occupancy schedules create abrupt changes. With high-quality inputs, these models capture such piecewise structure and interactions efficiently. Accuracy consistently improves with targeted feature engineering—lagged loads, rolling statistics, and cyclical calendars (Kuhn and Johnson, 2020). When very long temporal context or multi-step dependencies dominate, explicit features or hybrids may still be required; CatBoost's categorical handling helps with building IDs and holiday types, but extrapolation remains local. These characteristics explain why trees remain strong baselines for building portfolios once inputs are well crafted.

2.3 Deep Learning for sequence modeling: LSTM, CNN-LSTM, TFT

Deep Learning models aim to learn temporal dependencies end-to-end, potentially removing the need for extensive manual feature engineering. Long Short-Term Memory (LSTM) and CNN-LSTM have delivered strong short-horizon performance (1–6 h ahead), robustness to noisy exogenous inputs (i.e. weather), and the ability to model nonlinear dynamics, ultimately capturing dependencies over time (Rafi et al., 2021; Wang et al., 2020) - such models thrive with abundant building-specific history, highly non-linear environments (such as complex HVAC control), exogenous forecasts with uncertainty and multi-building data. However, LSTM performance degraded significantly for longer 24-hour forecasts, where immediate sequential context is less relevant than broader cyclical patterns that are more easily captured by engineered features.

More recently, attention-based models like the Temporal Fusion Transformer (TFT) have been adapted for time-series forecasting (Lim et al., 2021). TFT represents the state-of-the-art for multi-horizon forecasting, designed to ingest a complex mix of static metadata, known future inputs, and past time-series. A direct application to building systems highlighted that a Transformer-based model achieved high accuracy and stability for predicting cooling loads (Li et al., 2023).

The literature suggests that DL gains tend to materialise under specific conditions: the availability of very large datasets, the presence of highly complex and non-linear system dynamics, and the need to generate multi-horizon forecasts from a single model. Conversely, when features are well-engineered and computational resources are a consideration, DL models may not surpass strong GBDT baselines (Wang et al., 2020).

2.4 Summary and research gaps

Cross-domain benchmarks consistently find that well-tuned tree ensembles outperform specialised DLs on typical tabular tasks, often by large compute margins (Shwartz-Ziv and Armon, 2022; Grinsztajn et al., 2022; Borisov et al., 2024). Building-level load data, once augmented with seasonal, weather, and lagged/recency features, is effectively tabular. Hence, claims of universal DL superiority are not supported by current evidence (including

this study). At the same time, the building literature still includes heterogeneous setups (datasets, splits, horizons, metrics), with mixed generalisable conclusions (Amasyali and El-Gohary, 2018; Wei et al., 2018; Deb et al., 2017; Hong and Fan, 2016).

Table 2: What works when in STBELF.

Model family	**Best-fit data conditions**	**Typical horizon**	**Compute / ops cost**
Linear / SARIMA	High seasonality; near-linear temp response; limited history.	1–24h	Minimal; easy upkeep.
Random Forest	Heterogeneous buildings; noisy data; decent with modest features.	1–24h	Low–moderate train; fast predict.
GBDTs	Engineered lags/rolls; seasonality; strong accuracy-per-compute.	1–48h	Low train time; very fast inference.
LSTM / CNN–LSTM	Short horizons; complex non-linear patterns; noisy exogenous forecasts where sequence context helps.	1–6h	Higher train time; moderate inference.
TFT / Transformer	Many related series; multi-horizon outputs; attention/variable selection; cross-entity learning.	1–24h (multi)	High train/infer; careful tuning.
Stacking (meta-learners)	Diverse strong bases (GBDTs + DL) with leakage-safe meta-features; small but reliable gains.	Any	Low–moderate meta cost; higher ops complexity.

In response, this thesis was designed to address six specific gaps:

Gap 1: Limited scale and external validity. Much of the prior work evaluated a few models on a single building, with incomparable setups, limiting generalisability (Amasyali and El-Gohary, 2018; Deb et al., 2017; Wei et al., 2018). *This thesis* used a heterogeneous portfolio of 45 buildings and a consistent, leakage-safe walk-forward protocol with common metrics (MAE, RMSE, CV(RMSE), R^2).

Gap 2: Conflation of model and feature effects. Comparisons often pit a feature-rich GBT against a DL model trained on raw/ minimally-engineered inputs, making it unclear whether differences stem from architecture or representation. *This thesis* adopted a *feature-engineering-first* pipeline (calendar cycles, lags at $t-1$, $t-24$, $t-168$, rolling statistics, weather recency) and evaluated *all* model families on the *same* engineered inputs to isolate inductive biases.

Gap 3: Missing computational budget and efficiency reporting. Accuracy is frequently reported without training/inference cost, despite clear evidence that trees deliver superior accuracy-per-compute on tabular data (Shwartz-Ziv and Armon, 2022;

Grinsztajn et al., 2022). *This thesis* reported wall-clock training times and basic model footprints to make cost–accuracy trade-offs explicit.

Gap 4: Robustness under uncertainty and regime shifts. Few studies examine degradation under forecasted weather or across seasons/ occupancy regimes, though horizon-dependent error growth is well known (Wang et al., 2020). *This thesis* evaluated across seasons and horizons and discussed implications under weather-forecast uncertainty.

Gap 5: Interpretability and trust. Black-box perceptions hinder adoption; XAI remains under-used in building operations (Lundberg and Lee, 2017; Machlev et al., 2022). *This thesis* prioritised interpretable tree baselines (feature importance/ SHAP) to support diagnostics and handover.

Gap 6: Lack of unified, compute-aware ensemble baselines. Ensembling can improve robustness, yet leakage-safe stacking across diverse base learners is rarely reported. *This thesis* implemented simple validation-set stacking (ridge/LightGBM meta-learners) under identical inputs and quantified marginal gains versus cost.

3 Research Methodology

This section describes the dataset, preprocessing, and the *feature-engineering-first* pipeline that underpins all models in this thesis. The goal was a leakage-safe, reproducible framework in which competing model families are tested under identical inputs, in order for differences to be be attributed to architecture rather than data handling (see also §2.4).

