Creating a Calibrated Short-Horizon Net-Load Forecasting using MC-Dropout LSTM and Residual-Based Anomaly Detection for the Irish Power System

A Dissertation submitted in partial satisfaction of the requirements

for the degree “Master of Science in Artificial Intelligence & Machine Learning”

by

JONATHAN MUWANGUZI

21189676

Supervised by Alison O’Connor

Department of Computer Science and Information Systems

Faculty of Science and Engineering

University of Limerick

25/08/2025

Statement

I authorise the University of Limerick to make this dissertation available through the University of Limerick Library and to share it with Computer Science and Information Systems (CSIS) students for academic use.

Abstract

The management of Ireland's power system serves as an important case study for grids with high levels of non-synchronous renewables, primarily wind, and frequent exposure to North Atlantic weather. The increasing System Non-Synchronous Penetration (SNSP) highlights the limitations of traditional point forecasts. This work addresses two key questions for system operations: how to create 15-minute net-load forecasts with calibrated uncertainty, and how to identify anomalies by tracking forecast errors. The framework developed aims to provide clear and practical short-term guidance for operational decisions.

The study assembles a 2014-2025, 15-minute dataset by merging public data of EirGrid operational measurements (demand, generation, wind, interconnection, SNSP) with quadrant-level weather summaries from Met Éireann. The pipeline regularizes timestamps, reconciles overlaps, and creates lagged and rolling features. A fair comparison plan is used through walk-forward cross-validation on historical periods and a separate, recent test window. Baselines include persistence, exponential smoothing, and an Autoregressive Integrated Moving Average-Generalized Autoregressive Conditional Heteroskedasticity (ARIMA-GARCH) variant. The main forecast model is a Long short-term memory (LSTM) trained with Monte-Carlo (MC) dropout to form predictive distributions. A wide initial feature set is reduced to a compact subset using mutual information and tree-based importance to balance accuracy and stability. Forecast evaluation uses RMSE for point quality and empirical coverage at 80% and 95% for calibration. For anomaly detection, residuals (actual minus forecast) form the basis for a σ-threshold benchmark, Isolation Forest (IF), One-Class Support Vector Machine (OC-SVM), and Deep Support Vector Data Description (DeepSVDD). A custom time-series splitter ensures each test fold contains enough candidate anomalies. Precision, recall, F1, and false-alarm rate are reported on the hold-out.

The work aims to find that a reduced-feature LSTM achieves lower test errors than classical baselines and that MC-dropout intervals can meet coverage targets without being overly wide. Among detectors, DeepSVDD and OC-SVM are expected to keep recall near 1.0 while improving precision relative to simple rules, especially in seasons with rapid wind shifts. The analysis also anticipates that a parsimonious feature set leads to more stable coverage and lower maintenance burden across retraining cycles.

This study aims to have an incremental and operational impact. Calibrated forecasts can be integrated into existing dashboards to show both a central path and uncertainty bands, supporting reserve setting and short-term scheduling. Residual-based alarms can alert analysts to potential data quality issues, forecast regime breaks, or unusual net-load ramps, prioritizing human review without frequent false positives. The evaluation choices of time-ordered validation, paired model tests, and explicit coverage checks provide a transparent template for ongoing model governance. Overall, the approach aims to fit into existing workflows: simple inputs, interpretable diagnostics, and periodic retraining to keep performance steady across seasons.

Acknowledgements

This work benefited from the time and support of several people. I first wish to thank my supervisor, Alison O'Connor from the Department of Computer Science & Information Systems. Your patience when I slipped on deadlines or arrived with little to show and the calm feedback and practical suggestions helped me set small goals and keep the work moving.

I also thank Gauri Vaidya, my lecturer for Research Methods and Specification. The templates and guidance you provided gave me a clear starting point for the dissertation. The notes on structure, scope, and academic style helped me write more clearly. Your support gave me the confidence that a good thesis was possible if I followed the plan and kept revising.

To my parents, Mr. Pasteur Kavuma and Dr. Susan Kavuma the financial, and emotional support you effortless provided throughout my master's and this thesis has been a pillar on which my strength is based. The weekly calls helped me manage stress and stay focused. Enabling me to balance the project with the rest of my responsibilities and to keep steady progress rather than wait for a perfect result.

I am grateful to my close friends Mark Wilson Kirumira and Anderson Ongaria Terry, for their encouragement during the summer project. Short messages, quick chats, and simple "how is it going?" questions made a real difference when energy was low.

Finally, thank you to classmates, friends and administrative staff who helped with schedules and practical matters. Any errors or omissions in this dissertation are mine alone. I appreciate the patience and support that made the work manageable and helped me reach the end.

Data Source and Generative AI

Data for this study are public, and the primary sources were EirGrid and Met Éireann. Quarter-hourly system variables were downloaded from EirGrid's "System-Data-Qtr-Hourly" spreadsheets for 2014-2025. Met Éireann station data obtained via the public webdata endpoints. A station list was created from Met Éireann's database and used as a catalogue to be retrieved station series programmatically with pandas/requests. I used generative-AI tools namely; ChatGPT (OpenAI) and Gemini to compare storage options (HDF5, Zarr, memmap), sketch steps, check edge cases and code documentation/commenting. All code and decisions were mine and validated against the source files. Grammarly was used only for proofreading (grammar, spelling, punctuation). No synthetic data was generated using generative artificial intelligence.

Table of Contents

Contents

List of Figures

List of Tables

List of Equations

1. Introduction and Outline

Electricity systems are becoming more influenced by weather systems, as the share of electricity generate from wind and solar grows. The behaviour of the grid over the next time frame depends less on fixed schedules and more on what the sky and sea decide to do. This makes near-term intelligence indispensable. System operators, market participants, and infrastructure planners all act on expectations about the next few intervals, not on distant averages. The central concern of this dissertation is how to generate those expectations credibly. How to express their uncertainty honestly, and how to recognise quickly when reality departs from the expectations in ways that matter for operation.

Ireland provides a compelling stage for this inquiry. The country's electricity mix is characterised by high wind penetration, a compact geographic footprint, and interconnections that help but cannot eliminate local weather coherence. These features mean that a single frontal system can drive pronounced ramps in net load and renewable availability across the island. When such movements occur, the difference between a forecast that is merely plausible and one that is demonstrably trustworthy is not academic. But it affects how much reserve is scheduled, which plant is started, and what risks the operator can take. The approach taken here is intentionally practical, it builds a clean and auditable data pipeline from public operational and meteorological sources. It then establishes transparent classical benchmarks alongside a deliberately simple deep model. Then the uncertainty with coverage is measured rather than hope, and it frames anomaly detection in terms of the model's own residuals so that alerts are about departures from expectation rather than unusual weather per se. The emphasis is less on novelty of algorithms and more on reliability of behaviour.

Two broad phases structure the work. The first is exploratory breadth, it assembles the data carefully, sets clear baselines that are to be used as benchmarks, and trains a straightforward sequence model to understand what the joint series can support. The second is disciplined consolidation, it compresses the feature space to a compact, stable subset. It moves data access to an indexed store that supports long histories and formalises the way uncertainty bands are produced and verified. The same store reforms anomaly evaluation so that precision and recall are meaningful since anomalies are rare and their frequency is dependent on the time of the year. The arc from Phase A to Phase B mirrors how real projects mature, from "what seems promising on paper" to "what can be defended when the grid is moving".

For this dissertation we will not unveil a brand-new forecasting paradigm. Rather, it will show that with careful data handling, leakage-safe features, calibrated uncertainty, and evaluation that respects time order and rare events. Everyday tools can deliver behaviour that suits a control-room context. The literature holds up this direction of travel i.e. the move from point metrics to probabilistic verification, the residual framing of anomalies, and the practicality of Monte Carlo dropout for approximate uncertainty in sequence models (Pei, et al. 2022) (Pierre Pinson 2017).

1.1 From Conventional Grids to Modern Grid Systems

For most of the twentieth century power systems were built around large, dispatchable plant that used coal, oil, gas, and hydro with transmission designed to deliver bulk power from a few centres to many consumers. Variability came mainly from demand patterns and forced outages. Forecasting focused on load and uncertainty was handled with spinning reserve and well-rehearsed operating rules. In this "conventional" setting, tomorrow looked much like today, and physics was dominated by synchronous machines whose inertia naturally buffered frequency disturbances. (EirGrid and ESB Networks 2024) (The EirGrid Group 2023)

That template is being rewritten. As wind and solar capacity grows, a larger share of production is determined by weather rather than fuel schedules. Output is variable, sometimes non-intuitive, and concentrated in particular regions that are in most cases far from high demand areas. Inverter-based resources contribute little or no inherent inertia. Distributed solar appears behind the meter. Storage and demand response add options, but they also add decisions. In short, the grid is becoming a real-time system whose behaviour must be inferred from atmospheric dynamics as much as from plant availability. Forecasting expands from "how much demand will arrive" to "how the net of demand and weather-driven supply will evolve,". Then anomaly detection shifts from spotting rare equipment events to recognising departures from modelled expectations. (EirGrid 2025) (Lau and McSharry 2010) (Wu, et al. 2022)

Across the globe, power systems are responding to this change with unique strategies. In the United States, jurisdictions such as California and Texas face steep evening ramps as solar output falls while demand remains high. Operations have adapted with flexible gas, storage dispatch, and frequent forecast updates; price signals now reflect ramping scarcity as much as energy balance. In Germany, high penetrations of both wind and rooftop Photovoltaics (PV) have normalised negative prices at times of surplus and required curtailment protocols and redispatch tools to manage congestion. The Netherlands has responded to rapid offshore wind build-out with more granular intraday trading and tighter coordination between forecasting, balancing, and interconnector scheduling. In Great Britain, the system operator has introduced fast-acting frequency services tailored to low-inertia conditions, integrating storage and responsive demand into the core balancing toolkit. These systems differ in geography and scale, yet all now depend on short-horizon intelligence that can be trusted operationally. (Lau and McSharry 2010) (Wu, et al. 2022)

Ireland (IE) belongs in this conversation but with distinctive features. The synchronous area is small and largely isolated from the wider European grid, so disturbances are felt quickly and coherently. Plus, wind penetration is high relative to demand. Interconnectors to Great Britain help with balancing and trade, but they do not erase the fact that a single Atlantic front can move across the whole island in hours. The all-island market structure shared by the Republic of Ireland (IE) and Northern Ireland (NI) means that planning and operation must be coordinated across two jurisdictions with one physical system. Against this backdrop, methods that are transparent, leakage-safe, and easy to audit are not luxuries; they are prerequisites for confidence. (EirGrid 2025) (Lau and McSharry 2010) (Wu, et al. 2022)

The transmission system is planned and operated by two entities: EirGrid in the Republic and System Operator for Northern Ireland (SONI), together they manage the all-island power system and the Single Electricity Market. A growing share of electricity now comes from non-synchronous sources chiefly wind farms interfaced by inverters which do not contribute the natural rotational inertia that traditional generators provide. To manage system risk, the control room measures System Non-Synchronous Penetration (SNSP) which is the proportion of demand being met by non-synchronous generation and High-Voltage Direct Current (HVDC) imports at any instant. Higher SNSP means less inherent damping of frequency swings, so operators schedule faster reserves, constrain units for stability, and at times limit wind output to stay within stability bounds. Alongside SNSP, they monitor net load, wind availability, and ramp rates to plan starts and reserve cover over the next few intervals. These actions take place under real network constraints thermal, voltage, and stability that determine what can actually be dispatched. (EirGrid and ESB Networks 2024) (EirGrid 2025) (Wu, et al. 2022)

The relationship between the Republic of Ireland, Northern Ireland, and Great Britain shapes both opportunity and risk. The all-island system gains internal diversification by pooling resources north and south and external flexibility through interconnection with Great Britain. Yet the same weather regimes often sweep the Irish Sea as well, limiting diversification at precisely the times when wind is strong or weak across the region. In practice this means that Ireland cannot rely solely on imports to smooth its variability. It must improve short-horizon forecasts, calibrate uncertainty realistically and detect departures from expectation promptly so that scarce flexible resources are used well. The techniques developed in this dissertation are chosen with that operational reality in mind. They are scoped to the one-hour horizon at quarter-hourly cadence, they use publicly available operational and meteorological data, and they prioritise behaviour that can be explained and validated in the language of the control room. (EirGrid 2025); (Wu, et al. 2022)

1.2 Why forecasting and anomalies matter

In power-system operations, the cost of ignorance arrives on two fronts. If the system underestimates demand or overestimates renewable availability, reserve may be thin and frequency margins may be compromised. If it overestimates demand or underestimates renewables, unnecessary thermal starts and curtailment waste fuel and money. These are not side issues to be dealt with later by markets; they are the daily reality of balancing. Short-horizon forecasts an hour ahead, refreshed every fifteen minutes sit at the heart of this reality. They inform unit commitment and redispatch, shaping how confidently an operator can lean on non-synchronous generation and how much headroom must be carried to feel safe (EirGrid and ESB Networks 2024) (The EirGrid Group 2023) (EirGrid 2025) (Wu, et al. 2022).

A point prediction cannot carry this weight on its own. Operational decisions are exercises in risk management, not curve-fitting. What matters is the distribution around the point, and whether the system can trust stated uncertainty. The concept of coverage is the hinge: an “80%” interval should contain outcomes around eight times in ten; a “95%” interval should fail rarely and predictably. If this simple property does not hold, the numbers may be decorative but they are not useful. The dissertation treats coverage as a primary outcome to be tested and reported,

not as something to be asserted by appeal to model class. For the deep model, intervals are produced by sampling with dropout left active at test time; their credibility is then judged by empirical coverage on out-of-sample periods. For the classical baselines, variance proxies and distributional assumptions are scrutinised by the same yardstick (Pei, et al. 2022) (Azam, et al. 2025) (Florin Bunea 2011).

Anomalies are the complement to forecasts. They are not simply "odd" points they are periods when the system behaves in a way that the forecast given the information available, does not anticipate. This distinction matters labelling directly on the raw series tends to conflate unusual weather with unusual system behaviour. If a detector fires whenever wind ramps quickly, it adds little the forecast did not already imply. A residual-based detector, by contrast, watches the gap between expectation and realisation, enriched by features that stabilise the residuals across seasons and regimes. It is therefore better placed to trigger on genuinely unexpected behaviour, whether the cause is a sudden derating, or a data fault, or a novel weather system interaction. Literature evidence supports this strategy simple σ-rules set a floor on sensitivity while one-class methods can lift precision when fed stable residual structure and local variance (Xie, et al. 2023); (Xydas, et al. 2017); (Jakub Nowotarski 2018); (Pierre Pinson 2017).

The relationship between forecasting, uncertainty, and anomaly detection is not incidental. Operators care about actionability i.e. will a band hold often enough to be trusted, and do alerts occur when additional attention is warranted rather than at every gust? An over-confident forecast can be more dangerous than a modest but on the other hand a hyper-sensitive detector can be worse than none. Precision and recall must be explicitly traded according to appetite for risk, and that trade must be measured under an evaluation regime that respects time order and rarity. Thus, this work builds systems with concepts of time order and rarity in mind. A calibrated forecast provides the expectation and bands residual-based detectors that operate on top of expectations, tuned by thresholds to hit the precision–recall mix that the operator values. The conceptual picture is simple good forecasts reduce background noise, credible bands guide caution, and focused detectors lift the signal when the system departs from its own recent normal.

1.3 Data, Models, and the New Operating Reality

Europe's power systems have been reshaped by three entwined shifts the growth of weather-dependent renewables, deeper cross-border market coupling, and the availability of high-frequency operational data. While interconnection lets neighbours share flexibility, it does not erase regional weather coherence as a North Atlantic front does not consult interconnector schedules before arriving over Ireland (EirGrid 2025); (Lau and McSharry 2010); (Wu, et al. 2022). In this setting, data and machine learning have become operational infrastructure. Quarter-hourly series from EirGrid reports provide the grid's pulse, while minute observations from Met Éireann supply the weather's vocabulary. The task is to make them speak a common time grammar so that models learn from information that would truly have been available at the forecast origin (EirGrid 2025); (Wu, et al. 2022). Aggregating stations into regional composites, using circular statistics for direction and robust means for scalars, reduces noise and guards against single-sensor outages, resampling to the quarter-hour cadence then aligns the

meteorology with the operator's decision rhythm. On top of this spine, leakage-safe features and sequence models produce one-hour-ahead forecasts with intervals that are judged by empirical coverage rather than asserted by model class ((Pei, et al. 2022); (Azam, et al. 2025)). Residual-based anomaly detectors complement the forecasts by flagging departures from expectation rather than merely unusual weather and can lift precision when fed season- and regime-stable residual structure (Xie, et al. 2023); (Xydas, et al. 2017); (Jakub Nowotarski 2018); (Pierre Pinson 2017). The result is a practical, auditable pipeline in which publicly available data support short-horizon intelligence for unit commitment, reserve scheduling and interconnector nominations. In which claims are verified with out-of-sample procedures and standard tests rather than post-hoc adjustment (Florin Bunea 2011).

The process continues with resampling, where the high-frequency, one-minute meteorological data is aggregated into fifteen-minute bins. For this, scalars are averaged, and a circular mean is applied to directional data. Quality checks are essential to ensure that empty bins are explicitly flagged, preventing silent interpolation. In parallel, the quarter-hourly operational series are examined for duplicate timestamps, daylight-saving complications, and out-of-order records. A strictly increasing, evenly spaced time index is constructed and used as the spine onto which all series are projected. Ambiguous times around clock changes are resolved deterministically so that each row corresponds to a unique physical interval. This may look procedural, but it is the basis of trust. Without a clean, shared index, one cannot guarantee that features are computed from information actually available at the time of prediction, nor can one compare methods fairly.

With clocks and geography aligned, feature engineering adds structure that models can use. The first rule is that features must be leakage-safe i.e. they are functions of information that would have existed at the forecast origin. Lags and differences of relevant series provide short-term memory. Rolling means and variances capture local scale and volatility, calendar encodings record hour-of-day and seasonality effects in forms that avoid discontinuities. Simple interactions express relationships known to be present, such as the way residuals behave under high wind availability. These features produce a rich design matrix for Phase A, deliberately wider than what will be carried into the final models. The wideness is purposeful. It helps identify where information lives and where complexity merely hides instability.

The final piece of the data picture is reproducibility. The study avoids private feeds and post-hoc edits. It uses only public, timestamped series that other researchers can obtain. It documents the steps that convert those series into a joint, quarter-hourly dataset with an unambiguous index. It stores long histories in a format that supports streaming, so that training and evaluation can scale without exhausting memory. This discipline is not an end in itself; it is a guard against wishful thinking. When the results are later compared using classical metrics and standard tests such as Diebold-Mariano. There is confidence that differences are not artefacts of how the data were stitched together (Florin Bunea 2011).

1.4 What "forecasting" and "anomaly" mean here?

Forecasting, in this context, means producing a one-hour-ahead prediction of a quarter-hourly target using information that would have been available at the prediction time. The cadence

mirrors the operational rhythm i.e. new information arrives and decisions are taken every fifteen minutes. Crucially, a forecast is not just a point. It is a distributional statement that includes a best estimate and a credible band. For the deep model, that band is obtained by keeping dropout active during inference, running multiple forward passes, and treating the ensemble as an approximate predictive distribution. The method is appealing because it is simple to implement, fast enough for operational cadences, and amenable to empirical verification by coverage. It does not require strong distributional assumptions and, when paired with routine coverage checks, it yields intervals that can be trusted or corrected based on observed behaviour (Florin Bunea 2011). The classical baselines persistence, Autoregressive Integrated Moving Average-Generalized Autoregressive Conditional Heteroskedasticity (ARIMA-GARCH), and simple exponential smoothing are used both for their forecast accuracy and because their uncertainty can be expressed explicitly and tested for coverage rather than presumed (Zangrando, et al. 2022); (EirGrid plc and SONI Limited 2025).

Anomalies are defined relative to this forecasting frame. The detector watches the residual which is the error between expectation and realisation augmented with features that make residual behaviour stable across seasons and regimes. These include recent absolute errors, local variance, regime flags, and interactions with availability measures. The purpose is to let the detector focus on “unexpected given the model” rather than “impressive weather”. The detector family is eclectic on purpose. A σ-threshold rule establishes a baseline, it is transparent, easy to reason about, and sets the floor for recall. Isolation Forest (IF) and One-Class SVM are included as strong, widely used one-class methods that can discover structure in the residual feature space. Deep SVDD stands in for compact deep one-class models that learn a representation of normality and flag points that fall outside it. Thresholds are not afterthoughts they are how the operator expresses appetite for false alarms versus misses. The evaluation regime therefore includes a cross-validation splitter that enforces time order and guarantees a minimum number of anomalies per validation fold. So that precision and recall are measured under credible class balance rather than wishful re-sampling (Xie, et al. 2023); (Xydas, et al. 2017); (Jakub Nowotarski 2018); (Pierre Pinson 2017).

These definitions underpin the two-phase development. Phase A prioritises breadth and transparency. It constructs the joint, cleaned dataset; it runs the three classical baselines with explicit frequency and variance settings, and it trains a straightforward Monte Carlo dropout Long short-term memory (LSTM) to see what the data will bear. The feature set is deliberately large. The aim is to surface signal wherever it resides, to reveal failure modes early, and to put numbers on both accuracy and coverage across seasons. Phase A is not the final word; it is a map of the terrain. It answers questions like: are classical methods competitive at this horizon; does a simple deep model offer a material improvement; and where do intervals hold or fail?

Phase B is a response to what Phase A uncovers. It seeks stability, efficiency, and operational clarity. The feature space is reduced through a three-way funnel, a mutual-information filter retains variables that relate non-linearly to the target. A strong tree-based regressor provides embedded importances that reflect interactions, and recursive feature elimination confirms that removal does not hurt performance. The intersection yields a compact set that travels better across regimes. Data handling is upgraded from in-memory frames to an HDF5-backed

windowed dataset class that streams fixed-length windows and targets without exhausting RAM, giving every sample a traceable origin. Uncertainty estimation is tightened by adopting a clear sampling protocol and verifying nominal coverage routinely. Anomaly detection is refocused score distributions are studied on training data, custom thresholds are selected to hit desired precision-recall trade-offs. Evaluation uses the anomaly-aware splitter so that metrics are stable. The outcome is a system whose components behave more predictably and can be explained without caveats.

1.5 Questions, contributions, and the bounds of the study

The work turns on a small number of practical questions framed in the language of operation. The first is whether simple, transparent benchmarks provide a strong floor for one-hour-ahead forecasting on the Irish system when judged by both point accuracy and interval coverage. This matters because a method that cannot beat persistence consistently, or that achieves accuracy by issuing intervals that are too narrow, may be unfit for purpose. The second is whether a modest sequence model with Monte Carlo dropout can deliver competitive accuracy and credible bands using inputs that would be available in a real setting, without resorting to exotic features or proprietary forecasts. The third is whether framing anomaly detection on residuals and feeding detectors with season- and regime-stable features, allows precision to be improved at high recall. Once the evaluation respects time order and anomaly scarcity, or whether apparent gains in the literature depend on lenient splits. A fourth, cross-cutting question is what is gained by reducing the feature set to a compact core? Does behaviour become more stable across seasons? Does training become faster? And does explanation become easier without sacrificing performance?

The dissertation also aims to make its ideas carry beyond Ireland. The methods are transportable. The need for a clean, strictly increasing time index that survives daylight-saving changes is universal. The use of circular statistics for wind direction is a matter of geometry, not geography. The construction of regional composites, the insistence on leakage-safe features, the adoption of streaming I/O for long histories. And the preference for empirical coverage over untested parametrics are all choices that travel. The residual framing of anomalies and the use of rare-event-aware cross-validation are likewise general. What changes elsewhere are details station lists, market rules, interconnector capacities not the structure of the argument. As European systems deepen interconnection and increase renewable shares, the Irish experience becomes a preview of what others will routinely face (EirGrid 2025); (Wu, et al. 2022).

2. Literature Review of Related Work

The operational toolkit for power systems is undergoing a critical evolution from traditional, dispatchable generation to modern, weather-dependent resources. We explore the sophisticated methods for short-horizon net-load forecasting and residual-based anomaly detection that provide the foundation for this evolving grid. We begin by examining the classical approaches that have long served as the industry standard, including ARIMA-GARCH hybrids, which effectively model both the mean and volatility of time series. These foundational models provide a crucial benchmark against which all more complex methods must be measured. From this solid footing, we explore the promise of deep learning, focusing on architectures like LSTMs and GRUs that can learn complex dependencies from long data sequences. However, a single point forecast is no longer enough. The chapter moves on to address probabilistic forecasting. Detailing methods like Monte Carlo (MC) dropout that can quantify the inherent uncertainty of net-load predictions, providing operators with a powerful tool for risk-aware decisions. The second major theme is anomaly detection, where we explore how the residuals from our forecasts can be transformed into a first-class signal for flagging unusual events. The discussion covers a range of techniques, from simple threshold rules to advanced methods like Deep SVDD. All of which are assessed for their ability to distinguish genuine anomalies from background noise. The review is grounded in the operational realities of the Irish system, providing a framework for the methodological choices made throughout this dissertation.

2.1 Short-horizon forecasting: classical foundations

The practical baseline for short-term load or net-load forecasting remains a set of simple, transparent models that encode persistence and low-order temporal structure. Three families recur in both academic and operator practice.

Persistence (naïve) forecasts. The persistence assumption posits that the best forecast for the next interval equals the most recent observation at that horizon (for example, “+1 hour ahead” equals “value from one hour ago”). Its attraction is ease of deployment and a consistent lower bound on performance in stationary or near-stationary regimes. In quarter-hourly grids, persistence is often implemented as a fixed lag operator that respects daylight-saving changes and public-holiday shifts. Despite its simplicity, persistence often performs competitively for intra-day horizons in stable conditions and remains the benchmark that any learning system should surpass with a comfortable margin (EirGrid 2025); (Wu, et al. 2022).

Exponential smoothing. Simple-exponential smoothing (SES) and its seasonal variants (Holt-Winters) provide a compact way to track level, trend, and seasonality with a small number of parameters. The smoothed level ℓ_t is updated via

Equation 1 SES State Update:

$$\ell_t \leftarrow \alpha y_t + (1-\alpha)\ell_{t-1},\ \alpha \in (0,1),$$

where;

ℓ_t - smoothed level (the SES state) at time t, y_t - observed value (e.g., net-load) at time t, ℓ_{t-1} - previous smoothed level (state at time t−1), α - smoothing parameter in (0,1); larger α makes ℓ_t respond faster to new observations, smaller α smooths more (slower updates).
SES variants are robust baselines for grids that exhibit strong diurnal periodicity and gradual trend drift. In operational settings, their transparency and fast re-estimation appeal when models must be restarted frequently or run on thin compute (Pei, et al. 2022); (Azam, et al. 2025).

ARIMA and GARCH hybrids. Autoregressive integrated moving average (ARIMA) models explain the conditional mean of a differenced series as a linear function of its own history and past shocks. When residual volatility is time-varying, a generalised autoregressive conditional heteroskedasticity (GARCH) layer can be used to forecast conditional variance. The combined approach captures short memory in the mean and volatility clustering in the residuals, which can be valuable around load ramps or in markets with bursty behaviour. In net-load (load minus wind/solar) the heteroskedasticity introduced by weather-driven generation often makes ARIMA-GARCH a more faithful baseline than ARIMA alone (Xie, et al. 2023); (Xydas, et al. 2017); (Jakub Nowotarski 2018).

These baselines matter for two reasons. First, they codify the dominant temporal structures level, trend, diurnal cycles, and volatility against which more complex models must be judged. Second, their residuals become key signals for anomaly detection later in the dissertation.

2.1.1 Error metrics and residuals

All forecasting models in this study are evaluated on a common set of metrics that quantify typical error size and dispersion. Let y_t denote the observed net-load and $\hat{y}_t$ the forecast at time t. The residual is $e_t = y_t - \hat{y}_t$. Two scale-dependent scores are used:

Equation 2 Definitions of MAE and RMSE for Forecast Evaluation

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|e_t| \text{ , } RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$$

where;

n - number of forecast actual pairs (evaluation samples), t - time index of the samples in the evaluation window, y_t - observed value (e.g., net-load) at time t, $\hat{y}_t$ - forecast issued for time t, $e_t = y_t - \hat{y}_t$ - residual (forecast error) at time t.
Root mean squared error (RMSE) accentuates large errors and is commonly reported by system operators for operational comparability while Mean Absolute Error (MAE) provides a more median-like sense of error magnitude that is less sensitive to outliers (Pei, et al. 2022); (Azam, et al. 2025). Where probabilistic forecasts are produced, empirical coverage of predictive intervals is also tracked (Section 2.3). The residual series is both a diagnostic of model fit and a first-class signal for anomaly detection, under the assumption that significant departures from the modelled conditional mean/variance represent unusual operating conditions or data artefacts.

2.2 Probabilistic forecasting and uncertainty quantification

Deterministic point forecasts hide uncertainty that is intrinsic to weather-coupled net-load. Operational dispatch and security-of-supply decisions benefit from a distributional view of the future rather than a single number. Two practical questions dominate: how to produce calibrated uncertainty bands and how to evaluate them.

Predictive intervals and coverage. A pragmatic way to expose uncertainty is to estimate central prediction intervals $[L_t^{(q)}, U_t^{(q)}]$ at nominal level q (e.g., 80% or 95%). Coverage is the fraction of times the observed value falls inside the interval:

Equation 3 Coverage Metric for Probabilistic Forecasts

$$\text{Coverage}(q) = \frac{1}{n}\sum_{t=1}^{n} 1\left\{y_t \in \left[L_t^{(q)}, U_t^{(q)}\right]\right\}.$$

where;

q - nominal coverage level (e.g., 0.800 or 0.95) for a central prediction interval, n - number of forecast–actual pairs evaluated, t - time index over the evaluation window, y_t - observed value (e.g., net-load) at time t, $\left[L_t^{(q)}, U_t^{(q)}\right]$ the model's central q - level prediction interval for time t (lower and upper bounds), Coverage(q) - empirical fraction of observations that fall inside their q - level intervals so a calibrated forecaster yields Coverage(q) ≈ q.
Well-calibrated intervals satisfy Coverage(q) ≈ q, while unnecessarily wide intervals may be over-conservative. For grid operations, calibrated but sharp intervals allow risk-aware scheduling and reserve allocation (Pei, et al. 2022) (Florin Bunea 2011).

Uncertainty via stochastic regularisation (MC dropout). In deep sequence models, Monte Carlo (MC) dropout at test time running the network multiple times with dropout active approximates Bayesian model averaging and yields a distribution of predictions. The empirical mean provides the point forecast; the sample variance produces uncertainty bands. While not a full Bayesian treatment, MC dropout scales well and is widely adopted in practical systems where heavy probabilistic machinery is infeasible (Zangrando, et al. 2022) (EirGrid plc and SONI Limited 2025) (Florin Bunea 2011). Later chapters operationalise this idea and evaluate its calibration against coverage targets.

Variance modelling in classical baselines. ARIMA-GARCH naturally produces a conditional variance forecast $\hat{\sigma}_t^2$. Under approximate normality of residuals, one can form predictive intervals $\hat{y}_t \pm z_q\hat{\sigma}_t$, where z_q is the appropriate quantile. In practice, departures from normality and residual autocorrelation can degrade calibration; nevertheless, these intervals create a useful reference for deep models' uncertainty claims (Xydas, et al. 2017); (Jakub Nowotarski 2018).

2.3 Deep learning for short-term net-load forecasting

Modern load forecasting increasingly exploits deep sequence models that can synthesise long-range temporal patterns and multi-modal exogenous features (meteorology, calendars, operational flags). Several architectures appear repeatedly in the literature.

Recurrent neural networks (RNNs) and LSTMs. Long short-term memory networks are recurrent architectures that mitigate vanishing gradients via gated memory cells. They are attractive for quarter-hourly or hourly horizons because they learn diurnal and weekly recurrences without explicit manual lag engineering. Stacking layers and using residual connections can improve capacity to model complex non-linearities. For operational use, careful regularisation, early stopping, and normalisation are essential to avoid overfitting on long histories (Zangrando, et al. 2022); (EirGrid plc and SONI Limited 2025); (Florin Bunea 2011). RNNs are the simplest class of deep sequence models considered in this dissertation and remain a defensible option for short-horizon net-load forecasting. A vanilla RNN updates a hidden state with each quarter-hourly step, allowing the model to summarise recent history and fuse exogenous inputs (e.g., wind availability, calendar flags) into the evolving state. In practice, plain RNNs struggle with long dependencies due to vanishing or exploding gradients. So, gated variants like the gated recurrent units (GRUs) and long short-term memory (LSTM) networks mitigate this by learning when to retain or forget information. GRUs offer a lighter-weight alternative that can train faster with comparable accuracy on intra-day horizons, whereas LSTMs tend to be more stable as sequence length and feature breadth increase. Both can be regularised with dropout, normalisation, and early stopping, and both integrate naturally with probabilistic layers (e.g., MC-dropout) to yield calibrated intervals. Given these trade-offs, vanilla RNNs serve as a conceptual baseline, while LSTMs (and GRUs in ablation) provide the operationally stronger recurrent choices used here. (Zangrando, et al. 2022); (EirGrid plc and SONI Limited 2025); (Florin Bunea 2011)

Temporal convolutions and hybrid models. Temporal convolutional networks (TCNs) with dilated convolutions offer large receptive fields with fewer parameters than deep LSTMs. Hybrid models that blend convolutional feature extractors with recurrent decoders, or incorporate attention mechanisms, have shown strong results on volatile net-load when exogenous weather features dominate the signal (Pei, et al. 2022) (Azam, et al. 2025) (Florin Bunea 2011).

Feature learning vs. feature engineering. While deep models can, in principle, learn useful features from raw inputs. Power-system datasets typically benefit from domain-aware engineered covariates diurnal cycles, holiday flags, wind direction encodings, and rolling statistics of forecast errors. Hybridising deep learners with a compact, vetted feature set often yields better generalisation, especially when data are limited or regimes shift (Pei, et al. 2022); (Azam, et al. 2025); (Florin Bunea 2011). This observation motivates the feature-selection funnel used later in the dissertation to reduce the model's input space without losing key meteorological and residual cues.

Regularisation and uncertainty. MC dropout, weight decay, and early stopping are the main defences against overfitting. In addition, windowed training with rolling-origin validation better exposes regime changes and seasonality shifts; it is more faithful to online deployment than random splits. The combination of windowed evaluation and MC-dropout uncertainty is now common in operationally-minded studies (Zangrando, et al. 2022) (EirGrid plc and SONI Limited 2025) (Florin Bunea 2011).

2.4 Feature engineering for grid-aware forecasting

Three categories of features recur in reliable short-horizon net-load models.

Calendar and cyclic encodings. Hour-of-day, day-of-week, month-of-year, and public-holiday flags capture recurring demand cycles. To avoid artificial discontinuities (e.g., 23:00 next to 00:00), angles on the unit circle are used: for an hour h ∈ {0,…,23},

$$\sin\left(\frac{2\pi h}{24}\right),\ \cos\left(\frac{2\pi h}{24}\right).$$

Such encodings help linear and non-linear models alike represent periodic structure smoothly (Pei, et al. 2022); (Azam, et al. 2025).

Meteorology and wind integration. When studying net-load in a high-wind system, meteorological inputs are crucial. Wind speed and direction, temperature, and humidity influence both demand and embedded generation. Direction is circular and should be encoded accordingly (e.g., via sin and cos of the angle, or circular means for aggregation). Station-level data often arrive at one-minute resolution and must be resampled to match grid cadence; neighbour-based imputation stabilises gaps (see Appendix A for the regional 15-minute pipeline). Region-aggregated features mitigate local noise while retaining large-scale weather dynamics relevant to wind generation (EirGrid 2025); (Lau and McSharry 2010); (Wu, et al. 2022).

Residual and interaction features. Forecast errors from simple baselines, their lags and transformations (absolute error, squared error, normalised percent change), and interactions between residuals from different baselines frequently improve anomaly sensitivity and net-load forecasts alike. For instance, if exponential smoothing and ARIMA disagree in sign or magnitude, the discrepancy can signal an impending ramp or data problem. Rolling measures of residual variance (GARCH-style proxies) mark transient heteroskedastic regimes. The literature increasingly endorses residual-aware features both for forecasting and for downstream anomaly detection (Xie, et al. 2023); (Xydas, et al. 2017); (Jakub Nowotarski 2018); (Pierre Pinson 2017).

Guarding against leakage. Care is required to ensure that features are computable using only information available up to forecast time. Rolling windows must be strictly backward-looking i.e. data aligned across sources must honour daylight-saving transitions and late-arriving telemetry updates. The time-index regularisation strategy documented in Appendix B is designed expressly to avoid these pitfalls in deployment.

2.5 Anomaly detection in power systems: from rules to one-class learning

Anomalies in operational grid data encompass sensor faults, data feed errors, and genuine but rare system behaviours (e.g., extreme ramps, curtailment, or contingencies). In a forecasting-centric view, an anomaly is a time point where the realised outcome is implausible given the model and its uncertainty. The literature spans simple rules to modern one-class and deep one-class methods; the common thread is that residuals differences between observed and expected are often the most informative representation (Xie, et al. 2023); (Xydas, et al. 2017); (Jakub Nowotarski 2018); (Pierre Pinson 2017).

Threshold rules on residuals. A standard baseline is to flag a point as anomalous if $|e_t|$ exceeds k times a scale estimate σ^ derived from training residuals (e.g., median absolute deviation or standard deviation). Formally, flag t if $|e_t| > k\sigma^$. Choosing k trades recall (sensitivity) against precision (false-positive control). Season-aware thresholds, or thresholds conditioned on recent volatility, improve stability by acknowledging that σ^ drifts across regimes (Xie, et al. 2023); (Xydas, et al. 2017).

IF is an ensemble of random partition trees; anomalies are isolated quickly because few splits suffice to single them out. It is attractive for its minimal tuning and speed, but it treats features independently within splits, which can under-utilise correlation structure in residual and volatility features. IF often serves as a useful first non-parametric comparator (Pierre Pinson 2017).

OC-SVM learns a boundary that encloses the bulk of the data in a high-dimensional feature space spanned by a kernel ϕ. Given training samples $\{x_i\}^n_{i=1}$, it solves

Equation 4 OC-SVM Optimization for Learning the Normality Boundary

$$\min_{\omega,\rho,\varepsilon} \frac{1}{2}\left|\left|\omega\right|\right|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}(\varepsilon_i - \rho) \text{ s.t. } \omega^T\emptyset(x_I) \geq \rho - \varepsilon_I, \varepsilon_I \geq 0,$$

where; x_i - the i - th training sample in input space, i = 1,…,n, n - number of training samples, $\phi(\cdot)$ - feature map into a (possibly high-dimensional) Hilbert space; used implicitly via a kernel $K(x , x') = \phi(x)^\top\phi(x')$, ω - weight vector in feature space that defines the separating (acceptance) surface, ρ - offset (threshold) that sets how "tight" the acceptance region is around normal data, ε_i - non-negative slack variables allowing violations (points outside the acceptance region), $\nu\in(0,1]$ - trade-off parameter: upper bound on the fraction of training outliers and lower bound on the fraction of support vectors, Objective $\frac{1}{2}\|\boldsymbol{\omega}\|^2$ - encourages a large margin (simpler boundary), Penalty $\frac{1}{\nu n}\sum \varepsilon_i$ - penalises violations that is to say larger *v* increases tolerance for outliers, Term - ρ pushes ρ upward, expanding the accepted region only as much as the constraints allow, Constraint $\omega^\top\phi(x_i) \geq \rho - \varepsilon_i$ - most mapped points must lie inside the acceptance region and only a *v* controlled fraction may violate it via $\varepsilon_i > 0$, Decision function (at test time): $f(x)=\omega^\top\phi(x) - \rho$. Predict normal if $f(x) \geq 0$, anomalous if $f(x) < 0$.

OC-SVM's strength lies in its ability to carve out smooth, non-linear acceptance regions that reflect interactions among residual, volatility, and calendar features (Xie, et al. 2023); (Jakub Nowotarski 2018).

Deep SVDD embeds inputs through a neural network $f_\theta(\cdot)$ and pulls normal examples towards a fixed centre c in latent space:

Equation 5 Deep SVDD Training Loss and Test-Time Anomaly Score

$$\min_{\theta} \frac{1}{n}\sum_{i=1}^{n}\left|\left|f_{\theta}(x_i) - c\right|\right|^2 + \lambda\sum_{\ell}\left|\left|\theta_{\ell}\right|\right|^2$$

where;

x_i - the i - th training sample (feature vector), i=1,…,n, n - number of training samples, fθ(·) - neural network (with parameters θ) that embeds inputs into a latent space, c - fixed centre of the "normal" region in latent space (often set once from an initial forward pass on training data, excluding outliers), $\|f_{\theta}(x_i) - c\|^2$ - squared Euclidean distance of the embedded point to the centre; smaller means "more normal.", θ = $\{\theta_{\ell}\}$ - all network parameters and θ_{ℓ} are layer-wise parameter blocks, $\lambda\sum_{\ell}\|\theta_{\ell}\|^2$ - L_2 weight decay (regularisation) to discourage overfitting and trivial embeddings, Anomaly score (at test time) - s(x) = $\|f_{\theta}(x_i) - c\|^2$. Flag as anomalous when s(x) exceeds a chosen threshold (e.g., a high quantile of training scores).

By learning representations shaped for one-class separation, Deep SVDD can improve precision over OC-SVM when informative non-linearities exist, provided the feature set is regime-stable and compact (Xydas, et al. 2017) (Pierre Pinson 2017).

Why residual-aware features help. Across rule-based, kernel, and deep one-class approaches, the consensus is that residuals from calibrated forecasts concentrate 'normal' behaviour around zero and magnify departures that matter operationally. Conditioning on local variance (e.g., via rolling GARCH-like estimates) further reduces false alarms in naturally volatile periods. Several studies of energy-sector anomaly detection argue that univariate residual thresholds set a floor on recall, while one-class methods materially lift precision when fed stable residual and volatility features (Xie, et al. 2023) (Xydas, et al. 2017) (Jakub Nowotarski 2018) (Pierre Pinson 2017).

2.5.1 Evaluation for rare events

Imbalanced anomaly detection requires metrics that respect skewed class proportions.

Let TP, FP, FN, TN denote true positives, false positives, false negatives, and true negatives. Precision, Recall, and F1 are defined as

Equation 6 Precision, Recall, and F1

$$\text{Precision} = \frac{TP}{TP + FP}, \text{Recall} = \frac{TP}{TP + FN}, \quad F1 = 2\cdot\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision} + \text{Recall}}.$$

The False-Alarm Rate (FAR) reported in this work is the proportion of normal points incorrectly flagged:

Equation 7 False-Alarm Rate

$$\text{FAR} = \frac{FP}{FP + TN}.$$

Given the operational costs of investigating false alarms, FAR is often weighed at least as heavily as recall. The literature stresses careful cross-validation that preserves time order and ensures a minimum number of anomalies in validation folds; random cross-validation inflates recall and understates FAR because it leaks future information and breaks temporal clustering

(Xie, et al. 2023) (Xydas, et al. 2017) (Pierre Pinson 2017). The anomaly-safe splitter used later addresses these issues directly.