3.1 Data acquisition and description

This study used the COFACTOR Drammen dataset of hourly energy consumption for 45 public buildings (2018–2022), with matched weather and building metadata (*building_id*, floor area, building category, construction year, reported number of users, etc.), representative of a Nordic, heating-dominated context (Lien et al., 2025). The weather table includes outdoor temperature, relative humidity, wind speed, and solar irradiance from the nearest meteorological station. All timestamps were converted to local time with explicit Daylight Saving Time (DST) handling; all records were stored as a unique compound key `(building_id, timestamp)`.

Target and covariates. The target is hourly site energy (kWh). Exogenous covariates include calendar encodings, weather measurements, and engineered autoregressive/rolling features (details in §3.3). All features were computed *per building* after temporal sorting to prevent cross-site bleed.

3.2 Preprocessing and cleaning

Column names and units were standardised and exact duplicates were removed. Forward-fill short gaps ($\leq$ 2 hours) in `energy_kwh` only for plotting/EDA (not for modelling); and remove obviously corrupt sensor values using conservative physical bounds (e.g.,

temperature $[-40, 45]\,^{\circ}\mathrm{C}$). Weather and load are joined via nearest-timestamp merge (max drift 15 minutes) after resampling weather to strict hourly ticks.

For modelling, remaining missing inputs were minimal after cleaning: LightGBM/ XGBoost handled residual missing values internally, Random Forest used simple per-feature median imputation fitted on the training fold, while Deep models consumed a scaled and imputed copy of the same engineered panel to avoid NaNs in tensors.

Intelligent metadata imputation (users). Reported `number_of_users` was missing for a small subset of buildings. A category-specific density was used to impute values:

Let u_j be the reported number of users and A_j the floor area (m^2) for building j. For a building i in category $c = \mathrm{cat}(i)$, define the set $\mathcal{C}(i) = \{\, j : \mathrm{cat}(j) = c,\ u_j \text{ observed} \,\}$. Compute the category density

$$\delta_c \;=\; \mathrm{median}_{\,j \in \mathcal{C}(i)} \left(\frac{u_j}{A_j} \right).$$

Let $q_{0.10}^{(c)}$ and $q_{0.90}^{(c)}$ be the 10th/90th percentiles of the same density distribution within category c. The imputed users for building i are

$$\widehat{u}_i \;=\; \left[\mathrm{clip}(\delta_c,\, q_{0.10}^{(c)},\, q_{0.90}^{(c)})\right] \;\times A_i,$$

optionally rounded to the nearest integer.

Notation:

u_j reported number of users for building j.

A_j floor area (m^2) of building j.

$\mathrm{cat}(i)$ categorical type of building i (e.g., school, kindergarten, nursing home, office).

$\mathcal{C}(i)$ set of buildings in the same category as i with non-missing u_j.

δ_c median users-per–m^2 density for category c.

$q_{0.10}^{(c)}, q_{0.90}^{(c)}$ 10th/90th percentiles of the category-c density distribution.

$\mathrm{clip}(x, a, b)$ constrains x to the interval $[a, b]$.

$\widehat{u}_i$ imputed number of users for building i.

In this dataset, densities were computed from non-missing buildings within each category (schools, kindergartens, nursing homes, offices); if category support was insufficient, the global median density was used. A binary flag `users_imputed_flag`=1 marked imputations. The imputed value served only as a static descriptor and was not used to infer time-varying occupancy (see Notebook 1; Fig. 1).

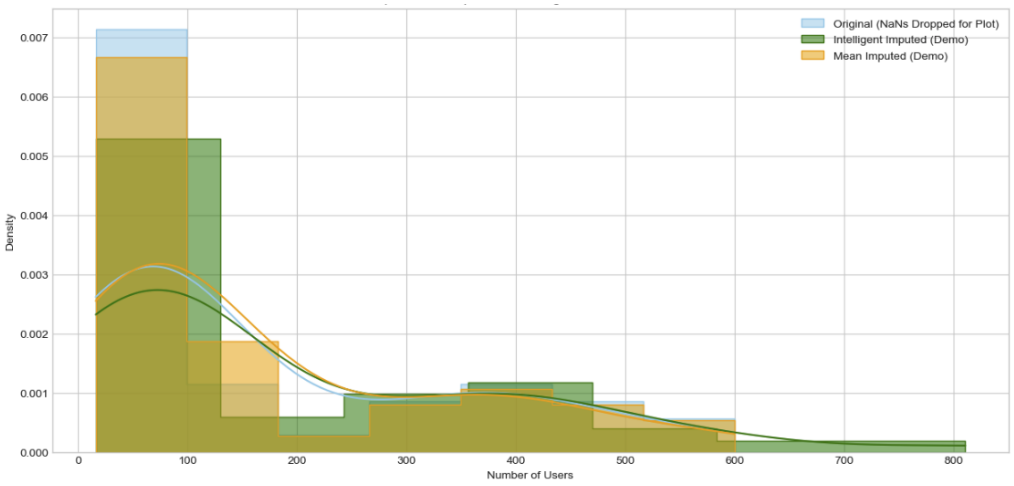

Figure 1: Imputation strategy comparison for `number_of_users`: raw vs. category density vs. global density, with clipping.

3.3 Feature-engineering pipeline (core contribution)

A feature-engineering-first approach was adopted: over 100 candidates were derived from time, weather, and lagged load, then reduced to 35 high-value predictors (§3.4). All features were computed *per building* using only information available at or before time t.

3.3.1 Cyclical time features

Periodic indices were mapped to the unit circle to preserve wrap-around proximity:

$$\text{hour\_sin}_t = \sin\left(2\pi \frac{\text{hour}(t)}{24}\right), \qquad \text{hour\_cos}_t = \cos\left(2\pi \frac{\text{hour}(t)}{24}\right),$$

and analogously for day-of-week (7) and month (12). Binary flags indicated weekends and Norwegian public holidays.[1]

3.3.2 Autoregressive (lagged) features

To capture short-, daily- and weekly-cycle dependence, lagged targets and temperature lags were computed:

$$y_{t-1},\ y_{t-2},\ y_{t-3},\ y_{t-24},\ y_{t-25},\ y_{t-26},\ y_{t-48},\ y_{t-168},\ y_{t-167},\ y_{t-169},$$

with the same index set applied to T_t (outdoor temperature). A short-recency mean $\overline{y}_{t-1:t-3}$ was also included. Lags were generated after sorting within each building; rows with insufficient history were excluded from training folds only.