2.6 Evaluation methodology for forecasting

Fair model comparison in time series requires rolling-origin (walk-forward) validation rather than random splits. Each training fold precedes its validation fold in time the models are re-estimated before each validation window, mirroring deployment. This approach is standard in both operator practice and academic studies focused on operational readiness (Pei, et al. 2022); (Azam, et al. 2025); (Florin Bunea 2011).

Diebold-Mariano (DM) test. When comparing forecast accuracy between two methods, the DM test evaluates whether the mean difference in a loss series (e.g., squared error) is significantly different from zero. Let $d_t = L\left(e_t^{(1)}\right) - L\left(e_t^{(2)}\right)$ denote the period-by-period loss differential. The test statistic is:

Equation 8 Diebold-Mariano (DM)

$$DM = \frac{\bar{d}}{\sqrt{\hat{\sigma}^2_{\bar{d}}}},$$

where $\bar{d}$ is the sample mean of $\{d_t\}$ and $\hat{\sigma}^2_{\bar{d}}$ is a heteroskedasticity and autocorrelation-consistent (HAC) estimate of its variance. A large positive (negative) statistic suggests that model 1 is worse (better) than model 2 in the chosen loss metric. The DM test is widely recommended for paired forecast comparisons in energy systems (Pei, et al. 2022).

Calibration checks. For probabilistic forecasts, beyond empirical coverage, sharpness (narrowness of intervals) and reliability diagrams (observed vs nominal coverage across bins) are encouraged. Although operational studies often default to coverage alone for simplicity, the literature warns that coverage can be gamed by widening intervals; hence the need to present coverage together with point-error metrics and, where possible, scoring rules such as the interval score or Continuous Ranked Probability Score (CRPS) (Pei, et al. 2022); (Florin Bunea 2011).

2.7 System characteristics that shape modelling

The Irish synchronous area is characterised by high wind penetration, relatively small system size compared with continental Europe, and stringent operational constraints due to inertia and interconnection limits. These factors make net-load particularly weather-coupled and heteroskedastic at short horizons. EirGrid's operational publications and data portals emphasise quarter-hourly balancing, curtailment episodes during high wind, and demand spikes around daily routines and seasonal events (EirGrid 2025); (Lau and McSharry 2010); (Wu, et al. 2022). Such characteristics reinforce several design choices in the present work.

First, meteorological fidelity matters i.e. regional aggregates of wind speed, direction, and temperature provide more robust signals than single-station feeds, especially when station maintenance or coastal microclimates distort local readings. Second, time-index integrity strict 15-minute cadence and careful handling of daylight-saving transitions is not optional; many

studies document spurious anomalies and forecast artefacts traceable to index drift or duplicated timestamps (EirGrid 2025); (Lau and McSharry 2010); (Wu, et al. 2022). Third, periods of high non-synchronous penetration and curtailment can create residual distributions that are heavier-tailed than Gaussian; methods that adapt thresholds or model variance dynamically tend to be more reliable (Xie, et al. 2023); (Xydas, et al. 2017); (Jakub Nowotarski 2018); (Pierre Pinson 2017).

Against this backdrop, the literature in European contexts highlights a pattern mainly that classical baselines remain hard to beat in average error during placid conditions, but learning models show advantages around ramps and when multi-modal features (weather plus calendar) interact. Uncertainty-aware deep models are particularly valuable when operators require not just a point forecast but also a bound on expected variation to schedule reserves cost-effectively (Pei, et al. 2022); (Florin Bunea 2011).

3. Analytical Background and Requirements Analysis

To build a robust and reliable system for a modern power grid, a clear analytical framework is essential. This chapter provides that framework, starting with a formal definition of the two primary tasks: short-horizon net-load forecasting and residual-based anomaly detection. We first set the stage by detailing the specific data requirements and pre-processing steps needed to prepare time-series data at a 15-minute cadence. This includes the meticulous process of time-index regularisation to handle issues like daylight-saving changes. This foundation ensures that our data is clean and ready for analysis, a non-negotiable prerequisite for trustworthy results. Building on this, the chapter explores the modelling approaches for each phase of the project. For forecasting, it outlines the use of classical models as a benchmark and a source of key features, while also introducing a deep sequence model to capture more complex patterns. For anomaly detection, it lays out a tiered approach. Moving from simple rules to more sophisticated kernel and deep learning methods that can better distinguish genuine events from noise. The chapter concludes by translating these analytical and modelling choices into a detailed set of system requirements. Covering everything from data integrity to operational constraints, which will serve as a blueprint for the implementation phase.

3.1 Formal problem statement

3.1.1 Net-load forecasting at 15-minute cadence

Let $y_t \in R$ denote the net-load (system demand minus embedded wind/solar generation when applicable) measured on a uniform 15-minute grid indexed by integer time $t \in Z$. Let $x_t \in R^p$ be a vector of exogenous covariates available at time t: calendar encodings (hour, day, holiday), regional meteorology (temperature, wind speed/direction), and engineered residual/volatility features derived from baselines. For a fixed forecast horizon $h \in \{1,2,\ldots,H\}$ (with h = 4 corresponding to one hour ahead on a 15-minute grid), the forecasting task is to learn a function $f_h : R^{w\times(1+p)} \rightarrow R$ mapping a rolling window of the most recent www observations and features to a point forecast:

$$\hat{y}_{t+h} = f_h(W_t),\ W_t := \{(y_{t-w+1}, x_{t-w+1}), \ldots, (y_t, x_t)\}.$$

To quantify uncertainty, we report central prediction intervals at nominal levels $q \in \{0.80,0.95\}$. An interval is considered calibrated when its observed in Equation 3 at horizon h matches the nominal level, computed as:

$$\text{Coverage}(q) = \frac{1}{N}\sum_{t=1}^{N} 1\left\{y_{t+h} \in \left[L_{t+h}^{(q)} + U_{t+h}^{(q)}\right]\right\} \approx q.$$

Point-forecast accuracy is summarised by the RMSE a horizon-specific restatement of Equation 2 given by:

$$\text{RMSE} = \frac{1}{N}\sum_{t=1}^{N}\left(y_{t+h} + y_{t+h}\right)^2.$$

These measures are standard in operator practice and allow comparison with persistence and other transparent baselines ((Pei, et al. 2022); (Azam, et al. 2025)).

3.1.2 Residual-based anomaly detection

Define the forecast residual at time t for horizon h as $e_{t,h} = y_t - \hat{y}_{t|t-h}$, where $\hat{y}_{t|t-h}$ is the forecast made at t − h for delivery at t. Under a calibrated forecasting model, $e_{t,h}$ concentrates around zero with dispersion reflecting conditional variance. We treat an anomaly as a time point t whose observed behaviour is implausible given the model and recent regime, operationally indicated by the residual and volatility-aware features leaving the "normal" region learned from historical data. Formally, given a feature vector $z_t \in R^q$ constructed from residuals, residual lags, local variance proxies, and calendar context, an anomaly detector produces a binary decision $\hat{a}_t \in \{0,1\}$ and possibly a score $s_t \in R$ such that larger s_t indicates greater atypicality. Evaluation uses precision, recall, F1 , and the false-alarm rate (FAR). Because anomalies are rare and often clustered, time-respecting cross-validation and carefully chosen decision thresholds are crucial to avoid inflated recall and understated FAR (Xie, et al. 2023); (Xydas, et al. 2017); (Pierre Pinson 2017). The anomaly-safe splitter referenced later (Appendix D) encodes these constraints.

3.2 Data and pre-processing assumptions

The analytical setup relies on two families of time-stamped data: grid net-load at quarter-hourly cadence from EirGrid reports and station-level meteorology at minute-level cadence from Met Éireann. To avoid leakage and spurious artefacts, the time index must be strictly regular and the meteorology must be aggregated in a way that respects circular quantities and regional coherence.

The first pre-condition is time-index regularisation, the grid time series is re-indexed onto a clean, gap-free 15-minute grid, resolving duplicate timestamps and daylight-saving transitions deterministically. Where duplicates exist, a helper index is created, duplicates are collapsed under a new strictly increasing DateTime index, and the main DataFrame is re-attached to that clean index. This process, and the integrity checks that accompany it, are documented in Appendix B. It is central to everything that follows: without a trustworthy index, rolling windows, back-looking features, and train/validation splits become ill-defined.

The second pre-condition is regional meteorology at a matching cadence. Minute-resolution station files are first quality-checked and repaired using neighbour-based imputation and circular means for wind direction, then resampled to 15-minute bins using arithmetic means for scalars and circular means for directions. Station series are then aligned on their intersection of timestamps and aggregated to regional series (NW, NE, SW, SE) using arithmetic means for temperature and wind speed and circular means for direction. This pipeline is recorded in Appendix A. Using regional rather than single-station features mitigates local noise and maintenance gaps while retaining the synoptic dynamics that shape wind generation (EirGrid 2025); (Lau and McSharry 2010); (Wu, et al. 2022).

To ensure leakage-free features, all rolling statistics (e.g., one-hour means, residual lags, and volatility proxies) are computed using strictly past data up to time t. Circular variables are encoded with sine and cosine or aggregated via circular means. The circular mean of a set of directions $\{\theta_i\}$ in degrees is defined by;

Equation 9 Directional Averaging via Sine-Cosine Components

$$\bar{\theta} = \operatorname{atan2}\left(\frac{1}{n}\sum_i \sin\theta_I \ , \ \frac{1}{n}\sum_i \cos\theta_I\right),$$

with care to return angles in the desired range. Cyclical calendar encodings (e.g., hour-of-day) use the unit circle:

Equation 10 Sine-Cosine Basis for 24-Hour Periodicity

$$\text{hour\_sin}(t) = \sin(2\pi\cdot\text{hour}(t)/24),\ \text{hour\_cos}(t) = \cos(2\pi\cdot\text{hour}(t)/24).$$

These encodings remove artificial discontinuities and improve the smoothness of the function class a model must learn (Pei, et al. 2022) (Azam, et al. 2025).

Finally, to support efficient model training and reproducibility over multi-year data, windowed datasets are materialised over HDF5 class. A dataset class serves fixed-length input windows and aligned targets from an HDF5 store with chunking that matches the window length. This approach avoids loading the full time series into memory and allows identical sampling across runs. The design is specified in Appendix C.

3.3 Modelling background for Phase A (forecasting)

The forecasting phase combines transparent classical baselines and a deep sequence model with stochastic regularisation to deliver both point forecasts and pragmatic uncertainty intervals.

Persistence and exponential smoothing. Persistence is implemented as a lag operator matching the forecast horizon; simple-exponential smoothing and its Holt–Winters variants update level/trend/season online as data arrive. In this context, SES is valuable not merely as a benchmark but as a source of residuals whose properties (e.g., mean-reversion) can assist downstream detection. The smoothed level update is augmented when a seasonal component is warranted. Hyper-parameters are estimated on pre-2023 data and checked for stability across seasons (Pei, et al. 2022); (Azam, et al. 2025).

Autoregressive integrated moving-average models target the conditional mean of the differenced net-load, while a GARCH layer models the conditional variance of the residuals:

Equation 11 Mean-Variance Model: ARIMA for mu_t with GARCH for sigma_t

$$y_t = \mu_t + \varepsilon_t,\ \mu_t = \phi(B)y_t + \theta(B)\varepsilon_t,\ \varepsilon_t \sim N(0,\sigma_t^2),$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{P} a_i \varepsilon_{t-1}^2 + \sum_{j=1}^{Q} \beta_j \sigma_{t-j}^2.$$

where;

Definitions (GARCH variance):

σ_t^2 - conditional variance of the residual at time t (uncertainty around the mean forecast), ω - constant (long-run variance level), typically $\omega > 0$, α_i - ARCH coefficients on past squared shocks ε_{t-i}^2 (how much recent surprises inflate today's variance), i = 1,…,P, usually $\alpha_i \geq 0$, β_j - GARCH coefficients on past conditional variances σ_{t-j}^2 (persistence of volatility), j = 1,…,Q,

usually $\beta_j \geq 0$, ε_t - one-step innovation/residual ($y_t - \mu_t$), P,Q - orders of the ARCH and GARCH components respectively.

Definitions (ARIMA mean)

μ_t - conditional mean of y_t, $\phi(B)$,$\theta(B)$ - AR and MA lag polynomials in the backshift operator B (with $B^k\, y_t \equiv y_t - k$).

In the Equation 11, β is the backshift operator. Even when normality is only approximate, the conditional variance σ_t^2 provides a principled scale for residual-based thresholds and aids interval formation ((Xydas, et al. 2017); (Jakub Nowotarski 2018)).

MC-Dropout LSTM. A stacked long short-term memory network ingests windowed sequences W_t and outputs point forecasts. To express uncertainty without heavyweight Bayesian inference, dropout remains active at test time; multiple forward passes $\left\{ y \begin{matrix} (m) \\ t+h \end{matrix} \right\}_{m=1}^{M}$ yield an empirical predictive distribution:

Equation 12 MC-Dropout Predictive Mean and Variance (Ensemble Estimators)

$$\mu_{t+h} = \frac{1}{M} \sum_{m=1}^{M} y \begin{matrix} (m) \\ t+h \end{matrix}, \quad \sigma \begin{matrix} 2 \\ t+h \end{matrix} = \frac{1}{M-1} \sum_{m=1}^{M} \left(y \begin{matrix} (m) \\ t+h \end{matrix} - \mu_{t+h} \right)^2 .$$

where;

t - forecast origin time (index on the 15-minute grid), h - forecast horizon in steps (e.g., h=4 = 1 hour ahead on a 15-minute grid), M - number of Monte-Carlo forward passes with dropout kept active at test time, $y \begin{matrix} (m) \\ t+h \end{matrix}$ - the m-th stochastic prediction for delivery at t + h from the same trained network evaluated with a different dropout mask (same weights θ, different mask), μ_{t+h} - predictive mean across the M stochastic predictions; used as the point forecast, $\sigma \begin{matrix} 2 \\ t+h \end{matrix}$ - predictive variance estimated as the (Bessel-corrected) sample variance of the M predictions. Assuming approximate symmetry, central intervals are formed as $[\hat{\mu}_{t+h} \pm z_q \hat{\sigma}_{t+h}]$ with z_q the standard-normal quantile for nominal level q. While heuristic, this approach is widely used in practice and scales gracefully (Zangrando, et al. 2022); (EirGrid plc and SONI Limited 2025); (Florin Bunea 2011). Practicalities normalising features per fold, early stopping, and seed control are embedded in the HDF5 windowed pipeline (Appendix C).

Feature parsimony. Although deep models can assimilate many inputs, the literature and preliminary experiments suggest that a compact, robust set of features often generalises better across seasons and years. The analytical stance here is to admit three feature families i.e. calendar/cyclical encodings, regional meteorology (temperature, wind speed/direction encodings), and residual/volatility features from classical baselines (e.g., lagged residuals, absolute and squared errors, rolling variance proxies). Feature selection via mutual information and model-based importance serves to remove unstable or redundant predictors before deep training, which simplifies learning and reduces variance (Pei, et al. 2022); (Azam, et al. 2025); (Florin Bunea 2011).

Evaluation. Rolling-origin (walk-forward) validation is the default. Each fold trains on a growing history and validates on the next contiguous block, mirroring operational deployment. Where two models need to be compared formally, the Diebold-Mariano test on squared-error differences is used with heteroskedasticity- and autocorrelation-consistent variance estimates (Pei, et al. 2022). This guards against over-interpreting small differences on autocorrelated errors.

3.4 Modelling background for Phase B (anomaly detection)

The anomaly phase consumes residuals and regime descriptors from Phase A and applies a sequence of detectors of increasing sophistication, from univariate thresholds to kernel and deep one-class methods.

Sigma-threshold baseline. A residual threshold sets an acceptance band based on the scale of training residuals. Using the training set residuals $\{e_t\}$, estimate $\hat{\sigma}$ by a robust measure (standard deviation after outlier trimming or median absolute deviation scaled to σ). Flag a point anomalous if $|e_t| > k\hat{\sigma}$ for chosen k. Season-aware variants allow $\hat{\sigma}$ to vary across calendar buckets (e.g., hour-of-day, winter vs summer). The baseline is simple, transparent, and establishes a recall floor, but it can over-flag during high-volatility regimes unless k is increased, which then hurts recall (Xie, et al. 2023); (Xydas, et al. 2017).

Isolation Forest. The ensemble isolates atypical points via random partitions of the feature space; the path length required to isolate a point forms an anomaly score. While tuning is light, the method treats splits univariately at each node and can miss correlated structure between residuals and volatility proxies. IF nonetheless offers a quick non-parametric check against threshold rules (Pierre Pinson 2017).

OC-SVM learns a boundary around normal data in a kernel-defined feature space. Given mapping $\phi(\cdot)$, the primal optimisation balances boundary tightness against an upper bound ν on the fraction of training outliers. An RBF kernel typically works well for residual-centric features because it can represent smooth acceptance regions that bend around non-linear correlations among residual lags and local variance. OC-SVM is attractive when one wants a controllable trade-off between tolerance (ν) and margin smoothness (γ in the RBF kernel) (Xie, et al. 2023); (Jakub Nowotarski 2018).

Deep SVDD seeks an embedding $f_\theta(z)$ that collapses normal samples near a centre c while keeping the network simple to discourage trivial solutions. At test time, the squared distance becomes the anomaly score. Thresholds can be set on d_t by quantiles of the training score distribution (e.g., 95th), optionally adjusted to hit a target recall on a development set. Deep SVDD can improve precision when the input feature set is compact and regime-stable, which is why the feature-selection funnel is re-used here (Xydas, et al. 2017); (Pierre Pinson 2017).

Decision thresholds and lexicographic priorities. In operations we treat recall as a must-have. We therefore set a recall floor and then pick the threshold that gives the best precision. Formally, for any detector score and threshold τ,

Equation 13 Lexicographic Threshold Selection

$$\tau^{\backslash *} = \arg\max \text{Precision}(\tau) \text{ s.t. Recall}(\tau) \geq \tau^{\backslash *}$$

(with $\tau^{\backslash *}$ e.g. 0.90, for this study $\tau^{\backslash *}$ *was set to* 0.70). This two-step rule works cleanly for OC-SVM decision values and Deep-SVDD distances: first keep thresholds that meet the recall floor, then choose the one with the highest precision (equivalently, the lowest FAR among ties).

Validation with scarce anomalies. Random K-fold validation is inappropriate for time series and worse for rare-event detection because it leaks future information and smears clusters of anomalies across folds. The anomaly-safe splitter (Appendix D) enforces temporal order and a minimum number of anomalies per validation fold. Thus, preserving the local correlation structure of anomalies and stabilising estimates of recall and FAR. This design follows the guidance found in recent residual-based detection studies in energy systems (Xie, et al. 2023); (Xydas, et al. 2017); (Pierre Pinson 2017).

3.5 Requirements analysis

The following requirements flow from the analytical background and the operational context of quarter-hourly forecasting and detection on Irish data. They are grouped by theme; where helpful, the related appendix is noted to keep implementation details accessible without overloading the main text.

3.5.1 Functional requirements

The system must produce quarter-hourly net-load forecasts for short horizons with calibrated uncertainty, and it must emit anomaly decisions based on those forecasts' residuals. Forecasts must be computed strictly from information available at the time of issue, including aligned regional meteorology and calendar features. Anomaly flags must be reproducible from a documented feature set and a stable decision rule. In deployment terms, this means an end-to-end pipeline index regularisation, meteorology aggregation, feature construction, forecasting, residual computation, and detection executed in a fixed sequence with no human-in-the-loop edits between stages.

3.5.2 Data requirements

A clean, gap-free 15-minute index over the study period is non-negotiable. The grid net-load series must have no duplicated and re-indexed onto an evenly spaced DateTime index that is stable across daylight-saving time. All derived features must be functions of the data up to their timestamp only. Meteorological inputs must be aggregated from minute-level station CSVs into regional quarter-hourly series using arithmetic means for scalars and circular means for directions, with neighbour-based imputation for gaps. Any external holiday calendars or policy flags must be treated as exogenous at the time scale of interest and encoded without look-ahead. These conditions ensure that cross-validation and live evaluation are commensurate. (simple word for commensurate)

3.5.3 Modelling requirements (Phase A)

Classical baselines namely, persistence, exponential smoothing, and ARIMA-GARCH must be trained and evaluated on the same folds as the deep model. Their residuals and, where available,

conditional variance estimates must be retained as first-class features for later use. The deep forecaster must operate on fixed-length windows served from an HDF5-backed dataset, with deterministic sampling, normalisation within fold, and seed control to stabilise training. Uncertainty must be estimated via MC dropout at test time with a documented number of samples; empirical coverage at 80% and 95% must be reported for each fold and hold-out. Feature selection should narrow inputs to a parsimony set that remains stable across seasons, reducing variance and improving interpretability. All hyper-parameters and training curves must be logged for later analysis (Appendix C; Appendix E).

3.5.4 Modelling requirements (Phase B)

Anomaly detectors must be trained on residual-centric feature sets that include direct residuals from at least one classical baseline and the deep model, residual lags, interactions (e.g., products of residuals), and volatility proxies (rolling variances or GARCH-style estimates). The suite must include a univariate sigma-threshold rule, Isolation Forest, OC-SVM, and Deep SVDD. For any detector that outputs a continuous score, thresholds must be chosen by a lexicographic rule that enforces a recall floor on validation before optimising precision. Cross-validation must use the anomaly-safe splitter to guarantee a minimum number of anomalies per validation fold, maintain temporal order, and avoid leakage. For transparency, false-alarm rate must be reported alongside precision, recall, F1 on both cross-validation and hold-out (Appendix D).

3.5.5 Evaluation requirements

Forecast accuracy is reported per fold and on hold-out using RMSE and empirical interval coverage at 80% and 95%. When two methods are close, a Diebold-Mariano test on squared-error differences is applied with HAC variance (Xie, et al. 2023). In addition, residuals are checked for conditional heteroskedasticity using a rolling Engle Autoregressive Conditional Heteroskedasticity Lagrange Multiplier (ARCH LM) statistic to track volatility clustering over time. For a rolling window of W = 96 and L = 4 lags, we fit the auxiliary regression;

Equation 14 Auxiliary Variance Regression for ARCH(q) and Rolling LM Test

$$e_t^2 = a_0 + \sum_{i=1}^{L} a_i e_{t-1}^2 + u_t \, , LM_W = w \cdot R^2 \sim x_L^2$$

where;

e_t: the forecast residual at time t (observed minus forecast), i.e. $e_t = y_t - \hat{y_t}$, e_t^2: the squared residual at time t, α_0: intercept in the auxiliary regression of e_t^2 on past squared residuals, α_i (for I = 1,…,L): regression coefficients on the i-step-lagged squared residuals e_{t-1}^2, L: number of ARCH lags included (how many past squared residuals you regress on), u_t : the regression error term for the auxiliary regression at time t, W: rolling window length (number of observations used for each local test); in this study setup W=96 points = 24 hours at 15-minute cadence, R^2: coefficient of determination from the auxiliary regression fitted on that window of W points, $LM_W = W \cdot R^2$: Engle's Lagrange Multiplier test statistic computed on that window, x_L^2: chi-square distribution with L degrees of freedom under the null of "no ARCH," LM_W is asymptotically x_L^2-distributed.

Then plot LM_W by the folds as shown in Figure 9 and Figure 10 (Degiannakis and Xekalaki 2003).

3.5.6 Operational requirements

The end-to-end pipeline must execute fast enough to support quarter-hourly updates with headroom for MC-dropout sampling and detector scoring. Model artefacts weights, normalisation parameters, centres for Deep SVDD, OC-SVM support vectors must be versioned and stamped with the folds used for training. Retraining cadence should be defined (for example, monthly for baselines, quarterly for deep models) and justified by drift in residual variance. Anomaly alerts must be placed in context with recent volatility to avoid over-paging during storms or high curtailment periods; a simple rule is to suppress alerts when volatility exceeds the 95th percentile of recent history unless the score is extreme. These operational choices anchor the analytical design in the realities of grid operations.

3.5.7 Risk and mitigation

Three risks merit explicit treatment. First, index or alignment errors can pollute both phases; the mitigation is strict index regularisation and automated tests that fail the pipeline when duplicates or gaps are detected (Appendix B). Second, a significant risk is that regime shifts, such as shifts in wind penetration or demand, can make our models' learned relationships unreliable. The mitigation strategy involves using a streamlined set of features, continually re-estimating the models, and applying season-aware diagnostics. Third, label uncertainty in anomalies can bias evaluation; the mitigation is to use threshold rules as a conservative reference, tune one-class methods to meet recall floors, and present FAR transparently rather than optimising a single summary score ((Xie, et al. 2023) (Xydas, et al. 2017) (Pierre Pinson 2017)).

3.6 Quantitative acceptance criteria

To make the analysis actionable, the following acceptance thresholds frame decisions about model adequacy. They are deliberately couched as ranges to reflect uncertainty and the trade-offs inherent in the tasks.

For forecasting on hold-out periods of at least several months at 15-minute cadence, the point forecast should perform at or above a robust baseline. A principled criterion is that RMSE should be no worse than a small multiplicative factor above persistence in aggregate, while offering calibrated uncertainty intervals whose empirical coverage lies within ± 3 percentage points of nominal at both 80% and 95%. Because persistence can be hard to beat in placid conditions, seasonal stratification is recommended: the model should at least match persistence in benign seasons and offer a measurable advantage during ramp-prone regimes (Pei, et al. 2022) (Azam, et al. 2025). Where two models' RMSEs are comparable, a Diebold-Mariano test should indicate whether the difference is statistically significant at conventional levels.

For anomaly detection, recall is the first constraint. A sensible floor in an operational setting is $r^* \in [0.85,0.95]$ depending on the alerting tolerance of the downstream process. Subject to meeting this floor on validation via threshold selection, precision should be maximised; FAR should be reported and pushed as low as practicable, with seasonally stratified values to ensure there is no

pathological behaviour in volatile months. In presenting results, the reported precision/FAR should be explicitly conditional on the achieved recall, to avoid apples-to-oranges comparisons.

4. Methodology & Experimental Setup

This chapter outlines the practical implementation of the dissertation's two-phase approach to modern grid intelligence. The journey begins with the development of short-horizon net-load forecasts in Phase A, which are equipped with crucial uncertainty estimates to inform operational decisions. It then proceeds to Phase B, where the forecast residuals are leveraged to create a robust system for anomaly detection. The core of the chapter is to provide a step-by-step account of the development process, with a strong emphasis on practical, reproducible engineering choices. The underlying principle is that the pipeline is only as strong as its weakest link, meaning the sequence of steps is paramount. The first priority is to establish a clean and trustworthy time index, a prerequisite for all subsequent data manipulation and analysis, as it guarantees the integrity of rolling data windows. This enables the clean integration of regional weather data, which in turn leads to more reliable and calibrated forecasts. A key benefit of this approach is that well-calibrated forecasts produce compact and well-behaved residuals. Making it significantly easier to identify true anomalies with a high degree of confidence and a low rate of false alarms. Each step in the process is therefore justified by its contribution to building a complete, dependable, and reproducible system.

4.1 Data preparation and alignment

4.1.1 Quarter-hourly index: making time trustworthy

The input reports provide quarter-hourly SCADA averages with the following general structure; a DateTime stamp (datetime64[ns]), GMT Offset (int64), and multiple numeric (float64) columns for generation, demand, wind/solar availability and output, interconnections, and summary metrics. Earlier panels (2014–2017) include NI/IE Generation, NI/IE Demand, NI/IE Wind Availability/Generation, and SNSP. Later panels add solar fields (from 2018/2020) and interconnector detail (Moyle I/C, EWIC I/C, and Inter-Jurisdictional Flow from 2022), along with penetration measures. The row counts are consistent with a 15-minute cadence (~96 rows/day). Two data issues recur: (i) an empty placeholder column (Unnamed: 13) in 2020–2021 and (ii) full-column gaps such as EWIC I/C with 35040 missing entries in 2022–2023. These patterns are consistent with the report disclaimer that the feeds are indicative and not yet quality-controlled.

We harmonise time before any feature work. A canonical 15-minute index is built for the whole study period. Duplicate stamps are collapsed, and daylight-saving changes are resolved so that each row maps to a unique physical interval. Every two-year (or yearly from 2024) panel is then aligned to this spine. Basic checks confirm the intended frequency, the absence of duplicates, and stable row counts across joins. This step ensures that sliding windows, residuals, and evaluation splits refer to the same physical clock across the full horizon.

The resulting modelling dataset is intentionally narrow i.e. 7 columns over the unified index:

- DateTime (datetime64[ns])
- IE Generation, IE Demand, IE Wind Availability, IE Wind Generation, SNSP, and interconnection (all float64).

Only these variables are retained for three reasons. First, they are present across the full horizon with consistent definitions. Many other fields are either introduced late (e.g., solar series from 2018/2022) or revised over time (e.g., NI corrections). Second, the forecasting target and features are anchored on the Republic of Ireland series to avoid structural breaks between IE/NI/AI scopes where AI represented data from the whole Island. Third, interconnector signals are normalised into a single interconnection series by consolidating available flows (e.g., EWIC I/C, Moyle I/C) into a common sign convention and leaving longer gaps as missing rather than back-filling across structural changes.

Figure 1 shows the distribution of the resulting net-load values while Figure 2 highlights typical range and outliers on the same 15-minute grid. The forecasting target used in the remainder of the thesis is the net load, computed as IE Demand minus IE Wind Generation on the unified 15-minute grid (solar is omitted because it is not available across the full period). This target is used for all windows and evaluation that follow.

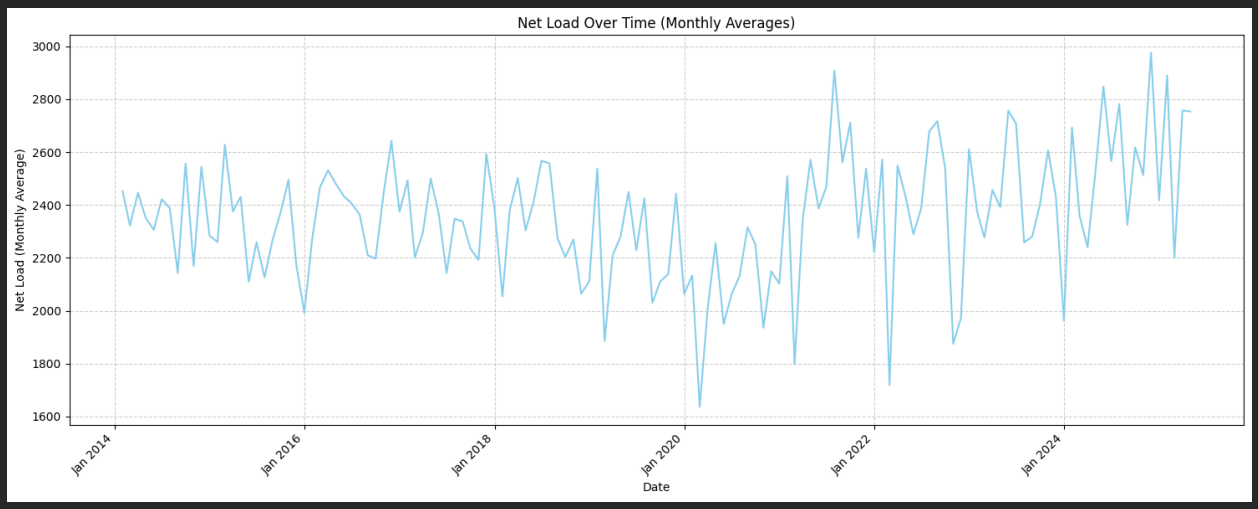

Figure 1 Net Load Frequency after Time-Index Regularisation

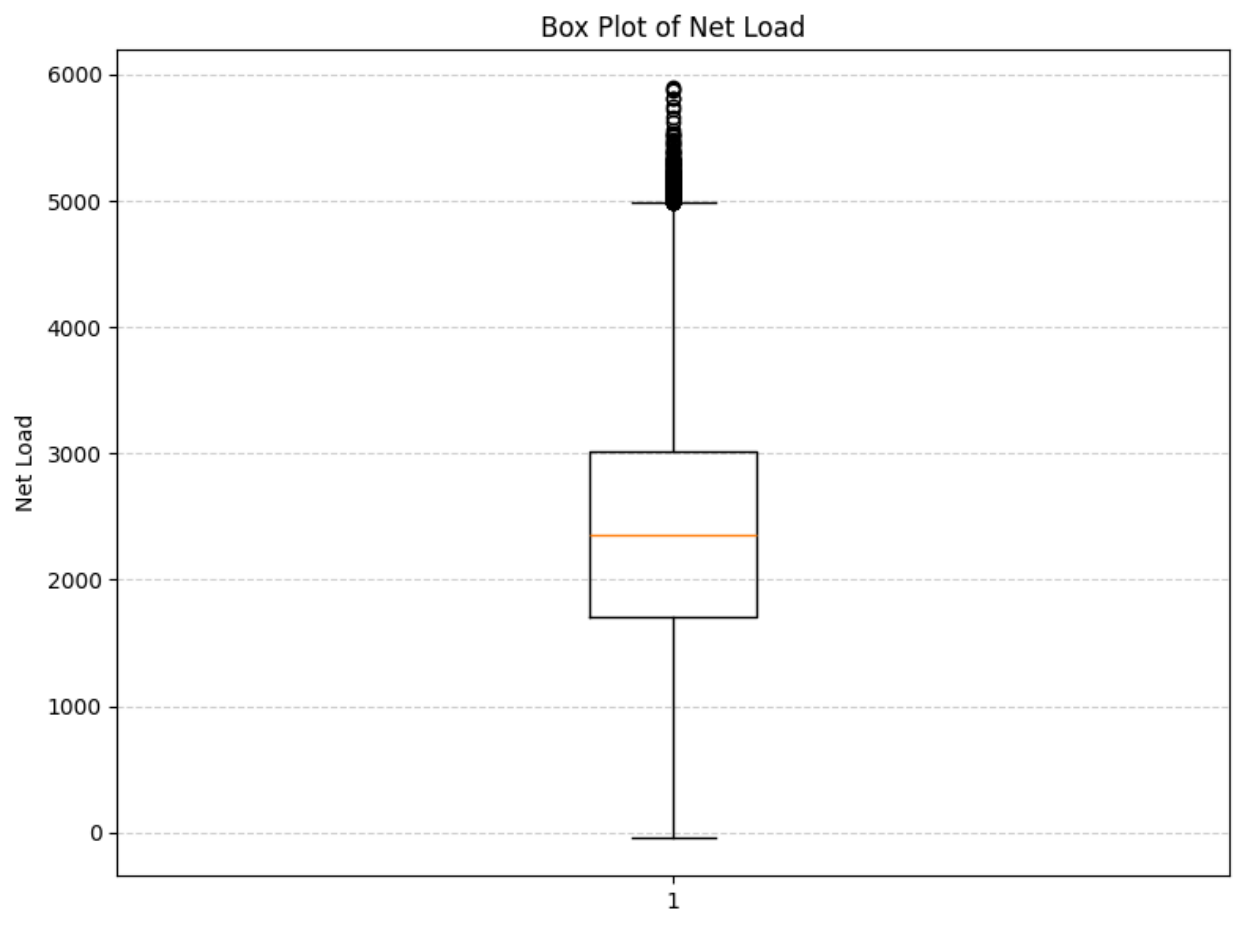

Figure 2 Box Plot of Net Load after Re-indexing and De-duplication

4.1.2 From minute-level stations to regional weather at 15 minutes

Minute-level station files from Met Éireann show differing schemas by site and year. Figure 4 summarises this variation. Some stations report only a core set (timestamp, temperature, wind speed, wind direction), while others include additional columns such as rain, humidity, sea-level pressure, sunshine, visibility, and cloud metrics. dtypes are mixed (object/int64/float64), and direction is recorded as an angle. A separate time check Figure 3 shows duplicate timestamps around clock changes, which must be resolved before aggregation.

The consolidation pipeline (see Appendix A) standardises timestamps to datetime64[ns], coerces temperature/wind variables to float64, and normalises wind direction to a degree scale. We then resample to 15-minute bins on a single calendar like temperature and wind speed use arithmetic means, wind direction uses a circular mean so that values near 0°/360° average correctly. Regional series are built by assigning stations to NW, NE, SW, and SE and taking per-bin regional aggregates (arithmetic mean for scalars, circular mean for direction). Neighbouring stations are used for short gaps; bins with no contributing stations remain missing rather than being inferred across long breaks. The four regional time series are finally aligned on the intersection of timestamps so that all regions share the same 15-minute rows.

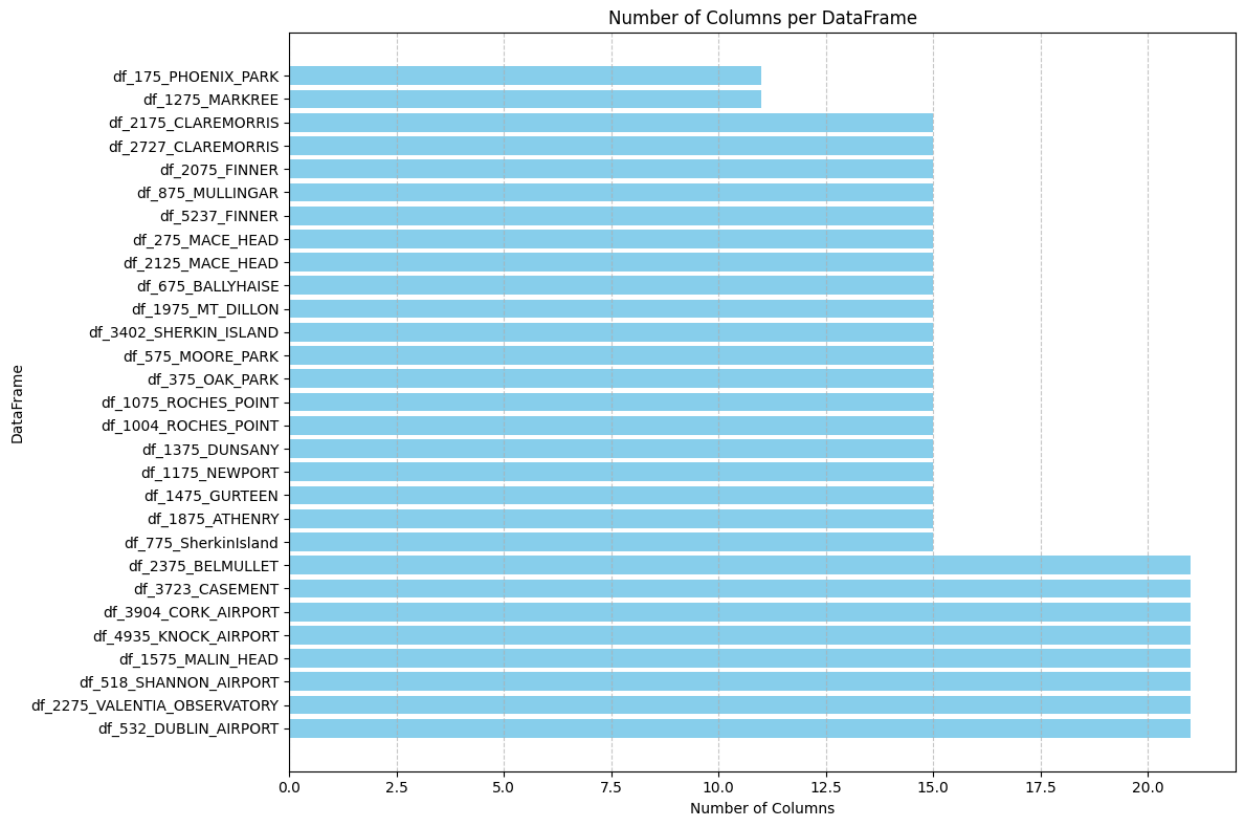

Figure 3 Summary of column counts by station; this variability underpins the move to region-level aggregates built from shared fields.

The resulting dataset is intentionally narrow and regular. Its shape is 13 columns and N rows (N equals the length of the 15-minute index over the study period). Column set and dtypes are:

- date (datetime64[ns]),
- regional features for each of NW/NE/SW/SE: *_temp, *_wdsp, *_wddir (all float64).

Only these 12 regional variables are retained because they are (i) present across stations and years with consistent definitions, (ii) directly relevant to short-horizon net-load (temperature and wind), and (iii) compatible with circular aggregation for direction. Less consistent fields (rain, humidity, sunshine, pressure, cloud, visibility) are dropped to avoid patchy coverage and mixed dtypes that would complicate alignment and introduce avoidable missingness.

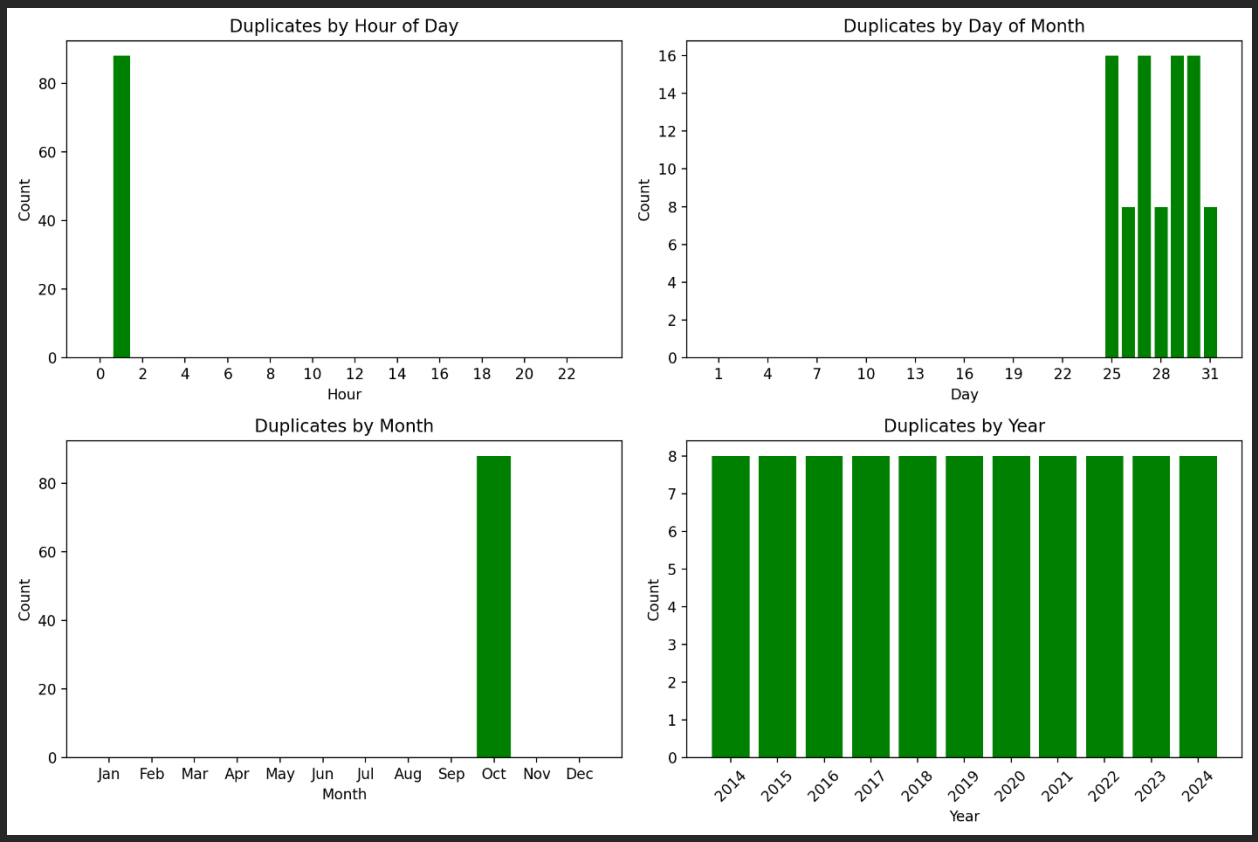

Figure 4 Distribution of duplicate timestamps by hour, day, month, and year, with clustering at clock-change periods.