3.3.3 Rolling-window statistics

Aligned, strictly past-looking windows were computed on *both* load and outdoor temperature:

$$\text{mean}(\cdot)_{[t-w,t-1]},\ \text{std}(\cdot)_{[t-w,t-1]},\ \text{min/max}(\cdot)_{[t-w,t-1]},$$

for $w \in \{3, 6, 12, 24, 72, 168\}$ hours. All windows closed at $t-1$.

[1] An early day-of-week cyclic encoding introduced a Monday/Sunday collision; this is discussed in §6.

3.3.4 Domain-specific and interaction features

Temperature–time interactions were engineered to capture schedule-dependent sensitivity (e.g., $T_t \times \text{hour\_sin}_t$, $T_t \times \text{hour\_cos}_t$, and $T_t \times \text{Is\_Weekend}$). Static descriptors (area, construction year, imputed users, basic operational markers) were included with restrained interactions. For deep models, the raw calendar/temperature streams plus a minimal set of lags were provided; tree models consumed the full engineered set.

3.4 Feature selection

The initial set of 100+ candidates was reduced to 35 predictors via a two-stage procedure, with categorical metadata one-hot encoded (e.g., building category, primary space heating type, ventilation type, lighting control, central heating system, energy label).

Table 3: Final 35 predictors used across *all* model families (grouped by type).

Group	**Feature name**
Cyclical time	hour_sin, hour_cos dow_sin, dow_cos month_sin, month_cos
Calendar / flags	DayOfWeek (0–6) Is_Holiday_NO
Autoregressive (target)	y_lag_1, y_lag_2, y_lag_3 y_lag_24, y_lag_48, y_lag_168 y_lag_167, y_lag_169 y_lag_25, y_lag_26 y_mean_1_3 (short recency mean)
Rolling stats (target)	y_mean_6, y_std_6 y_mean_24, y_std_24, y_min_24, y_max_24 y_mean_168, y_std_168
Temperature & interactions	temp_t (outdoor at t) temp_lag_1, temp_lag_24 temp_mean_6, temp_mean_24 temp_x_hour_sin, temp_x_hour_cos
Static metadata	area_m2 construction_year users_imputed

(1) Filter stage. A `VarianceThreshold` of 0.0 (drop only zero-variance) removed none in practice; from any pair with absolute Pearson correlation $|\rho| > 0.95$ one feature was dropped. During EDA (Notebook 1) raw columns with $> 70\%$ missing were removed; in the feature-engineering stage (Notebook 2) engineered candidates with $> 40\%$ missing were excluded unless operationally essential (then retained with a flag).

(2) Model-based stability selection. A LightGBM model was trained on rolling, leakage-safe time splits (expanding origin; grouped by building). Split-wise gain importances were averaged; features selected in at least 70% of splits were retained and

sanity-checked with permutation importance on the final validation window. The *same* 35-feature set was then used for *all* model families to ensure a fair, architecture-only comparison.

Reproducibility. All steps were implemented as deterministic pipeline components (Notebook 1: EDA/cleaning; Notebook 2: feature generation/selection). The process follows CRISP-DM at a high level (business understanding → data → modelling → evaluation), with emphasis on leakage controls and consistent folds. Figure 2 in §4 summarises the four phases used throughout the thesis.

4 Design and Implementation Specifications

This chapter describes the forecasting task, experiment protocol, model configurations, and the software/hardware environment used to execute the study. All steps respected temporal order and applied identical engineered inputs across model families to isolate architectural effects. The design followed the guidance to provide a complete, verifiable account of data handling and modelling decisions.

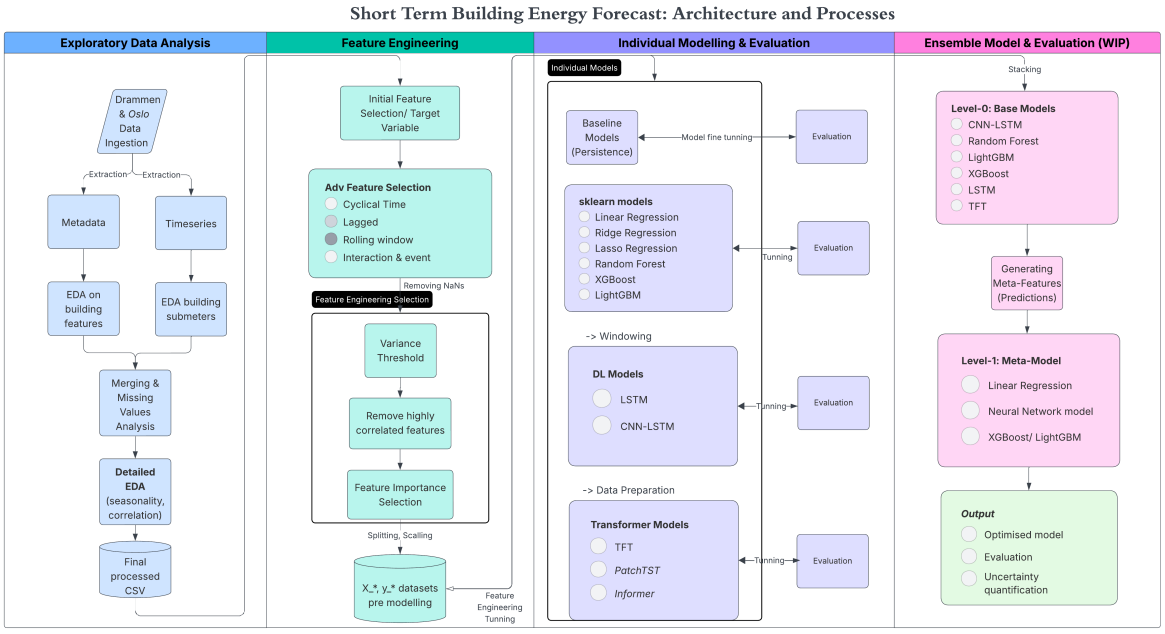

Figure 2: System architecture and process flow used in this thesis (four phases).

4.1 Experimental design

Task and horizon. Hourly site-load forecasting for a heterogeneous municipal portfolio was formulated as a supervised learning task. All models produced point forecasts at hourly resolution. Sequence models used a multi-horizon setting (24-step ahead) with a fixed 72-hour lookback window.