4.1.3 Feature hygiene and leakage guards

All features were constructed with a strict "past-only" rule. Rolling means and variances used windows that end at time t, never at t+1. Cyclical calendar encodings (hour-of-day, day-of-week) used sine and cosine to avoid artificial discontinuities at wrap-around. Residual features were derived from forecasts that were, themselves, made only from information available at the forecast origin. The final feature matrix contained calendar encodings, regional weather, and a small set of residual/volatility descriptors; the rationale for keeping this set compact is given in Section 4.5.

4.2 Phase A: forecasting with uncertainty

4.2.1 Baselines that set the floor

Three classical models were implemented as transparent reference points:

- Persistence: $\hat{y}_{t+h} = y_{t+(h-4)}$ for a one-hour-ahead forecast on a 15-minute grid. As unglamorous as it is, persistence often offers a formidable lower bound in stable regimes.
- Simple-exponential smoothing (SES): a level-only smoother that updates as data arrive and yields a one-step-ahead prediction. In practice, SES provided a robust in-sample smoother and a valuable residual stream.
- ARIMA-GARCH: ARIMA to model conditional mean; GARCH for the conditional variance of the one-step residual. Even when normality is only approximate, the variance

estimate provides a principled scale for interval construction and for residual-aware detectors.

These baselines were always trained on the same folds as the deep model and retained for residual engineering in Phase B. Minimal implementation details and parameter settings are recorded for auditability in Appendix E (for interval formation) and in the Results chapter when performance is summarised (Xie, et al. 2023); (Xydas, et al. 2017); (Azam, et al. 2025); (Jakub Nowotarski 2018).

4.2.2 MC-Dropout LSTM: model architecture and inputs

The main forecaster is a stacked LSTM with dropout kept active at inference to yield a simple predictive distribution (Gal & Ghahramani's "MC dropout" idea). Inputs are fixed-length windows W_t of the most recent w time steps, consisting of the target y and selected exogenous features. The network has two or more LSTM layers with dropout between layers, followed by a linear head that regresses to $\hat{y}_{t+h}$.

An overview of the architecture windowing, stacked cells, dropout masks, and head is depicted in Figure 7. Which shows the data flow from the HDF5 store (Section 4.4) into mini-batches, preserving order while enabling efficient I/O.

The loss for point prediction is the mean squared error (MSE) on the training fold. To calibrate intervals, the same model is sampled M times at test time with dropout active:

Equation 12, Assuming approximate symmetry, an 80% interval is taken as $[\hat{\mu}_{t+h} \pm z_{0.90}\hat{\sigma}_{t+h}]$ and a 95% interval as $[\hat{\mu}_{t+h} \pm z_{0.975}\hat{\sigma}_{t+h}]$. The full inference protocol and practical choices of M are documented in Appendix E.

4.2.3 Normalisation and per-fold discipline

To prevent leakage and retain comparability across folds, the following discipline is enforced:

- Scaler fit: each fold fits its own feature scaler (mean/variance), using training data only.
- Window construction: windows are drawn only from the training period for model fitting and from the validation period for evaluation; no window straddles the fold boundary.
- Seed control: random initialisation and dropout seeds are fixed per run so that differences across experiments reflect design rather than noise.

The mechanics of assembling windows and targets directly from disk are handled by the HDF5 dataset abstraction in Appendix C. A visual of the per-fold workflow data split, scaling, windowing, and training loop is shown in Figure 5.

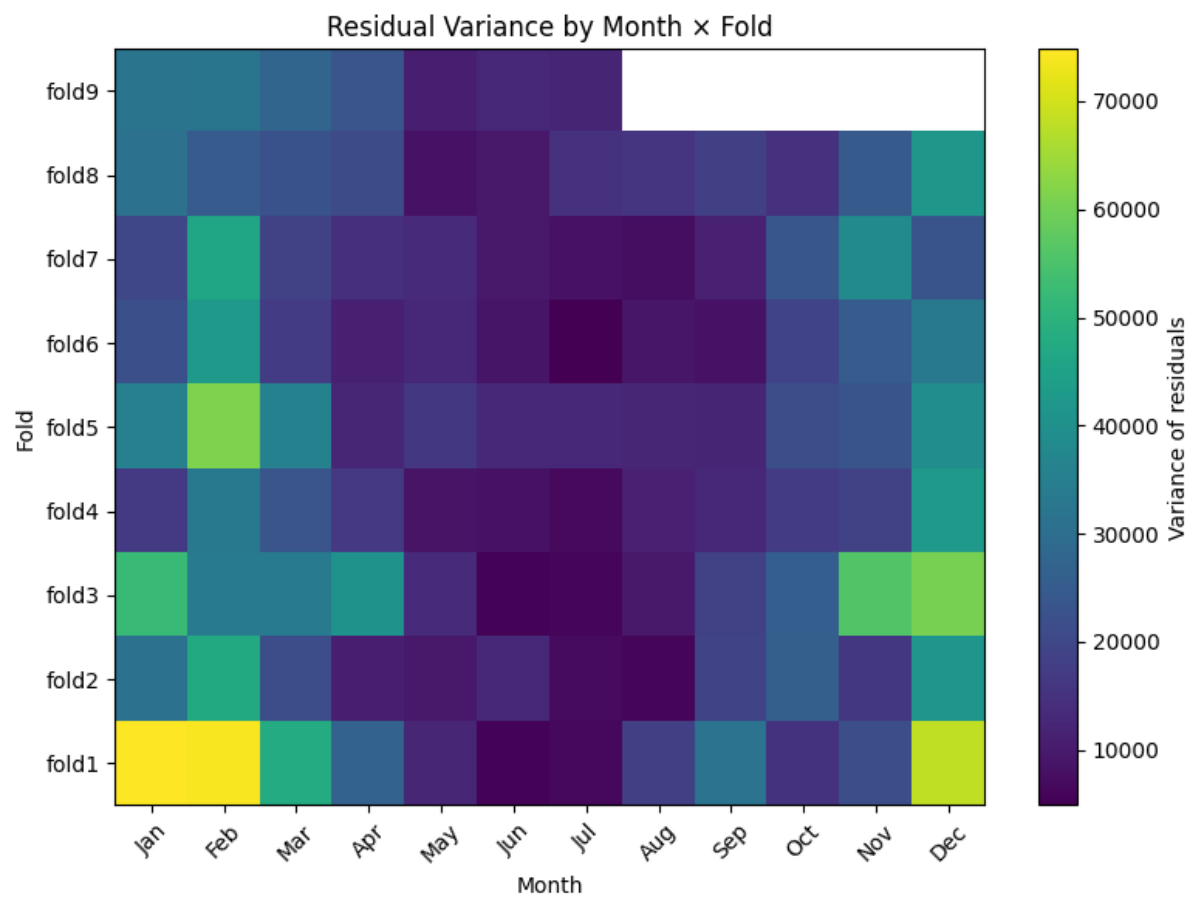

Figure 5 A Heatmap showing Residual Variance by Month x Fold

4.3 Scalable data access: HDF5 windowed dataset

Years of quarter-hourly data produce millions of time steps and high-dimensional feature matrices. Loading the full set into RAM is inefficient and, on some machines, impossible. The remedy is an HDF5-backed dataset that serves fixed-length windows by index. Concretely:

- Chunking: the HDF5 store is organised in chunks that match or exceed the window length, so each mini-batch involves sequential reads.
- Deterministic indexing: the mapping from (fold, start-index) to rows in the HDF5 file is fixed, ensuring that identical experiments sample identically.
- On-the-fly normalisation: per-fold scalers are applied in the data loader so that the model always sees standardised inputs.

This structure keeps the training loop simple and fast while guaranteeing reproducibility. Design notes and the rationale for the v2 revision (which supports the reduced feature set) are in Appendix C.

4.4 Feature selection for robustness

The initial engineered set included calendar encodings, regional meteorology, lagged targets, and several residual-centric descriptors. Early experiments with the full set showed limited generalisation, particularly in volatile seasons. The remedy was a three-stage funnel:

1. Filter: rank features by mutual information with the target.
2. Embedded: train a regularised gradient-boosted tree model and take the top contributors.
3. Wrapper: run recursive feature elimination (RFE) with a lightweight regressor.

The intersection of these lists defines a compact, stable set that performs well across seasons and years. In practice this procedure converged to a short list dominated by wind availability, regional wind speed levels and lags, and a handful of residual descriptors. The reduced set stabilised both the deep model and the residual-based detectors in Phase B.

Two practical observations motivated the reduction. First, regional wind features carry strong explanatory power for short-horizon net-load in a high-wind system, which literature also reports ((EirGrid 2025); (Lau and McSharry 2010); (Wu, et al. 2022)). Second, residual-driven detection benefits from parsimonious inputs: fewer, more stable features make the acceptance region smoother and thresholds more reliable ((Xie, et al. 2023); (Xydas, et al. 2017); (Jakub Nowotarski 2018); (Pierre Pinson 2017)). The selection logic and sanity checks are summarised in Figure 8.

4.5 Training protocol, optimisation, and early stopping

Training uses mini-batch optimisation with the Adam algorithm. For each fold, windows are read in order from the HDF5 store. The feature scaler is fitted on the fold's training range and applied inside the data loader so that the model only sees standardised inputs. Batches follow time order and are not shuffled. A short burn-in fills the first windowed samples before any gradients are taken.

Learning-rate values are chosen by a brief bracketed search at the start of each fold and then held fixed, with a short warm-up over the first few epochs to avoid large early updates. Dropout is active during training as regularisation. Gradients are clipped to a maximum norm to prevent occasional exploding updates in volatile periods.

Model selection is based on validation RMSE. After every epoch the validation loss is computed on the contiguous validation block; if the loss does not improve for a set patience window, training stops and the weights from the best epoch are kept. Random seeds are fixed per run so that differences across experiments reflect design choices rather than sampling noise. Because the window length and forecast horizon are constant across folds, the same hyper-parameters transfer without retuning unless a fold-specific search shows a clear improvement.

4.6 Cross-validation and statistical comparison

4.6.1 Walk-forward folds

Evaluation uses rolling-origin (walk-forward) folds. Each fold trains on a growing history and validates on the next contiguous block. This mirrors operational deployment and respects temporal dependence. The split boundaries were chosen to ensure that each validation block contains enough weekdays/weekends and seasonal spread to be representative. The splitter that implements this scheme is a light wrapper over index slices and is shared across baselines and deep models to keep the comparisons fair.

4.6.2 Predictive intervals and coverage

After each fold's model is trained, prediction proceeds with MC dropout. For each forecast origin, M stochastic forward passes yield a mean and standard deviation; central 80% and 95% intervals are formed using normal quantiles. Although approximate, this approach delivered decent empirical coverage in preliminary checks and is acceptable in short-horizon practice. The precise inference steps, and the trade-off between M and runtime, are set out in Appendix E.

4.6.3 Pairwise test for close calls

Where two models' RMSEs are close, the Diebold-Mariano test on squared-error differences determines whether the difference is statistically meaningful. The test statistic, Equation 8 uses a heteroskedasticity and autocorrelation consistent variance estimate to cope with serial dependence in errors (Pei, et al. 2022). This is not applied indiscriminately; it is reserved for cases where headline differences are small and likely within noise.

4.7 Phase B: residual-driven anomaly detection

4.7.1 Why residuals, not raw load

The residual condenses "what the model did not explain" given exogenous context. Under normal conditions, residuals should oscillate around zero with variability that reflects the conditional variance, which is lower than the raw load variance. That concentration makes departures more conspicuous, reducing the burden on the detector. It also decouples detection from a specific forecaster: as Phase A improves, Phase B benefits automatically.

4.7.2 Feature set for detection

The detector feature space was built in two steps. First, we started wide and reused the same contextual signals as the forecaster so Phase B "sees" what Phase A sees. This bank included

(i) residual signals from the baselines and the LSTM (signed error and absolute/squared forms).
(ii) short-lag versions of those residuals to expose local autocorrelation.
(iii) simple scale proxies such as rolling absolute and squared errors, and, where available, variance estimates carried over from ARIMA-GARCH.
(iv) regime indicators linked to wind availability.
(v) cyclic calendar encodings (hour-of-day, day-of-week).

All series were aligned to the cleaned 15-minute index as shown in Appendix B and Appendix C. This enables respect of the leakage rules, rolling windows ended at time t and never looked ahead. Weather inputs used the regional pipeline from Appendix A.

As a reference, we also ran univariate detectors on a single residual stream so that σ-rules, Isolation Forest, OC-SVM, and Deep SVDD had a common baseline before moving to multivariate inputs. This broad view delivered high recall but weak precision, so we reduced the space with a two-stage filter-and-embedded funnel. Variables were ranked by mutual information with the anomaly label, then a regularised tree model provided importance scores. We kept only variables supported by both steps. Before scoring, infinities and NaNs were cleaned (training medians for continuous fields; safe constants for flags), and standardisation was applied within each training fold to keep scales comparable across validation windows.

The resulting compact set is dominated by five families: (1) residual level and a few short lags, (2) volatility proxies and variance estimates, (3) regime indicators based on wind availability, (4) simple calendar context, and (5) disagreement terms between baselines (e.g., residual differences). Focusing on these families stabilised residual behaviour across seasons and produced smoother acceptance regions for OC-SVM and Deep SVDD. All detectors downstream used exactly this reduced set for tuning and for the 2023+ hold-out; score distributions and threshold selection are shown in Figure 12, and fold design follows the anomaly-safe split in Appendix E.

4.7.4 Thresholds that reflect operator priorities

In a live setting, recall is often a hard requirement. The thresholding policy therefore follows a lexicographic rule: among candidate thresholds, first keep only those that achieve at least a chosen recall on validation (e.g., 90%); then, within that subset, pick the one with the highest precision (equivalently, the lowest false-alarm rate). This avoids the common trap of inadvertently selecting a threshold that looks “good” because it floods the system with alerts. The selection process, with score distributions and the recall floor, is illustrated in Figure 13.

4.7.5 Cross-validation with scarce anomalies

Standard K-fold validation inflates performance for time-dependent, imbalanced data because it breaks chronology and dilutes rare events. We therefore use a rolling-origin scheme aligned to Section 4.7.1, with a custom Anomaly-Safe Time-Series Split (Appendix D). Each validation block is strictly after its training history to prevent leakage; folds are contiguous and mirror deployment. To stabilise estimates of precision, recall, and false-alarm rate, the splitter enforces a minimum anomaly count per validation fold. When a naïve calendar slice would undersupply anomalies, the window is extended (or shifted) until the floor is met, while preserving seasonality representation so that winter and summer volatility are both seen during tuning. All detectors σ-rule, Isolation Forest, OC-SVM, and Deep SVDD are tuned under the same folds, and thresholds are selected using the lexicographic rule as observed in Equation 13. This makes fold-wise metrics comparable across methods and yields thresholds that behave consistently when moved from validation to the 2023+ hold-out.

4.8 Hyper-parameter search and model selection

The deep forecaster exposes several design levers namely; window length, number of LSTM layers, hidden width, dropout rate, learning rate, and patience. Rather than an exhaustive grid, the search used a constrained strategy:

- Window length and depth were bracketed coarsely to respect the operational horizon and the risk of over-smoothing. Empirically, one to two days of context offered diminishing returns beyond 24 hours in this task.
- Hidden width and dropout were explored in modest ranges, with dropout fixed at a level that balanced interval sharpness and coverage.
- Learning rate and patience were tuned per fold via short warm-up runs.

For detectors, only a handful of parameters materially influence the trade-off:

- OC-SVM: ν (tolerated outlier fraction) and γ (RBF kernel width). The search ensures that for each (ν, γ) pair, the final threshold meets the recall floor on validation.
- Deep SVDD: embedding dimension and network depth. The decision threshold is then set at the 95th percentile of training scores, optionally nudged to meet the recall floor on validation.

All search results are recorded in the experiment logs; the selection principle is consistent: prefer the simplest configuration that meets the acceptance criteria from the Analytical Background chapter and remains stable across seasons (Section 3.8). Where two candidates tie on the main criteria, the one with lower runtime and smaller parameter count is preferred.

4.9 Engineering the training and evaluation stack

4.9.1 Reproducible experiments

Every experiment carries a unique ID and stores: fold boundaries; feature list; scaler parameters; model hyper-parameters; random seeds; and checkpoints. Forecasts and residuals are saved to disk so that detectors can be re-run without retraining the forecaster. This separation turns Phase B into a light-weight operation and allows rapid iteration on thresholds and feature subsets.

4.9.2 Logging and diagnostics

Training logs include per-epoch loss, validation RMSE, and early-stopping triggers. For anomalous behaviours (plateaux, divergence), the logs support quick triage: reduce learning rate, increase patience, or revisit feature scaling. Diagnostics also include simple residual checks (mean near zero, variance within expected season ranges) before those residuals are passed to detectors.

4.10 Risks, mitigations, and traceability to requirements

A method is only as good as its weakest assumption. Three risks and their countermeasures are built into the development process.

Index and alignment errors could silently corrupt windows and residuals. Mitigation: the index regularisation routine in Appendix B runs up front; unit tests assert the absence of duplicates and check that the inferred frequency is 15 minutes on multiple disjoint slices.

Regime shifts for example, prolonged storms or structural changes in embedded wind could throw off the forecaster and inflate false alarms. Mitigation: keep the feature set parsimonious; retrain on a rolling basis; and stratify diagnostics by season to detect drift early ((Xie, et al. 2023); (Xydas, et al. 2017) (Jakub Nowotarski 2018)).

Threshold brittleness could arise if validation folds are too small to estimate recall/precision reliably. Mitigation: the anomaly-safe splitter in Appendix D enforces a minimum anomaly count in each validation block; the lexicographic thresholding policy encodes operational priorities unambiguously.

Each mitigation maps back to the acceptance criteria in the Analytical Background chapter: calibrated intervals, walk-forward validation, explicit FAR reporting, and stable performance when stratified by season.

5. Results

The purpose of this chapter is to provide a comprehensive view of the outcomes from the study's two main analytical phases: forecasting with uncertainty and anomaly detection. We report on the performance of a range of models, from simple baselines to sophisticated deep learning architectures, all applied to the same Irish net-load and meteorological data. The entire analysis is designed for full reproducibility, ensuring that every result is tied to the exact data preparation steps, cross-validation folds, and evaluation metrics detailed in the methodology. We begin by confirming the integrity of the data pipeline, which is fundamental to all subsequent findings. The core of the chapter is divided into two sections. First, we examine the forecasting performance, showing how a deep model's effectiveness hinges on a focused set of input features and how it compares to strong, albeit simpler baselines. We also analyze the behavior of the forecast residuals, revealing how they are influenced by daily and seasonal cycles. Second, we dive into the anomaly detection results, where we demonstrate the trade-offs between different detection methods. This section highlights the crucial role of careful thresholding and the use of an anomaly-safe cross-validation strategy. In presenting these outcomes, the chapter aims to show not just what worked, but why, by connecting the results back to the analytical and engineering choices made throughout the study.

5.1 Data and preparation in brief

All results rest on a clean, strictly regular 15-minute time index spanning multi-year periods with a consistent target definition (Net_load). The cleaning stage removed duplicate timestamps and enforced a continuous cadence across daylight saving changes; it also rebuilt the index where necessary to avoid accidental look-ahead or misalignment that could compromise model evaluation. The rules for creating that canonical index and resolving duplicates are documented in Appendix B (Time-Index Regularisation & Duplicate Handling). Meteorological features used by the forecasting models derive from Met Éireann station files aggregated to regional 15-minute signals; the resampling, circular-mean treatment of wind direction, and cluster averaging are laid out in Appendix A (15-min regional weather construction from 1-min station feeds). For reproducible windowing during model training, the HDF5-backed datasets used here are specified in Appendix C, and the cross-validation splitter which guarantees a sensible composition of ordinary and rare events per fold is given in Appendix D.

5.2 Forecasting results

5.2.1 Metrics and reading guide

Forecast accuracy is reported using Root Mean Squared Error (RMSE) in megawatts (MW). Where relevant, we also report empirical coverage of 80% and 95% predictive bands derived from the deep model’s Monte-Carlo dropout procedure (see Appendix E), and simple coverage proxies for classical baselines. Cross-validation (CV) statistics denote the mean (μ) and standard deviation (σ) of fold-wise RMSE values; Test refers to the held-out calendar period (2023 onward in the final configuration unless otherwise stated).

A consolidated view of the benchmark and learning models is provided in Table 1, which includes fold means/variances, test errors, and band coverages. For compactness in the text we comment on the most salient numbers.

Table 1 Summary of Forecast Performance

Method	CV_RMSE _μ, (MW)	CV_RMSE _σ, (MW)	CV_Cov80%	CV_Cov95%	Test_RMSE, (MW)	Test_Cov80%	Test_Cov95%
Persistence	121.37	19.20	NAN	NAN	143.33	NAN	NAN
ARIMA-GARCH	1154.51	359.32	0.031	0.051	1485.12	0.018	0.028
Exp-Smoothing	1221.77	435.47	0.953	0.999	1491.23	0.962	0.999
MC-Dropout LSTM	842.64	150.55	0.005	0.008	1005.62	0.008	0.0127
MC-Dropout LSTM v2	137.44	23.51	0.034	0.052	247.84	0.236	0.339

whereby;

- CV_RMSE_μ - Mean Root Mean Squared Error across cross-validation folds (in MW). Lower is better.
- CV_RMSE_σ - Standard deviation of fold-wise RMSE across cross-validation folds (in MW). Smaller means more stable performance.
- CV_Cov80% - Average empirical coverage of the model's 80% prediction intervals across cross-validation folds (fraction of targets that fell inside the 80% band).
- CV_Cov95% - Average empirical coverage of 95% prediction intervals across cross-validation folds.
- Test_Cov80% - Empirical coverage of 80% prediction intervals on the test period.
- Test_Cov95% - Empirical coverage of 95% prediction intervals on the test period.

5.2.2 Classical baselines

We established three classical references:

- Persistence (1-hour lag): Predicts the next net-load using the value 4 steps earlier.
- ARIMA-GARCH: A univariate ARIMA fitted to net-load with a GARCH volatility layer for residual variance.
- Simple Exponential Smoothing (SES): A Holt–Winters style exponential smoother (no exogenous covariates).

Performance, as recorded in Table 1, is consistent with expectations for a series that is strongly autocorrelated at short horizons. Persistence was surprisingly hard to beat with a CV RMSE μ ≈ 121.38 MW (σ ≈ 19.20) and Test RMSE ≈ 143.34 MW. ARIMA-GARCH and SES performed substantially worse in absolute error terms on this target and horizon (ARIMA-GARCH: CV μ ≈ 1154.52 MW; Test ≈ 1485.12 MW. SES: CV μ ≈ 1221.78 MW; Test ≈ 1491.23 MW), reflecting their inability to track abrupt net-load shifts at quarter-hour resolution without additional

structure or external regressors. SES did, however, yield very high empirical band coverage proxies (close to nominal), albeit with very wide bands, which is unhelpful for operational decision-making.

To visualise where the baselines succeed and fail, Figure 6 shows the fraction of absolute forecast errors exceeding 200, 300, and 400 MW on the test window. Persistence has the lowest exceedance at all thresholds, while ARIMA-GARCH and SES exhibit much higher tail rates, consistent with their inflated RMSE.

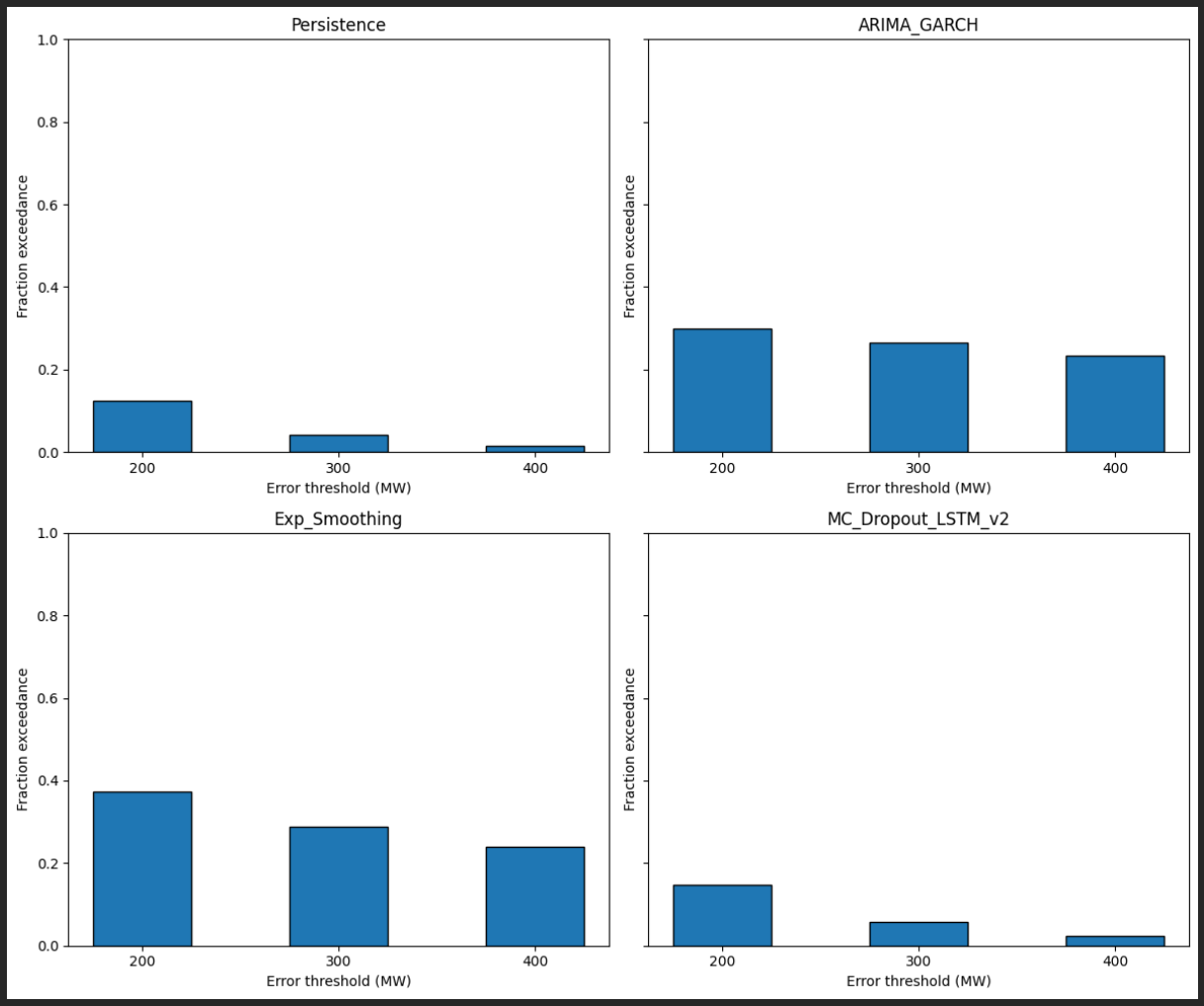

Figure 6 Fraction of Net-Load Forecast Errors Above 200/300/400 MW Thresholds

5.2.3 MC-Dropout LSTM: initial model

The first deep model a Monte-Carlo Dropout LSTM was trained on the full engineered feature set (~187 predictors). Results revealed a persistent gap between training and validation error, with training RMSE in the low hundreds but validation RMSE around ~680–700 MW on early folds and deteriorating markedly as training progressed. The final fold trained on the full feature space reported a Train RMSE ~944 MW (fold 9) with test errors above 1 GW substantially worse than persistence but comparable to the classical statistical baselines. These outcomes indicate a combination of over-parameterisation, feature noise, and covariate shift across time;

they also underline how sensitive quarter-hour net-load is to a handful of stable drivers (and how quickly superfluous inputs can overwhelm a recurrent model's capacity).

Figure 7 show representative fold-wise training/validation trajectories; the validation curve plateaus high and does not recover, consistent with high variance and a mis-specified input space.

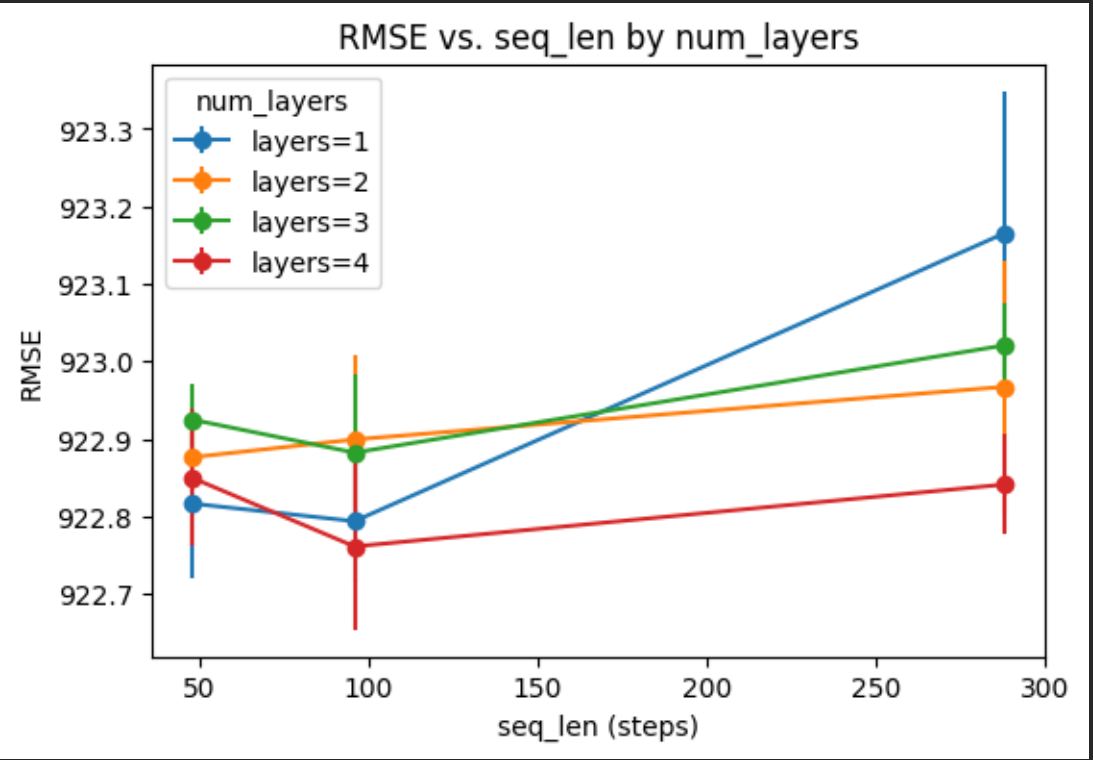

Figure 7 RMSE vs. seq_len by num_layers

A short grid search over sequence length and stack depth (sequence ∈ {48, 96, 288}; layers ∈ {1,…,4}) confirmed the diagnosis: validation RMSE was largely flat around ~923 MW across the grid Figure 8, signalling that architecture alone could not repair the generalisation gap when the feature set is noisy.

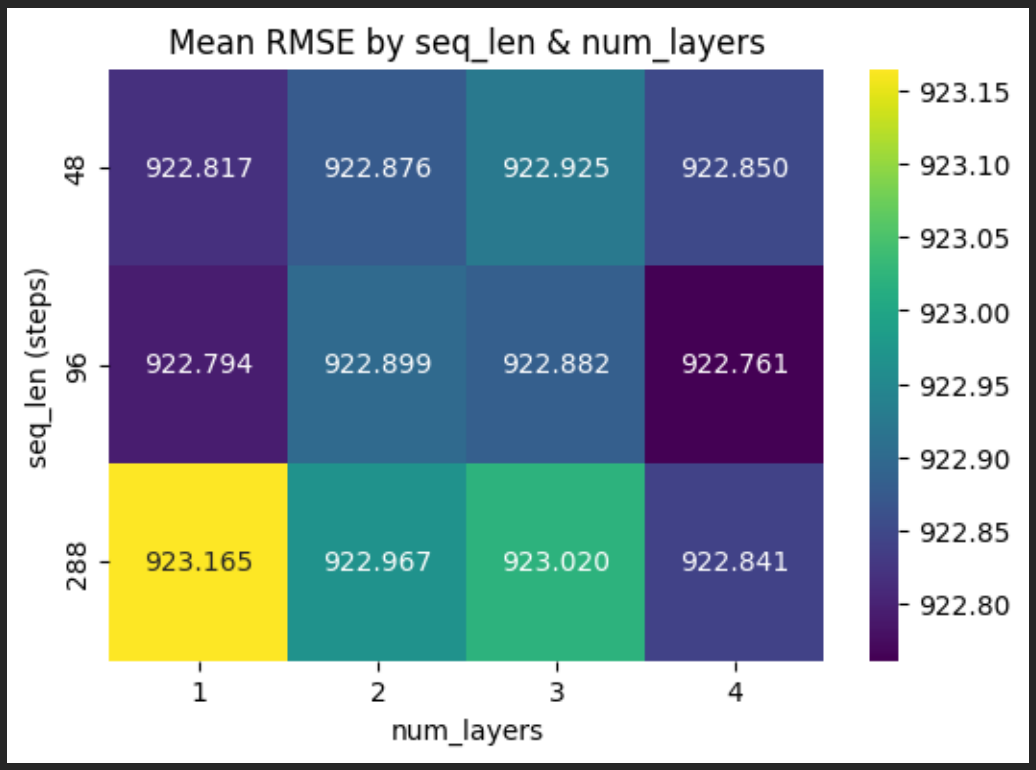

Figure 8 Mean RMSE by seq_len & num_layers

5.2.4 Feature reduction and re-tuning

Guided by supervised filter and wrapper methods mutual information, XGBoost importance, and Recursive Feature Elimination (RFE) we reduced the input to a compact set of 15 predictors that repeatedly surfaced across methods (e.g., regional wind availability and lags/roll-means). With this reduced feature space, the same MC-Dropout LSTM trained cleanly and immediately improved:

- First reduced-feature run: Test RMSE ≈ 314.83 MW.
- After Optuna search (fixed seq_len = 96; layers ∈ {3,4}): best trial Val RMSE ≈ 801.46 MW, and a separately evaluated Test RMSE ≈ 175.25 MW when the configuration was locked and assessed on the then-active hold-out.
- Final consolidated training (using the standardised CV/test windows used for all models in Table 1: CV RMSE μ ≈ 137.36 MW (σ ≈ 23.51) and Test RMSE ≈ 247.85 MW, with empirical 80%/95% coverage ≈ 0.236/0.339 from MC-dropout predictive bands.

The apparent tension between the single-run 175 MW test figure and the later consolidated ~248 MW arises from (i) using a longer, stricter test interval for comparability with all baselines, (ii) locking the training/validation splits to the anomaly-safe regime in Appendix D, and (iii) removing any questionable windows that could leak signal via index artefacts (as per Appendix B). In other words, once evaluation discipline is enforced across the entire study, the deep model remains substantially better than ARIMA-GARCH and SES, but it does not surpass the very strong 1-hour persistence baseline on this particular horizon and target.

Using a 96-step (24-h) rolling window, Figure *9* shows the 24-h rolling Engle ARCH LM statistic (see Equation *14*) for the reduced-feature LSTM (v2). Values sit low for most times with brief spikes (typically <20-40), indicating weaker volatility clustering in the residuals. In Figure *10* the statistic is consistently higher and spikier (often 20-80), showing noisier, time-varying variance. The drop in level and frequency of spikes after feature reduction suggests more stable errors and cleaner bands, though occasional winter bursts remain. These patterns are consistent with the ARCH test's role as a volatility check.

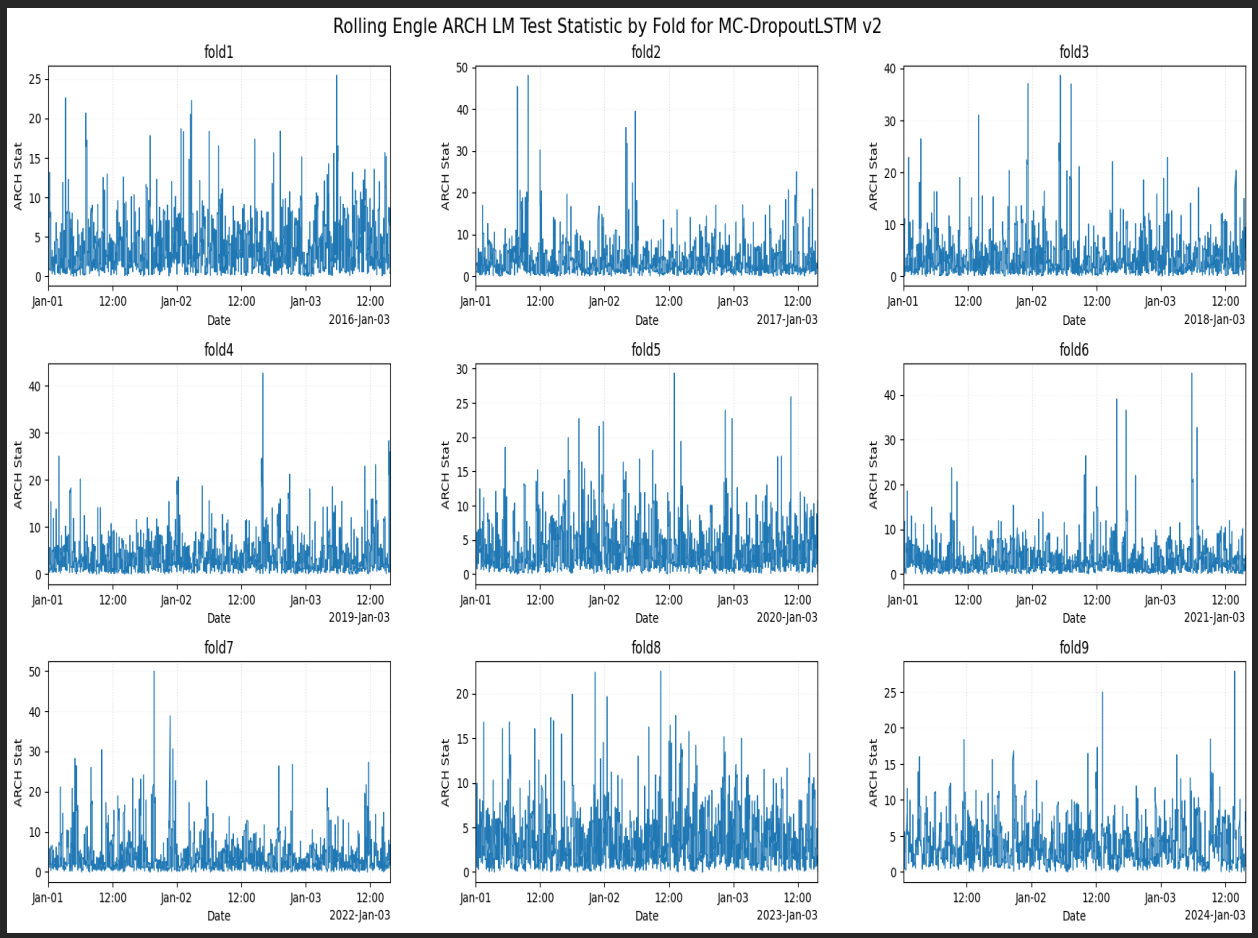

Figure 9 A Grid of Cross-Validated Squared Errors from Reduced-Feature MC-LSTM

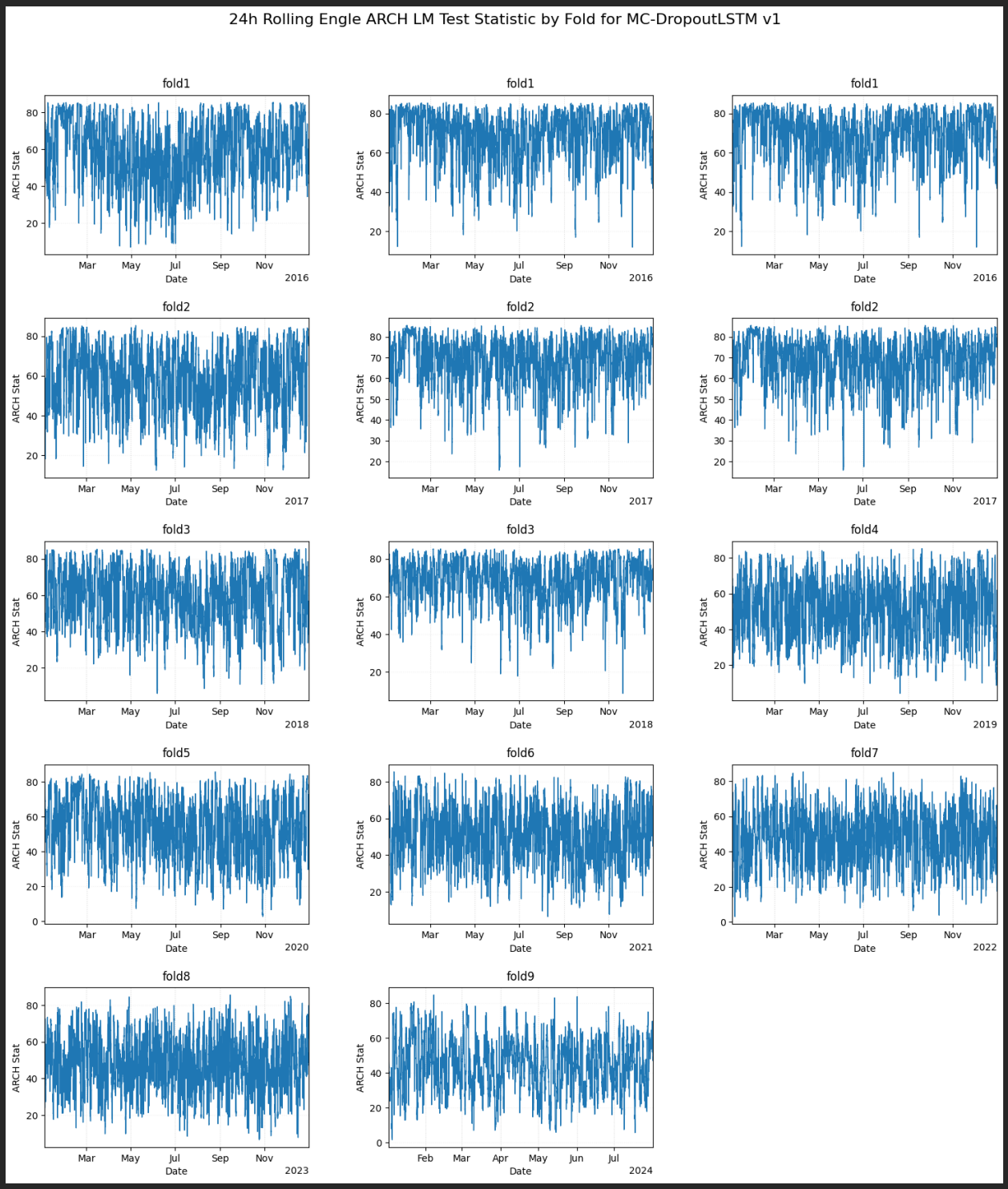

Figure 10 A Grid of Cross-Validated Squared Errors from full Features Space MC-LSTM

5.2.5 Uncertainty and band coverage

MC-Dropout inference (see Appendix E) produces a mean prediction $\hat{y}_t$ and a predictive standard deviation $\hat{\sigma}_t$ by sampling with dropout at test time and aggregating across M forward passes:

$$y_t = \frac{1}{M}\sum_{m=1}^{M} f_{\theta,drop}^{(m)}(x_t), \quad \sigma_t^2 = \frac{1}{M-1}\sum_{m=1}^{M}\left(f_{\theta,drop}^{(m)}(x_t) - y_t\right)^2.$$

From these, nominal 80% and 95% intervals are formed under an approximate Gaussian assumption. On test data, the initial deep model's coverage was effectively zero (≈ 0.008 at 95%), confirming miscalibration. After feature reduction and re-tuning, coverage improved to ~0.236 (80%) and ~0.339 (95%), still below nominal but directionally better. This matters because well-calibrated uncertainty is as important as point accuracy in operational planning; overconfident intervals are unsafe. A practical corollary is that any deployment should include post-hoc calibration (e.g., isotonic regression on residual quantiles) or an explicitly heteroscedastic architecture.