Chronological splits. To avoid leakage and mirror operations, the data were split chronologically: training up to 2020-12-31, validation to 2021-06-30, and testing thereafter. Splits were applied to a MultiIndex `(building_id, timestamp)` with timezone-aware timestamps. The final 35-feature panel (§3.4) was reassembled and split using the same boundaries.

Scaling and sanitisation. For tabular models (Random Forest, XGBoost, LightGBM), the engineered 2D panel was consumed *without feature scaling*; categorical one-hots were left as-is. Deep models used a *scaled and imputed* copy (StandardScaler fit on the training fold only, then applied to validation/test) to avoid NaNs in tensors. This unified pipeline ensured DL models received appropriately scaled inputs, while tree models—being scale-invariant at splits—were unaffected.

Baselines. Persistence baselines (1 h, 24 h, 168 h) were computed on identical splits to contextualise MAE, RMSE, R^2, and CV(RMSE).

4.2 Model architectures and configurations

4.2.1 Baselines (persistence)

The baseline set comprised: (i) one-hour lag persistence, (ii) 24-hour seasonal persistence, and (iii) 168-hour weekly persistence. These required no training and provided transparent anchors for the main metrics.

4.2.2 Tree-based models

Random Forest, XGBoost, and LightGBM consumed the 2D engineered feature set of 35 predictors shared across families (§3.4). Models were fitted on the training period and evaluated on validation and test without refitting on future data. LightGBM used `metric=mae` and a modest learning rate (0.05); XGBoost used `reg:squarederror`. On the final split (approx. 1.16M training rows), XGBoost trained in approx. 2.6 s and LightGBM in approx. 3.3 s; Random Forest required approx. 116 s.

4.2.3 Deep learning models

Sequence models operated on 3D tensors constructed via a sliding window (lookback 72, horizon 24) from the scaled feature panel. Architectures included:

- **LSTM (seq2seq):** two stacked LSTM blocks (64 and 32 units) with a time-distributed dense head to 24 outputs; early stopping mitigated overfitting. Reported training time: ∼13,497 s.
- **CNN–LSTM:** a shallow 1D convolution + max-pooling front-end feeding stacked LSTMs, then dense layers to 24 outputs; training time: ∼2,238 s.
- **Temporal Fusion Transformer (TFT):** implemented with `pytorch-forecasting` (MAE loss, early stopping); effective training time: ∼21,831 s on Apple Silicon (MPS).

Visuals. Figure 3 illustrates the 72→24 sliding-window construction; Figure 4 shows a compact CNN–LSTM schematic.

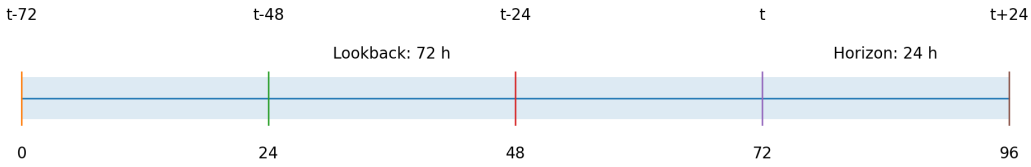

Figure 3: Sequence construction for DL models: 72-hour lookback to 24-hour horizon.

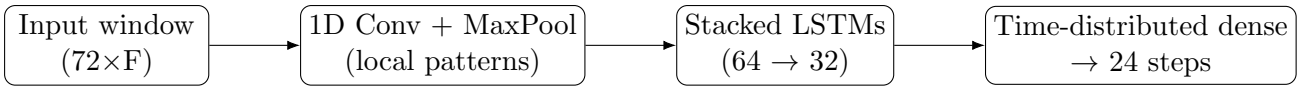

Figure 4: CNN–LSTM for multi-horizon forecasting: convolution/pooling extracts short-range structure, stacked LSTMs encode longer temporal dependencies, and a time-distributed head emits 24 hourly outputs.

4.2.4 Stacking ensembles

The ensemble stacked diverse base learners (RF, XGBoost, LightGBM, LSTM, CNN–LSTM, TFT) using a level-1 meta-learner. To generate meta-features without the compute of time-series cross-validation, a *fixed validation-set* strategy was adopted: base models were trained on the training period (to 2020-12-31); their predictions on the validation period (to 2021-06-30) formed an aligned meta-feature matrix, and a ridge or LightGBM meta-learner was fit on those predictions against the true validation targets. The trained meta-learner was then evaluated on aligned test-set meta-features. This design traded the theoretical robustness of out-of-fold (OOF) stacking for computational feasibility given long sequence/TFT training times. The LightGBM stack achieved MAE $\approx 3.58\,\text{kWh}$ and $R^2 \approx 0.978$, comparable to the best single tree; the ridge stack performed similarly (MAE $\approx 3.70\,\text{kWh}$). Meta-learner training time was $< 0.5\,\text{s}$.

4.3 Implementation environment and tuning

Experiments were executed in Python with `pandas`/`numpy` for data handling; `scikit-learn`, `xgboost`, and `lightgbm` for tabular models; `tensorflow`/`keras` for LSTM and CNN–LSTM; and `pytorch-forecasting`/Lightning for TFT. Early stopping and modest heuristic/grid tuning kept compute budgets realistic for municipal deployment. Training-time logging appears in the notebooks; representative wall-clock times on the final split are summarised in Table 4.

4.4 Reproducibility checklist

- **Keys & time:** unique key `(building_id, timestamp)`; timezone-aware; DST handled before merging; hourly regularisation of weather; nearest-timestamp join with $\leq$15 min tolerance.
- **Splits:** Train $\leq$ 2020-12-31; validation $\leq$ 2021-06-30; test $>$ 2021-06-30; applied after feature assembly; no refit on future data.
- **Features:** Candidate set >100; final = 35 predictors shared across all models; all lags/rolls computed per building, aligned to close at $t-1$ (no look-ahead).

Table 4: Model families, key setup, and indicative train times on the final split.