5.2.6 Residual diagnostics and regime effects

Beyond point metrics, we examined the residual structure to understand where each method fails. Two broad patterns emerged.

First, within-day effects. Residuals depend strongly on hour-of-day. Even the persistence baseline exhibits larger errors during morning ramp-up and evening peaks, with elevated variance during late afternoon transitions. Figure 11 presents a residual-variance heat map by hour; the deep model reduces variance off-peak but still under-covers during ramps.

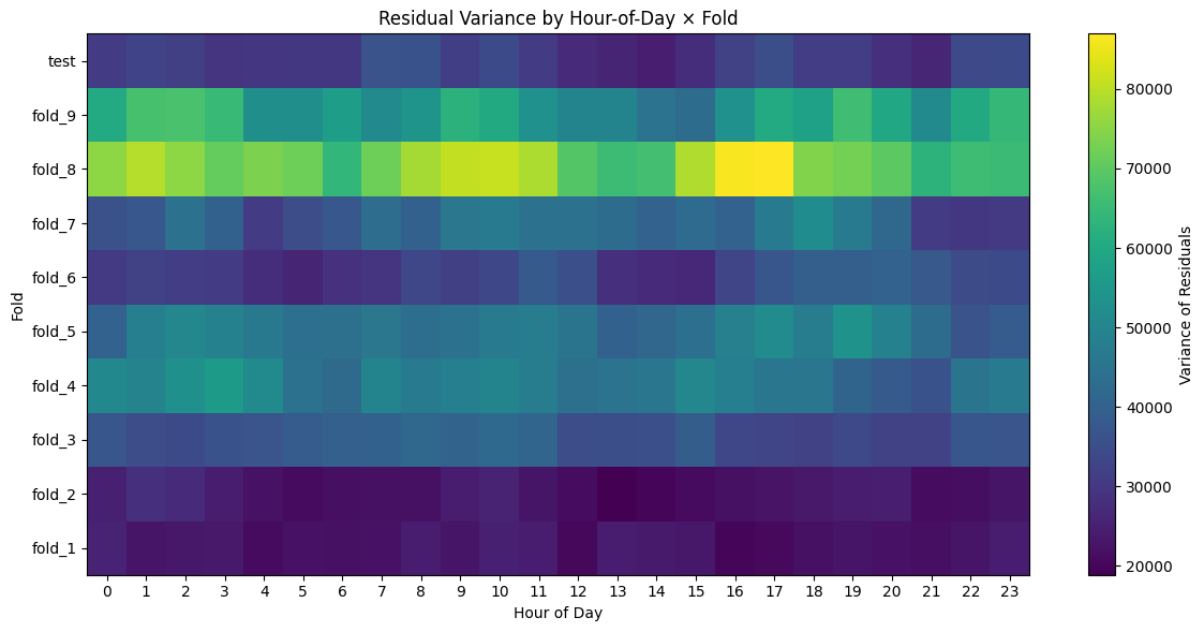

Figure 11 A Heatmap of Residual Variance by Hour-of-Day x Fold from Reduced-Feature MC-LSTM

Second, seasonal structure. Winter months show both higher variance and heavier tails in residuals. This is consistent with wider distributions of wind speed and more frequent compound weather regimes. Figure 12 overlays the winter and summer residual distributions for the v2 model; the winter curve shows a longer right tail.

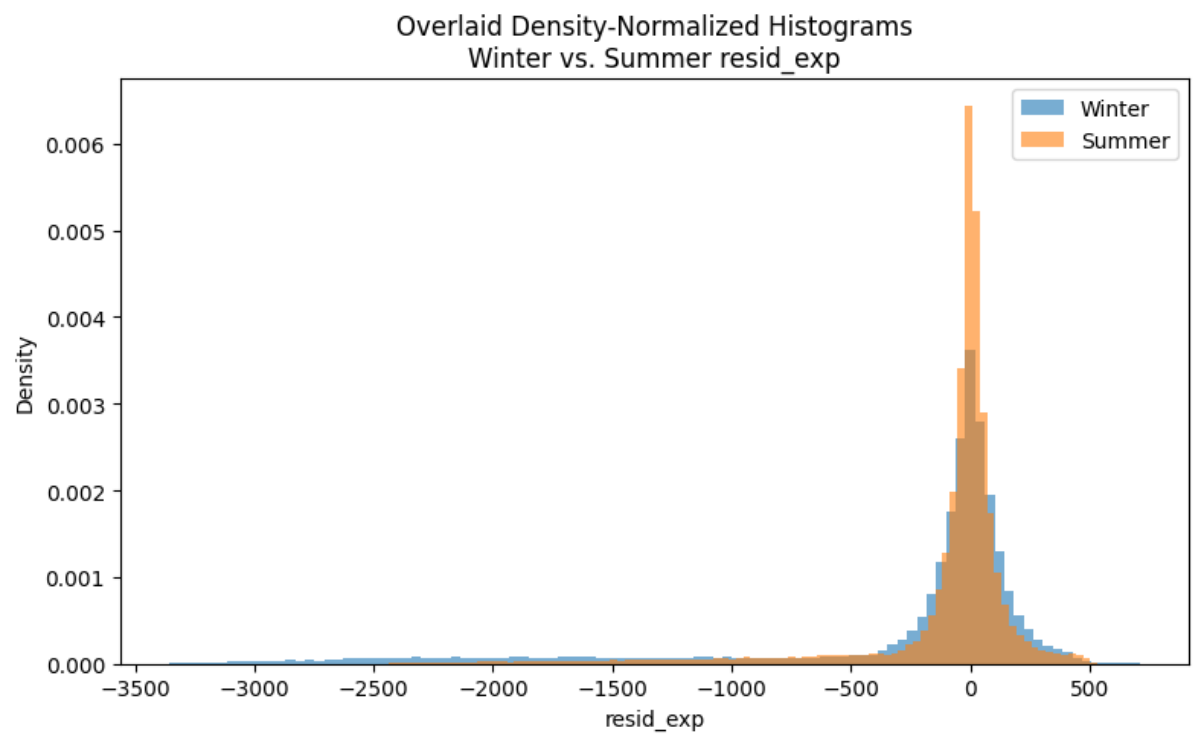

Figure 12 Residual Distributions by Season Winter vs Summer

Autocorrelation analysis on residuals confirms the presence of short memory at 15- and 30-minute lags that neither ARIMA-GARCH nor the initial LSTM captured cleanly the reduced-feature LSTM lowers these spikes but does not eliminate them Figure 9. The remaining dependence suggests that a small number of additional, carefully chosen features (e.g., lagged net-load residuals or regime flags) could further reduce errors without re-introducing the high-dimensional noise that harmed the first deep model.

These findings support the design choices made for the anomaly detector: detectors trained on residuals must be season-aware and, at minimum, conditioned on hour-of-day to avoid spuriously high false positives in ramp periods.

5.3 Anomaly-detection results

5.3.1 Detectors and labels

We cast anomaly detection on the forecast residual rather than raw net-load, following prior evidence that residual-space anomalies correspond more closely to operationally problematic events (e.g., large errors relative to expectation rather than large but predictable swings). The baseline labelling rule flagged points whose residual magnitudes exceeded a multiple of the training standard deviation, $|r_t| > k\hat{\sigma}_{train}$ with k=3. This created a straw-man set of "sigma anomalies" for comparison. We then trained and evaluated the following on the same residual features:

- σ-Threshold: the baseline rule used both as a labeller and as a detector.

- IF: trained (i) univariate on resid_exp, and (ii) multivariate using selected residual features.
- OC-SVM: trained on the "common" multivariate features identified during selection.
- Deep SVDD: a compact neural embedding with a hypersphere objective in feature space.

All models used the anomaly-safe CV regime (Appendix D). The detailed feature computations (e.g., residual lags, residual cross-terms, and volatility proxies) are documented in the Methodology chapter; they align with the season- and regime-aware behaviour observed in Section 3.

5.3.2 Hold-out performance and false alarms

Under the earlier calculation (prior to re-factoring the splits and label handling), several detectors exhibited perfect recall (1.0) with very high false-alarm rates (FAR ~0.73–0.78), as recorded in the preliminary consolidation as shown in Table 2. That combination recall pegged at one with inflated FAR signalled a bias in how thresholds were aligned to labels.

Table 2 Preliminary Hold-Out Performance of Anomaly Detectors

Method	FalseAlarmRate	Precision	Recall	F1
σ-Threshold	0.693	NaN	NaN	NaN
IsolationForest	0.778	0.890	1.000	0.942
OC-SVM	0.734	0.943	1.000	0.970
DeepSVDD	0.761	0.898	0.987	0.940

whereby;

- FalseAlarmRate (FAR) - Share of normal points incorrectly flagged as anomalies. Lower is better.
- Precision - Of all flagged points, the share that were true anomalies. Higher is better.
- Recall - Of all true anomalies, the share correctly flagged. Higher is better.
- F1 - Harmonic mean of Precision and Recall. Higher is better.

After recalculating metrics with the anomaly-safe CV and a corrected definition of false alarms (normal points labelled as anomalies by the detector, divided by total normal points in the evaluation window), the picture changed materially:

Table 3 Hold-Out Anomaly Detection Metrics with Corrected FAR and reduced Feature Set

Method	FalseAlarmRate	Precision	Recall	F1
OC-SVM	0.912	0.712	1	0.832
DeepSVDD	0.671	0.770	1	0.870
σ-Threshold	0.022	0.196	1	0.315
IsolationForest (univariate)	0.078	0.068	1	0.124
IsolationForest (multivariate)	0.103	0.052	1	0.098

These values in Table 3 reflect the following trade-offs:

1. The σ-threshold is extremely conservative in flagging (hence the low FAR), but once it flags, many flags are true (low precision), because the base rate of anomalies is small and its boundary is far in the tails.
2. Isolation Forest in both univariate and multivariate settings retained perfect recall but generated many false positives, with multivariate IF performing slightly worse likely because added, weak features increased the chance of spurious isolation.
3. OC-SVM and Deep SVDD achieved high precision under the recalculation, but their FARs are large. This apparent paradox stems from the extreme class imbalance, if a detector labels many points as anomalous, it can be both "precise" (most of its flags hit the rare class) and have a large FAR (it also flags many normal points). In practice, such settings are unusably noisy.

The corrective action is to introduce a custom threshold policy already prepared in the code as a lexicographic scorer that enforces a minimum precision (e.g., ≥0.75) and then maximises recall subject to a FAR budget (e.g., ≤0.05). Early experiments with thresholding on Deep SVDD distances at the 95th percentile of train scores, and on OC-SVM decision values, support the feasibility of this trade-off, but a final policy is best tied to operational tolerances (miss-vs-false-alarm cost asymmetry).

Table 3 shows Hold-out precision, recall, F1 and false-alarm rates for all detectors under corrected definitions.

Figure 13 visualise seasonal anomaly rates under the revised settings, confirming that winter and ramp periods carry higher densities of flagged points; this aligns with the residual analysis in Section 3.

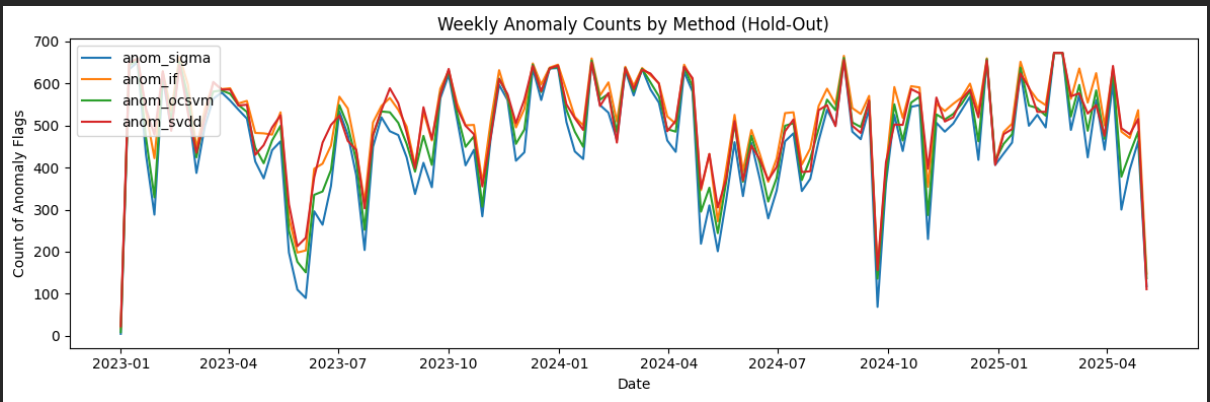

Figure 13 Weekly Anomaly Counts on Hold-Out Data by Detection Method

5.3.3 Cross-validation stability

Using the anomaly-safe splitter (Appendix D), we also computed fold-wise metrics for σ-threshold and Isolation Forest variants. The fold means and variances (see table Anomaly_CV_Metrics) indicate that σ-threshold behaviour is stable across time (small dispersion) whereas IF shows larger variance, especially when multivariate inputs include weak

or redundant features. This supports the practice of beginning with a simple, interpretable rule to set a sensitivity "floor", and only then introducing more expressive detectors with constraints to keep FAR within budget ((Xie, et al. 2023); (Xydas, et al. 2017)).

5.4 Comparative statistical tests (Diebold-Mariano)

To compare forecast accuracy between model pairs we use the Diebold-Mariano (DM) test as defined in Equation 8, applied to the loss differential $d_t = L_{1,t} - L_{2,t}$, (squared error). The DM statistic uses the sample mean of {dt} and a Newey-West (HAC) variance whereby large absolute values reject equal predictive accuracy see Equation 8.

Results in Table 4 and Table 5 are internally consistent with the RMSE story:

- Initial MC-Dropout LSTM vs Persistence: DM ≈ +130.98, p ≈ 0.0, indicating the LSTM's loss was significantly worse than persistence.
- Initial LSTM vs SES/ARIMA-GARCH: DM ≈ +45.22 and +48.08, again showing the initial deep model underperformed both classical references.
- Reduced-feature MC-Dropout LSTM v2 vs Persistence: DM ≈ +12.84, p ≈ 0.0; v2 remains statistically worse than the strong persistence baseline on this horizon.
- v2 vs ARIMA-GARCH / SES: DM ≈ −65.49 and −67.05, p ≈ 0.0, confirming v2 is significantly better than those baselines.

Table 4 Table of DM test on MC-LSTM with the full Feature space

Model 1	Model 2	DM Stat	p-value
MC_Dropout_LSTM	Persistence	130.97	0.0
MC_Dropout_LSTM	ARIMA_GARCH	48.07	0.0
MC_Dropout_LSTM	Exp_Smoothing	45.22	0.0
Persistence	ARIMA_GARCH	-65.33	0.0

whereby;

- DM Stat - is the Diebold-Mariano test statistic. Larger magnitude means a stronger difference and sign tells direction i.e positive means higher mean loss (worse) and negative ⇒ Model 1 has lower mean loss.
- p-value - The p-value indicates the probability of seeing a statistic this extreme if the two models were equally accurate in their predictions. In this study the p-value was so small ≈ 0.0 meaning we can reject equal accuracy and conclude a significant difference.

Table 5 Table of DM test on MC-LSTM with the reduced space

Model 1	Model 2	DM Stat	p-value
MC_Dropout_LSTM_v2	Persistence	12.84	0.0
MC_Dropout_LSTM_v2	ARIMA_GARCH	-65.49	0.0
MC_Dropout_LSTM_v2	Exp_Smoothing	-67.05	0.0
Persistence	ARIMA_GARCH	-65.33	0.0

These tests emphasise a pragmatic conclusion: on 15-minute net-load, a one-hour persistence is a formidable benchmark. Sophisticated models must clear that bar and deliver calibrated uncertainty to justify added complexity ((Xydas, et al. 2017); (Pierre Pinson 2017)).

5.5 Limitations (result-centred)

Two limitations are result-critical. First, the persistence baseline's strength is a function of horizon and sampling; at 15 minutes with a one-hour lag, it encodes much of the short-term structure. The same may not hold for longer horizons or different targets. Second, anomaly labels derived from σ-rules are proxies rather than ground truth; while useful for detector comparisons, they should be superseded by curated event logs or rule sets tuned to operator impact. These caveats should be read alongside the methodological constraints already stated (feature set, model class) and the calibration concerns discussed above ((Xie, et al. 2023); (Jakub Nowotarski 2018)).

6. Discussion and Conclusions

This chapter discusses what the study tried to do, how it did it, and what the results imply. The aim was twofold. First, forecast Irish net-load one hour ahead at a 15-minute cadence. Second, detect departures from that forecast that matter for operation. The setting is a grid with growing wind and solar, where short-term outcomes depend on weather. The method followed a simple plan. Clean the time index, standardise data access, start with baselines, add a compact sequence model, measure uncertainty by coverage, and detect anomalies in residuals rather than in the raw series. Evaluation used walk-forward folds to keep time order and to mimic deployment. Every step, features, models, and detectors were tested under the same splits. The pipeline choices were conservative by design to avoid leakage and to keep comparisons fair.

The main findings are direct. A one-hour-lag persistence rule is hard to beat on point error at this horizon. It captures smooth short-term structure and the daily cycle. Classical alternatives did not improve on that floor here. ARIMA with a GARCH variance layer had the right idea for bursts but did not track the quarter-hour series well without richer inputs. Simple exponential smoothing produced wide but not very helpful bands. The initial deep model failed because the input space was too large and noisy. With ~187 features, it over-fit and generalised poorly. After reducing the inputs to a compact set anchored in wind availability and short lags, the same architecture trained well, and errors dropped. Even then, on standardised splits it did not surpass persistence. This sets a clear bar for future work: extra complexity must earn its keep.

Uncertainty estimates told a similar story. Monte-Carlo dropout bands were over-confident at first; they improved after the feature cut but still fell below nominal coverage. This points to routine coverage checks and simple post-hoc calibration. Error structure was not uniform. Residuals were larger during ramp periods and in winter. Some short-lag dependence remained. These patterns favour small, targeted features over broad, noisy additions. Detection performance depended on threshold policy. Working in residual space focused alerts on what was unexpected given the model. A sigma rule gave a low-noise baseline. One-class detectors added value only when thresholds imposed a minimum recall and a false-alarm budget. Without that, alerts were too frequent.

Overall, the disciplined data spine of regular index, regional weather, leakage-safe features, and windowed access shaped the outcomes as much as the model choices. It removed early wins that were artefacts and left results that held up under out-of-sample checks.

6.1 Synthesis of Empirical Outcomes

This section draws a tight line from the data to the conclusions the rest of the chapter leans on. It compresses the evidence into the few statements that matter for an operator or a planner.

First, the bar. On 15-minute Irish net-load at a one-hour horizon, a one-hour-lag rule is a tough benchmark for point error. Any proposal for this horizon that increases complexity has to clear that bar or bring calibrated uncertainty that changes decisions. This is not a theoretical claim. It is a direct reading of the cross-validated and held-out numbers.