Model	Key configuration (inputs & setup)	Train time
Random Forest	35 engineered features; chronological train/val/test; evaluated on test without refit.	∼116 s
XGBoost	Same features; `reg:squarederror` objective; modest LR; early stopping on validation.	∼2.6 s
LightGBM	`metric=mae`, LR=0.05; 35 features; ∼1.16M train rows.	∼3.3 s
LSTM	Seq2seq; lookback 72 → horizon 24; early stopping; scaled inputs.	∼13,497 s
CNN–LSTM	1D Conv + Pool → LSTM stack; 72 → 24; scaled inputs.	∼2,238 s
TFT	`pytorch-forecasting` + Lightning; MAE loss; hidden size 32; 4 heads; early stopping.	∼21,831 s
Stack (ridge)	Validation-set meta-features → ridge meta; leakage-safe split; test on holdout.	< 0.2 s
Stack (LGBM)	As above with LightGBM meta-learner; similar accuracy.	< 0.5 s

- **Scaling:** DL models used StandardScaler fit on train; trees used unscaled engineered features (scale-invariant at splits); one-hots unscaled.
- **Baselines:** Persistence at 1 h, 24 h, 168 h on identical splits.
- **Model configs:** RF/XGBoost/LightGBM on 35 features; LSTM and CNN–LSTM with 72 → 24; TFT via PyTorch Forecasting/Lightning (hidden size 32, 4 heads, MAE loss, early stopping).
- **Ensembling:** Validation-set stacking; aligned meta-feature matrices for val/test; ridge and LightGBM meta-learners; no OOF due to compute.
- **Compute logging:** Wall-clock train times captured per model on the final split (RF ∼116 s; XGB ∼2.6 s; LGBM ∼3.3 s; LSTM ∼13,497 s; CNN–LSTM ∼2,238 s; TFT ∼21,831 s).
- **Determinism:** Fixed random seeds (where supported) for scikit-learn/ XGBoost/ LightGBM/ Keras/ Lightning; identical feature lists across families; explicit index alignment for meta-features before stacking.

5 Evaluation

This section evaluates the competing model families on a held-out test set under a unified, leakage-safe protocol using the 35 selected predictors (§3). Accuracy (MAE, RMSE, CV(RMSE), R^2) and computational efficiency (wall-clock training time) are reported, and ensembles are compared against their base learners.

5.1 Metrics and experimental setup

The primary accuracy metrics were MAE, RMSE, R^2, and CV(RMSE), reported on the chronological test window defined in §4. Three persistence baselines (1h, 24h, 168h) provided intuitive anchors. Training time was logged to contextualise accuracy gains against computational cost.

5.2 Overall accuracy on the test set

Tree ensembles delivered the highest test-set accuracy. Random Forest achieved the best MAE of 3.30 kWh with $R^2 = 0.9819$, followed by XGBoost (MAE 3.42 kWh, $R^2 = 0.9817$) and LightGBM (MAE 3.58 kWh, $R^2 = 0.9803$). These results substantially outperformed linear baselines and all persistence references, confirming the effectiveness of the engineered representation for tabular learners (Notebook 3).

Deep sequence models underperformed on this portfolio: LSTM (MAE 10.13 kWh, $R^2 = 0.8623$) and CNN–LSTM (MAE 12.44 kWh, $R^2 = 0.8071$) lagged the stronger persistence baseline, while the best TFT configuration achieved MAE 5.11 kWh with $R^2 \approx 0.952$ (Notebook 3).

Table 5: Test-set accuracy and training time (final split). Lower is better for MAE/RMSE/CV(RMSE).

Model	MAE (kWh)	RMSE (kWh)	CV(RMSE) %	R^2	Train (s)
Random Forest	3.300473	6.403286	14.475689	0.981881	115.796
XGBoost	3.418656	6.443359	14.566280	0.981653	2.620
LightGBM	3.577596	6.678540	15.097946	0.980290	3.272
Stack (LGBM meta)	3.582445	7.030462	15.811166	0.978240	0.439
Stack (Ridge meta)	3.697746	7.051277	15.857978	0.978111	0.180
Weighted average	4.080889	7.841386	17.634894	0.972931	0.011
Lasso	4.201299	7.880407	17.814965	0.972557	3.792
Linear	4.214552	7.767148	17.558924	0.973340	0.571
Ridge	4.214558	7.767156	17.558944	0.973340	0.179
Persist-1h	4.561322	9.586920	21.672821	0.959385	–
TFT (MAE loss, tuned)	5.113802	10.423864	23.568891	0.951953	21831.254
TFT (baseline)	8.576343	13.441574	19.510801	0.947853	21831.254
Persist-24h	8.761971	19.383023	43.818534	0.833974	–
Persist-168h	9.620687	19.258531	43.537100	0.836100	–
LSTM	10.132110	17.685640	39.774141	0.862303	13496.862
CNN–LSTM	12.435477	20.930222	47.071046	0.807146	2237.679

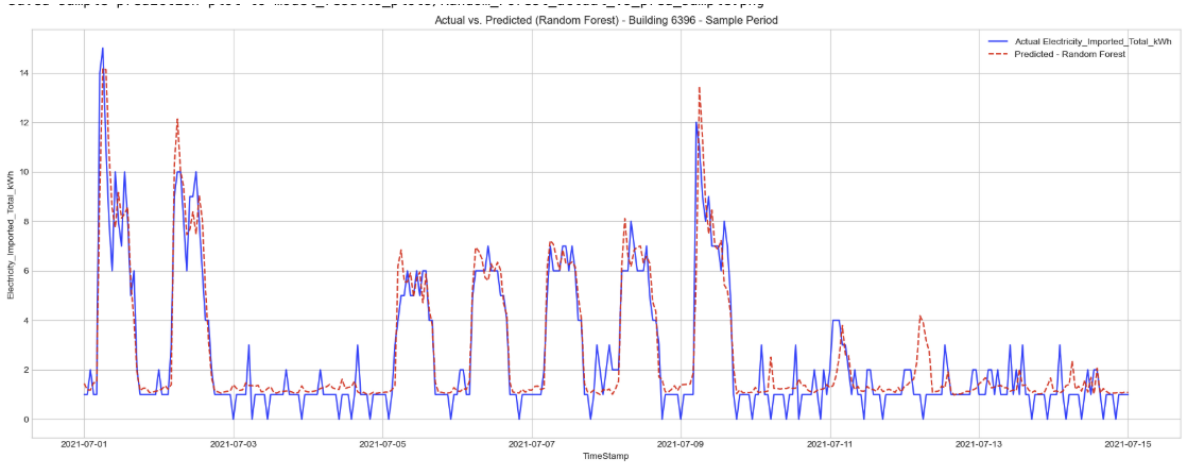

Figure 5: Actual vs. predicted load for Random Forest on a representative building (two-week slice, test period).

5.3 Compute–accuracy trade-offs

Training time differences were pronounced. On the final split, XGBoost trained in ∼2.62 s and LightGBM in ∼3.27 s, versus ∼13,497 s (∼3.75 h) for LSTM, ∼2,238 s for CNN–

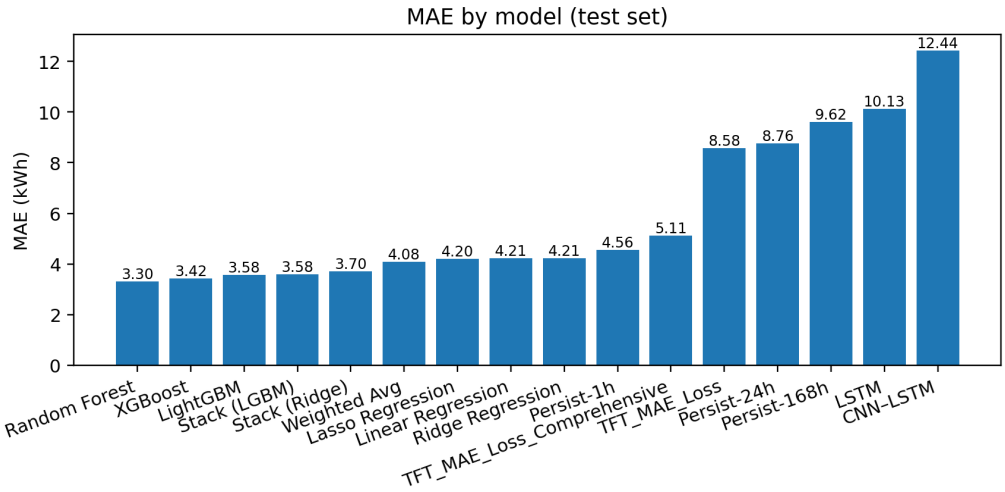

Figure 6: MAE by model on the test set (lower is better).

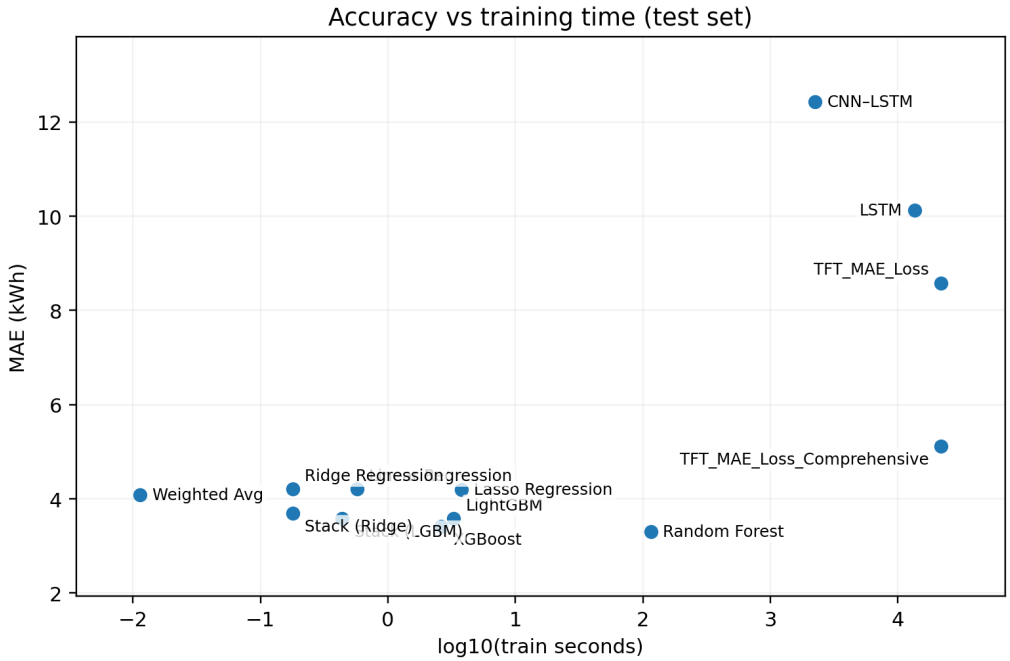

Figure 7: Accuracy vs. training time (x-axis in $\log_{10}$ seconds).

LSTM, and ∼21,831 s (∼6.06 h) for TFT. Thus, tree ensembles not only provided the best accuracy but also orders-of-magnitude lower compute cost, strengthening their suitability for portfolio-scale, iterative operations (Notebook 3 timing logs).

5.4 Ensembles

Validation-set stacking (level-1 meta-learner trained on validation predictions) achieved MAE 3.58–3.70 kWh and $R^2 \approx 0.978$ on the test set. This was competitive but did not surpass the best single tree (Random Forest). A simple weighted-average ensemble performed worse (MAE 4.08 kWh). Given the substantial training time of TFT and sequence models, out-of-fold stacking was not used on cost grounds; this limitation is revisited in §6.

5.5 Per-building performance and robustness

Errors varied by building, as expected in a heterogeneous municipal portfolio. Targeted lags (1, 24, 168 h) and rolling statistics generally stabilised accuracy across occupancy schedules and weather regimes, while the small gains from stacking suggested most attainable signal was already captured by the feature-engineered trees. A detailed per-building breakdown (MAE distribution and ranks) is provided in Appendix X.

5.6 Interpretability

Gain-based feature importance for tree models highlighted calendar encodings, targeted lags, and weekly/short-window statistics, consistent with heating-dominated operations. Where available, SHAP/permutation analyses corroborated these findings, attributing marginal contributions to recency and weekly seasonality. Detailed panels are collated in Appendix Y.

5.7 Summary of findings

In this setting, a feature-engineering-first approach paired with tree ensembles achieved the best balance of accuracy and efficiency. Deep sequence models trained considerably longer yet fell short on accuracy, with TFT narrowing but not closing the gap. A lightweight stack added small robustness but did not change the model ordering. These outcomes support the central hypothesis that careful representation design enables efficient tabular learners to match or exceed more complex architectures for STBELF.

5.8 Threats to validity and sensitivity checks

Two items should be mentioned. First, an early day-of-week cyclic encoding used denominators 23/6 - later code and documentation use 24/7. The retained raw `DayOfWeek` and strong lag/rolling features mitigated impact on trees - future model retrains will use the corrected encoding. Second, validation-set stacking (instead of out-of-fold) trades some robustness for compute savings - running OOF stacking is identified as future work.

6 Conclusions and Discussion

6.1 Summary of findings

This thesis evaluated linear models, tree ensembles (Random Forest, XGBoost, LightGBM), and deep sequence models (LSTM, CNN–LSTM, TFT) on a four–year, 45–building municipal portfolio under a unified, leakage-safe framework and a common set of 35 engineered features/ predictors. Tree ensembles achieved the best balance of accuracy and efficiency on the held-out test set: Random Forest delivered the lowest MAE (3.30 kWh; $R^2 = 0.982$), with XGBoost (MAE 3.42 kWh) and LightGBM (MAE 3.58 kWh) close behind. Relative to the strongest persistence baseline (1 h lag; MAE 4.56 kWh), Random Forest reduced error by $\approx 27.6\%$, and by over 62% versus 24 h seasonal persistence. Deep models underperformed in this setting: LSTM and CNN–LSTM exhibited materially higher errors, and the best TFT configuration (MAE ≈ 5.11 kWh) did not surpass the 1 h persistence anchor. Training-time differences were substantial: XGBoost and LightGBM trained in seconds, whereas TFT required $\sim$6.1 h—$\sim$8,300$\times$ longer than XGBoost and $\sim$6,700$\times$ longer than LightGBM on the same split. A validation-set stacking ensemble matched but did not exceed the best single tree. These outcomes are consistent with the pragmatic guidance in Table 2, which summarises when each model family is most effective under typical STBELF data conditions.

6.2 Interpretation and relation to prior work

The performance ordering aligns with evidence from tabular learning and prior building-domain studies: targeted lags ($t-1$, $t-24$, $t-168$), short/weekly rolling statistics, and calendar encodings capture most operational signal in hourly loads. In this portfolio, short- and weekly-recurrence plus temperature-driven thresholds were more influential than very long sequential dependencies, reducing the relative advantage of end-to-end sequence models. TFT's attention over mixed covariates narrowed the gap but remained compute-intensive, reinforcing the practical advantage of tree ensembles for portfolio-scale iteration.

6.3 Practical implications

For municipal building energy management, tree ensembles offer (i) strong accuracy with low latency, (ii) stable behaviour across heterogeneous buildings, and (iii) accessible interpretability (feature importance/SHAP) for diagnostics and handover. Where multi-horizon outputs with rich exogenous context are mission-critical, a Transformer can be justified, but the cost–benefit should be assessed against the marginal gains observed here. Lightweight stacking can add small robustness but increases operational complexity.

6.4 Scope and scalability/ generalisability

The dataset reflects a Nordic, heating-dominated context (schools, kindergartens, nursing homes, offices). Results therefore generalise best to similar climates and public-building portfolios. The engineered features/ predictors were intentionally generic (calendar, temperature recency, autoregressive and rolling features) and should transfer easily. However, cooling-dominated portfolios, sub-hourly controls, or rich Behind the Meter (BTM) tech-

nologies may favour architectures that exploit longer temporal context and cross-entity learning.

6.5 Limitations and threats to validity

First, early notebook versions encoded day-of-week and hour cycles with non-standard denominators (23/6), collapsing endpoint categories. The final selected set retained the raw integer `DayOfWeek` with high importance, and lag/rolling features dominated, so impact is assessed as negligible; the codebase now uses standard 24/7 encodings (but it hasn't been executed due to high training times against a tight deadline to publish this thesis). Second, stacking used a fixed validation-set strategy rather than out-of-fold meta-features—a deliberate compute trade-off given long TFT/sequence training times—which may slightly over-estimate ensemble generalisation. Third, hyperparameter tuning for deep models was intentionally modest to respect realistic compute budgets; stronger tuning might narrow the gap but at substantial cost. Finally, evaluation used observed weather; explicit stress-tests under forecasted-weather noise were discussed but not executed end-to-end.

6.6 Conclusions

Across 45 buildings and four years of hourly data, a feature-engineering-first pipeline paired with tree ensembles delivered the best accuracy–efficiency trade-off for short-term building energy load forecasting. Sequence models trained far longer and did not outperform the strongest tabular baselines - a simple stack provided small additional robustness but did not change the ordering. For typical municipal building portfolios and 1–24 h horizons, tree ensembles should be considered the default baseline, with deep architectures reserved for settings where their specific advantages (multi-horizon forecasting with rich exogenous context and cross-entity transfer) are essential —*a conclusion that aligns with the synthesis in Table 2.*

6.7 Future work

Promising directions include: (i) probabilistic forecasting (quantile losses, conformal calibration) for risk-aware scheduling and BESS dispatch; (ii) leakage-safe out-of-fold stacking/blending with horizon-aware meta-features; (iii) cross-entity transfer/meta-learning for cold-start buildings and rapid onboarding; (iv) explicit evaluation under forecasted-weather uncertainty and regime shifts; (v) concept-drift monitoring and online adaptation; (vi) cost-aware objectives reflecting tariffs and control actuation; and (vii) extension to additional Nordic portfolios (e.g., Oslo) and to building-specific fine-tuning where heterogeneity is extreme.

References

Amasyali, K. and El-Gohary, N. M. (2018). A review of data-driven building energy consumption prediction studies, *Renewable and Sustainable Energy Reviews* **81**: 1192–1205.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S1364032117306093*

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M. and Kasneci, G. (2024). Deep Neural Networks and Tabular Data: A Survey, *IEEE Transactions on Neural Networks and Learning Systems* **35**(6): 7499–7519. arXiv:2110.01889 [cs].
URL: *http://arxiv.org/abs/2110.01889*

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. arXiv:1603.02754 [cs].
URL: *http://arxiv.org/abs/1603.02754*

Chesbrough, H. (2020). *Enel X: Driving Digital Transformation in the Energy Sector*, The Berkeley-Haas Case Series. University of California, Berkeley. Haas School of Business, 1 Oliver's Yard, 55 City Road, London EC1Y 1SP United Kingdom.
URL: *https://sk.sagepub.com/cases/enel-x-driving-digital-transformation-in-the-energy-sector*

Deb, C., Zhang, F., Yang, J., Lee, S. E. and Shah, K. W. (2017). A review on time series forecasting techniques for building energy consumption, *Renewable and Sustainable Energy Reviews* **74**: 902–924.
URL: *https://www.sciencedirect.com/science/article/pii/S1364032117303155*

EirGrid (2025). Renewables provide a third of electricity as July marks first month of new coal-free era for Ireland's power system.
URL: *https://www.eirgrid.ie/news/renewables-provide-third-electricity-july-marks-first-month-new-coal-free-era-irelands-power*

European Commission (2020). 2030 climate targets - European Commission.
URL: *https://climate.ec.europa.eu/eu-action/climate-strategies-targets/2030-climate-targets$_e$n*

Fang, T. and Lahdelma, R. (2016). Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system, *Applied Energy* **179**: 544–552.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S0306261916309217*

Gandhi, O., Zhang, W., Kumar, D. S., Rodríguez-Gallegos, C. D., Yagli, G. M., Yang, D., Reindl, T. and Srinivasan, D. (2024). The value of solar forecasts and the cost of their errors: A review, *Renewable and Sustainable Energy Reviews* **189**: 113915.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S1364032123007736*

Grinsztajn, L., Oyallon, E. and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data?, *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Curran Associates Inc., Red Hook, NY, USA, pp. 507–520.

Hirvonen, S. (2024). Comparison of data-driven models for building energy load forecasting.

Hong, T. and Fan, S. (2016). Probabilistic electric load forecasting: A tutorial review, *International Journal of Forecasting* **32**(3): 914–938.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S0169207015001508*

Hyndman, R. J. and Athanasopoulos, G. (2018). Forecasting: Principles and Practice. Publisher: OTexts.
URL: *https://research.monash.edu/en/publications/forecasting-principles-and-practice-2*

IEA (2023a). Ireland 2024 – Analysis.
URL: *https://www.iea.org/reports/ireland-2024*

IEA (2023b). Norway - Countries & Regions.
URL: *https://www.iea.org/countries/norway*

IEA (2024). Electricity 2024 – Analysis and forecast to 2026.
URL: *https://www.iea.org/reports/electricity-2024*

Jomaux, J. (2025a). Heatwave on Europe: the impact of a "Hitzeflaute".
URL: *https://gemenergyanalytics.substack.com/p/heatwave-on-europe-the-impact-of*

Jomaux, J. (2025b). What has been the impact of solar in Europe so far this year? A Mid-May Review.
URL: *https://gemenergyanalytics.substack.com/p/what-has-been-the-impact-of-solar*

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree.

Kim, T.-Y. and Cho, S.-B. (2019). Predicting residential energy consumption using CNN-LSTM neural networks, *Energy* **182**: 72–81.
URL: *https://www.sciencedirect.com/science/article/pii/S0360544219311223*

Kuhn, M. and Johnson, K. (2020). *Feature engineering and selection: a practical approach for predictive models*, Chapman & Hall/CRC data science series, CRC Press, Taylor & Francis Group, Boca Raton London New York.

Li, L., Su, X., Bi, X., Lu, Y. and Sun, X. (2023). A novel Transformer-based network forecasting method for building cooling loads, *Energy and Buildings* **296**: 113409.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S0378778823006394*

Lien, S. K., Walnum, H. T. and Sørensen, L. (2025). COFACTOR Drammen dataset - 4 years of hourly energy use data from 45 public buildings in Drammen, Norway, *Scientific Data* **12**(1): 393.
URL: *https://www.nature.com/articles/s41597-025-04708-3*

Lim, B., Arık, S. , Loeff, N. and Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting, *International Journal of Forecasting* **37**(4): 1748–1764.
URL: *https://www.sciencedirect.com/science/article/pii/S0169207021000637*

Lundberg, S. and Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. arXiv:1705.07874 [cs].
URL: *http://arxiv.org/abs/1705.07874*

Machlev, R., Heistrene, L., Perl, M., Levy, K. Y., Belikov, J., Mannor, S. and Levron, Y. (2022). Explainable Artificial Intelligence (XAI) techniques for energy and power systems: Review, challenges and opportunities, *Energy and AI* **9**: 100169.
URL: *https://www.sciencedirect.com/science/article/pii/S2666546822000246*

Miller, C., Hao, L. and Fu, C. (2022). Gradient boosting machines and careful pre-processing work best: ASHRAE Great Energy Predictor III lessons learned. arXiv:2202.02898 [cs].
URL: *http://arxiv.org/abs/2202.02898*

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V. and Gulin, A. (2018). CatBoost: unbiased boosting with categorical features, *Advances in neural information processing systems*.

Rafi, S. H., Nahid-Al-Masood, Deeba, S. R. and Hossain, E. (2021). A Short-Term Load Forecasting Method Using Integrated CNN and LSTM Network, *IEEE Access* **9**: 32436–32448.
URL: *https://ieeexplore.ieee.org/document/9358156/*

Shwartz-Ziv, R. and Armon, A. (2022). Tabular data: Deep learning is not all you need, *Information Fusion* **81**: 84–90.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S1566253521002360*

Wang, Z., Hong, T. and Piette, M. A. (2020). Building thermal load prediction through shallow machine learning and deep learning, *Applied Energy* **263**: 114683.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S0306261920301951*

Wei, Y., Zhang, X., Shi, Y., Xia, L., Pan, S., Wu, J., Han, M. and Zhao, X. (2018). A review of data-driven approaches for prediction and classification of building energy consumption, *Renewable and Sustainable Energy Reviews* **82**: 1027–1047.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S136403211731362X*

Yildiz, B., Bilbao, J. and Sproul, A. (2017). A review and analysis of regression and machine learning models on commercial building electricity load forecasting, *Renewable and Sustainable Energy Reviews* **73**: 1104–1122.
URL: *https://linkinghub.elsevier.com/retrieve/pii/S1364032117302265*

Zhao, H. and Magoulès, F. (2012). A review on the prediction of building energy consumption, *Renewable and Sustainable Energy Reviews* **16**(6): 3586–3592.

***AI Usage Disclaimer*:** This paper was developed with the support of OpenAI's ChatGPT and Google's Gemini. These language models were used during coding and throughout the writing process, including rewording sections for clarity and flow, assisting in ideation, organizing the structure of the paper, and interpreting and understanding complex concepts from the literature reviewed. The insights provided by ChatGPT and Gemini were instrumental in enhancing the quality of this paper, although final analysis, interpretations, and conclusions are my own.