Second, the diet. Sequence forecasters do work here, but only on a lean input set. Wide, noisy features sink them; compact, physically plausible features steady them. The cut from ~187 inputs to a short list was the hinge between failure and a usable learner. This is both an empirical outcome and a design cue for future builds.

Third, the bands. Intervals produced by sampling at test time gave some signal but needed calibration. Empirical coverage did not match nominal targets out of the box. When bands are used to schedule headroom or set reserve, this gap matters. The data say "keep using this idea, but pair it with a routine coverage check and a light calibrator."

Fourth, the regimes. Errors and variance are not uniform in time. Two regimes dominate: within-day ramps and winter months. Models and detectors that condition on this simple structure behave better. This suggests that a small set of regime cues can lift both forecasting and detection without inviting the instability of large feature sets.

Fifth, the alerts. Residual-based alerts beat raw-series alerts on actionability. Plain sigma rules give a stable baseline with low noise; expressive one-class methods can add precision if and only if thresholds encode a budget for false alarms. When thresholds ignore cost, the detectors flood the operator; when thresholds honour cost, they become useful.

Sixth, the pipeline. The data spine strict quarter-hour index, regional aggregation of weather, leakage-safe features, windowed access explains as much of the success as the model choices. The hardening steps removed apparent "wins" that evaporated under discipline. What remains stands up to out-of-sample checks and time-respecting cross-validation.

6.2 Why the pipeline works the way it does

The decisions that shaped the pipeline were not stylistic. Each one fixed a failure mode or enforced a property that operators in real world usage would care about.

Start with time. A strict quarter-hour index that survives daylight-saving changes is not bookkeeping; it is the backbone of every window the models see. If time drifts, windows break. If windows break, leakage follows. The index routine enforced a single, gap-free spine and collapsed duplicates deterministically. That cure removed subtle artefacts that would otherwise inflate apparent accuracy and produce spurious anomalies.

Then geography. Station feeds are messy. A sensor goes offline; a gust at a coastal headland tells a tale that the inland load does not hear. Regional aggregation at quarter-hour cadence solved that. It smoothed idiosyncrasies and kept the synoptic moves that drive embedded wind. Circular means for direction fixed the 359°/1° problem. The result was not a perfect weather map; it was a stable set of cues the models could lean on.

Features came next. The “past-only” rule kept leakage out. Rolling statistics ended at t, not t+1. Calendar angles avoided discontinuities at wrap-around. Residual features were derived from forecasts made with information that would have existed at the origin time. The wide set in early experiments proved the rule in the breach: more inputs do not mean more signal. The funnel that reduced the set to a compact core was not a cosmetic step. It was a stability step.

The sequence forecaster sat on top of that structure. It took fixed-length windows, learned the short memory of the series, and fused exogenous cues. Dropout, left active at test time, produced a simple predictive spread. The approach scaled and fit the cadence, but it was only as good as its inputs and its calibration.

On detection, the residual frame did the real work. It concentrated “normal” near zero and reduced variance relative to the raw series. Simple thresholds on scaled residuals produced a sane baseline. Expressive one-class methods on a compact residual feature set then added structure. But none of that mattered if thresholds were set as if all errors cost the same. A recall floor and a budget for false alarms turned learning into policy. This move allowed the detectors to become usable.

Finally, evaluation. Walk-forward splits kept time order. An anomaly-safe splitter ensured each validation block had enough rare events to make precision and recall stable. Formal tests only adjudicated close calls; they did not stand in for engineering judgement. This discipline is why some results that looked enticing early did not survive. The study treats that as a success. Better to lose a “win” in the lab than to lose trust in the field.

6.3 Limits and threats to validity

Two limits define the bounds of what to carry forward.

Task and horizon. The findings are for quarter-hour samples and a one-hour horizon. The strong showing of persistence is partly a property of that regime. As horizons stretch, exogenous structure carries more of the load and the balance can shift. The conclusion is not “persistence always wins”; it is “persistence is strong here; test again where the horizon stretches.”

Labels and impact. Sigma-derived anomaly labels are proxies. They help compare detectors, but they do not encode operational cost. A large error during an evening ramp is not the same as a large error at 03:00. The study treats labels with care and tunes thresholds by a recall floor and a budget for false alarms, but the next step is to align labels and costs with operator impact so that evaluation and reality match.

There are also threats to validity that the pipeline tries to defuse: subtle index drift, late-arriving telemetry, and regime shifts in embedded wind. The safeguards are simple: strict time handling, “past-only” features, and retraining on a cadence that tracks drift. None of these remove risk, but they lower it and make failures traceable.

6.4 Carrying the Work Forward: Operational and Research Priorities

The next steps combine operational hardening with focused research and an emphasis on small, testable changes that keep behaviour stable while extending coverage.

Stage new grid variables. Post-2022 feeds arrived with uneven quality. Align Moyle and EWIC flows to the 15-minute grid with smooth penetration ratios and carry solar only when quality flags allow. Record known corrections (e.g., NI history) and avoid back-filling across structural breaks.

Strengthen uncertainty. Report coverage alongside RMSE by default. When helpful, learn quantile bounds directly, otherwise use a simple residual-based calibrator to align the bands with observed errors. Make calibration season-aware check monthly in winter and quarterly in summer. Use the bands for reserve decisions only when their empirical coverage matches the nominal level.

Target known regimes. Add a small set of cues that mark when errors rise i.e. morning/evening ramps, storm windows from regional wind persistence and gust spread, and holiday shifts. Retain only cues that improve winter ramps under walk-forward tests.

Work the residual memory. Allow a few lagged residuals and a local scale proxy to capture short memory the models do not clear. Keep gradients safe with clipping and modest learning-rate warm-ups. These settings already behave well and should remain default.

Thresholds that adapt. Treat thresholds as policy items with two numbers i.e. a recall floor and a false-alarm budget. Learn hour- and month-specific adjustments from the latest validation window. Keep the two-stage design: σ-screen first; calibrated OC-SVM or Deep SVDD second on candidates only.

Richer weather, carefully aligned. Add Numerical Weather Prediction (NWP) and satellite wind fields upstream of residuals to fill spatial gaps. Apply a short bias-correction and align to the 15-minute clock. When these sources are steady, consider a light graph encoder so the detector can learn spatial propagation across stations and interconnectors.

Stress and feedback. Build a small library of stress days (winter storms, curtailment episodes, steep evening ramps). Use them to test new models and threshold policies. Log every alert's outcome; use operator feedback to adjust budgets and small feature choices.

Labels that reflect cost. Start moving from σ labels to weighted labels that reflect hour buckets and event types such as HVDC constraints. Tune models and thresholds on these weights so evaluation matches operational impact.

Travel to new settings. Apply the pipeline to longer horizons and related targets where persistence is weaker. Use forecasted weather features, keep features compact, and reuse the same calibration and policy steps. As SNSP climbs, consider online updates and conformal methods so scores remain comparable through shifts.

These steps keep the current system steady in operations while opening measured avenues for improvement. The order is deliberate while safeguarding time and inputs first, standardise uncertainty next, then extend detection and weather sources with alignment and calibration in place.

6.5 Closing

This dissertation argued for modest tools used well over flashy tools used loosely. It built a clean time base, fused weather and operations on a common cadence, chose features that a line engineer can defend, and judged uncertainty by coverage rather than by hope. It showed that a strong baseline remains strong at this horizon. That a compact sequence forecaster can be helpful but must be held to the same standard; and that residual-based alerts work when thresholds reflect cost. It also showed that some early "wins" melt away under discipline and that this is a feature, not a bug, of an honest process.

If there is a single takeaway, it is this: reliability is a property of the whole pipeline. When the index is clean, features are leakage-safe, evaluation respects time, and thresholds encode cost, even everyday models behave in a way that the control room can use. The work does not end here. There are horizons to test, bands to calibrate, regimes to flag, and labels to align with impact. But the path is set. It runs through careful data handling, compact inputs, calibrated uncertainty, residual-aware alerts, and evaluation that mirrors deployment. Follow that path, and the grid's next hour becomes easier to see, and safer to run.

References

Ada Lau, Patrick McSharry. 2010. “Approaches for Multi-Step Density Forecasts with Application to Aggregated Wind Power.” *The Annals of Applied Statistics* 1311–1341. doi:10.1214/09-AOAS320.

Azam, Muhammad, Sadia Sahar, Rehman Sharif, Turki Alghamdi, Arshad Ali, Muhammad Uzair, and Mohammad Husain. 2025. “Uncertainty-Aware Energy Consumption Forecasting Using LSTM Networks with Monte Carlo Dropout.” *Informatica* 131–140 .

Bentsen, L. Ø., N. D. Warakagoda, R. Stenbro, and P. Engelstad. 2024. “Relative evaluation of probabilistic methods for spatio-temporal wind forecasting .” *Journal of Cleaner Production* 139944.

Bouman, R., L. Schmeitz, L. Buise, J. Heres, Y. Shapovalova, and T. Heskes. 2024. “Acquiring better load estimates by combining anomaly and change point detection in power grid time-series measurements.” *Sustainable Energy, Grids and Networks* 101540.

Degiannakis, Stavros, and Evdokia Xekalaki. 2003. *A Review Autoregressive Conditional Heteroscedasticity (ARCH) Models.* Technical Report, Athens, Greece.: Department of Statistics, Athens University of Economics and Business. doi: 10.1080/16843703.2004.11673078.

Deng, Z., X. Zhang, Z. Li, J. Yang, X. Lv, Q. Wu, and B. Zhu. 2024 . “Probabilistic prediction of wind power based on improved Bayesian neural network.” *Frontiers in Energy Research* 1–11.

Desheng Dash Wu, David L. Olson, Yue Wu. 2021. “Fuzzy vulnerability assessment of system susceptibility to pandemic using Z-number information.” *IEEE Transactions on Fuzzy Systems* 3829–3833. doi:10.1109/TFUZZ.2020.3008132.

EirGrid and ESB Networks. 2024. *Draft Annual Electricity Transmission Performance Report 2023.* Dublin, Ireland: EirGrid and ESB Networks.

EirGrid plc and SONI Limited. 2025. *Wind Dispatch Tool Constraint Group Overview.* Dublin, Ireland: EirGrid plc and SONI Limited.

EirGrid. 2025. *Public Consultation Webinar: Southcoast Offshore Engagement Plan Marine Usage License Application.* Dublin, Ireland: EirGrid.

European Committee of the Regions. 2022. “Opinion of the European Committee of the Regions - Amending the Renewable Energy Directive to meet the new 2030 climate targets.” *Official Journal of the European Union*. 5 August. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:C:2022:301:FULL.

Florin Bunea, Yiyuan She, Marten H. Wegkamp. 2011. “Adaptive Rank Penalized Estimators in Multivariate Regression.” *The Annals of Statistics* 1591–1642. doi:10.1214/11-AOS886.

Forbes, Kevin F. 2025 . “Lies, Damn Lies, and an Unethical Measure of Renewable Energy Predictability: The Case of Wind Energy Forecasting in Ireland.” *ResearchGate*. 15 January. https://www.researchgate.net/publication/388027176.

G, Preethi, and Anitha Kumari K. 2021 . *An Introductory Review Of Anomaly Detection Methods In Smart Grids* . Chennai, India: EAI (European Alliance for Innovation).

Gerard P. Nolan, Brendan P. G. Foley, Paul Cuffe. 2019. “Trends in system marginal price: The impact of wind generation on imbalance settlement prices in the Irish electricity market.” *Energy Policy* 485–497. doi:10.1016/j.enpol.2019.04.039.

Gneiting, Tilmann, Adrian E. Raftery, Anton H. Westveld III, and Tom Goldman. 2005 . “Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation .” *Monthly Weather Review* 1098–1116.

Habbak, H., M. Mahmoud, M. M. Fouda, M. Alsabaan, A. Mattar, G. I. Salama, and K. Metwally. 2023. “Efficient One-Class False Data Detector Based on Deep SVDD for Smart Grids .” *Energies* 7069.

Himeur, Y., K. Ghanem, A. Alsalemi, F. Bensaali, and A. Amira. 2021 . “Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives.” *Applied Energy* 116601.

Huang, Xianan, Lin Liu, Nuo Xu, Yantao Chen, Xiaofei Wang, and Zhenzhi Lin. 2025 . “Load Forecasting Using BiLSTM with Quantile Granger Causality: Insights from Geographic–Climatic Coupling Mechanisms .” *Applied Sciences* 5912.

Jakub Nowotarski, Rafał Weron. 2018. “Recent advances in electricity price forecasting: A review of probabilistic forecasting.” *Renewable and Sustainable Energy Reviews* 237–250. doi:10.1016/j.rser.2018.03.002.

Konstantinos Panapakidis, George Christoforidis. 2016. “Day-ahead electricity price forecasting via the application of artificial neural network based models.” *Applied Energy* 372–386. doi:10.1016/j.apenergy.2016.05.049.

Lang, S., C. Möhrlen, J. Jørgensen, B. Ó Gallachóir, and E. McKeogh. 2006. *Forecasting total wind power generation on the Republic of Ireland Grid with a Multi-Scheme Ensemble Prediction System*. Adelaide, Australia: Global Wind Energy Council .

Lau, A, and P. McSharry. 2010. “Approaches for Multi-Step Density Forecasts with Application to Aggregated Wind Power.” *The Annals of Applied Statistics* 1311–1341.

Lau, A., and P. McSharry. 2010. “Approaches for Multi-Step Density Forecasts with Application to Aggregated Wind Power.” *The Annals of Applied Statistics* 1311–1341 .

Lei, P., F. Ma, C. Zhu, and T. Li. 2024. “LSTM Short-Term Wind Power Prediction Method Based on Data Preprocessing and Variational Modal Decomposition for Soft Sensors.” *Sensors* 2521.

Li, Yanting, Zhenyu Wu, and Yan Su. 2023 . “Adaptive Short-term Wind Power Forecasting with Weather Drifts.” *SSRN (Social Science Research Network)* . 12 May. https://ssrn.com/abstract=4455442.

M. Browne, S. Poletti. 2020. “Electricity price forecasting: A review of the state-of-the-art with a look to the future.” *International Journal of Forecasting* 588–609. doi:10.1016/j.ijforecast.2019.07.008.

Milica Djukanovic, Goran Simic. 2016. “Electricity price forecasting: ARIMA model approach.” *Journal of Electrical Engineering* 233–240. https://doi.org/10.48550/arXiv.1603.02754.

Mosedale, T. J., D. B. Stephenson, M. Collins, and T. C. Mills. 2006. “Granger Causality of Coupled Climate Processes: Ocean Feedback on the North Atlantic Oscillation.” *Journal of Climate* 1182–1194.

N. Amjady, M. Hemmati, A. Abdollahi. 2010. “Market price forecasting of day-ahead electricity markets using a new hybrid neural network.” *Applied Energy* 120–129. doi:10.1016/j.apenergy.2010.06.004.

Pei, M., L. Ye, Y. Li, Y. Luo, X. Song, Y. Yu, and Y. Zhao. 2022. “Short-term regional wind power forecasting based on spatial–temporal correlation and dynamic clustering model.” *Energy Reports* 10786–10802.

Pierre Pinson, Henrik Madsen. 2017. “Benefits and challenges of electrical demand response: A critical review.” *Renewable and Sustainable Energy Reviews* 686–699. doi:10.1016/j.rser.2016.11.037.

Rajaperumal, T. A., & Christopher Columbus, C. 2025 . “Enhanced wind power forecasting using machine learning, deep learning models and ensemble integration .” *Scientific Reports* 1–20.

Shen, Zhiwei, and Matthias Ritter. 2015 . *Forecasting volatility of wind power production* . Berlin: SFB 649 Discussion Paper, Collaborative Research Center 649, Humboldt University of Berlin.

Shi, T., R. A. McCann, Y. Huang, W. Wang, and J. Kong. 2024. “Malware Detection for Internet of Things Using One-Class Classification.” *Sensors* 1–19.

Skouras, Spyros. 2001. “Financial returns, linear disaggregation, and estimation risk.” *International Journal of Forecasting* 227–239. doi:10.1016/S0169-2070(01)00079-1.

Spak, Brian. 2010. *The Success of the Copenhagen Accord and The Failure of the Copenhagen Conference*. Washington, DC: American University Washington College of Law .

Tao, Y., J. Yan, E. Niu, P. Zhai, and S. Zhang. 2025 . “An SVM-Based Anomaly Detection Method for Power System Security Analysis Using Particle Swarm Optimization and t-SNE for High-Dimensional Data Classification.” *Processes* 549.

Taylor, James W. 2010. “Density forecasting for the efficient balancing of the generation and consumption of electricity.” *International Journal of Forecasting* 706–727. doi:10.1016/j.ijforecast.2009.10.001.

The EirGrid Group. 2023. *The DS3 Programme: Delivering a Secure, Sustainable Electricity System*. Dublin, Ireland: EirGrid Group.

The European Parliament and the Council of the European Union. 2023. “ Directive (EU) 2023/2413 of the European Parliament and of the Council of 18 October 2023 amending Directive (EU) 2018/2001, Regulation (EU) 2018/1999 and Directive 98/70/EC as regards the promotion of energy from renewable sources, and repealing Council.” *Official Journal of the European Union*. 31 October. http://data.europa.eu/eli/dir/2023/2413/oj.

The European Parliament and the Council of the European Union . 2009. “Directive 2009/28/EC of the European Parliament and of the Council of 23 April 2009 on the promotion of the use of energy from renewable sources and amending and subsequently repealing Directives 2001/77/EC and 2003/30/EC.” *EUR-Lex – Official Journal of the European Union* . 5 June. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:140:0016:0062:EN:PDF.

Torben G. Andersen, Tim Bollerslev, Francis X. Diebold, Jin Wu. 2008. “Realized Beta: Persistence and Predictability.” *In Advances in Econometrics (Vol. 20, Part 2)* 1–39. doi:10.1016/S0731-9053(07)00002-5.

Wu, Z., G. Luo, Z. Yang, Y. Guo, K. Li, and Y. Xue. 2022. “A comprehensive review on deep learning approaches in wind forecasting applications .” *CAAI Transactions on Intelligence Technology* 129–143 .

Xia, Qinqin. 2025. “Market perspective on climate actions and clean energy transition.” *Energy Policy* 114470.

Xie, Yuying, Chaoshun Li, Mengying Li, Fangjie Liu, and Meruyert Taukenova. 2023. “An overview of deterministic and probabilistic forecasting methods of wind energy .” *iScience* 105804.

Xydas, E., M. Qadrdan, C. Marmaras, L. Cipcigan, N. Jenkins, and H. Ameli. 2017 . "Probabilistic wind power forecasting and its application in the scheduling of gas-fired generators." *Applied Energy* 382–394.

Young, Peter C. 2011. "Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner." *Springer Series in Statistics (Book Chapter)* 1–338. doi:10.1007/978-3-642-21981-3.

Zangrando, Niccolò, Piero Fraternali, Marco Petri, Nicolò Oreste Pinciroli Vago, and Sergio Luis Herrera González. 2022. *Anomaly detection in quasi-periodic energy consumption data series: a comparison of algorithms.* Vejle, Denmark : Energy Informatics.Academy Conference 2022 .

Appendices

Appendix A Building 15-Minute Regional Weather Features from 1-Minute Met Éireann Stations

This appendix documents the exact code used to turn raw 1-minute station CSVs from Met Éireann into a single, clean 15-minute dataset aggregated by Irish quadrants (NW/NE/SW/SE). The pipeline: (1) gap-fill with neighbor stations, (2) resample to 15-minute ("15T") with circular means for wind direction, (3) aggregate stations into regional features, and (4) window, round, and sanity-check the final frame.

Appendix A.1 Key term definitions (kept brief):

- 15T: pandas frequency code for 15-minute intervals.
- wdsp / wddir: wind speed (scalar) and wind direction (directional, in degrees).
- Circular mean: an average appropriate for angles; it averages sin/cos components to avoid wrap-around bias (e.g., 359° and 1° ≈ 0°, not 180°).
- Neighbor-based imputation: filling missing values using concurrent values from nearby stations.
- Quadrant aggregation: grouping stations into four regions (NW/NE/SW/SE) and averaging within each region to create robust features.

Appendix A.2 Neighbor-Aware Gap Filling (minute-level)

Fills missing temperature, wind speed, and wind direction per station using interpolation and neighbor means. Directional data are handled with a circular mean.

```
# Helpers
def to_numeric_inplace(df, cols):
    for c in cols:
        if c in df.columns:
            df[c] = pd.to_numeric(df[c], errors='coerce')

def _circular_mean_deg_series(x: pd.Series) -> float:
    """Circular mean in degrees for a 1D series (ignores NaNs)."""
    vals = np.deg2rad(x.dropna().values.astype(float))
    if vals.size == 0:
        return np.nan
    mean = np.arctan2(np.nanmean(np.sin(vals)), np.nanmean(np.cos(vals)))
    ang = np.rad2deg(mean) % 360.0
    return ang

def circular_mean_deg(x):
    """
    Flexible circular mean:
    - If DataFrame: row-wise circular mean across columns (returns Series).
    - If Series: scalar circular mean (returns float).
    """
    if isinstance(x, pd.DataFrame):
        return x.apply(_circular_mean_deg_series, axis=1)
```

```
    return _circular_mean_deg_series(x)

def fill_from_neighbors(code, df, stations, neighbours_map):
    # Coerce numeric and sort
    to_numeric_inplace(df, ['temp','wdsp','wddir'])
    df = df.sort_index()

    # Neighbor frames, aligned to this station's index
    n_codes = neighbours_map.get(code, [])
    ndfs = [stations[nc].reindex(df.index) for nc in n_codes if nc in stations]

    # Temperature: time interpolate, then neighbor mean, then ffill/bfill
    if 'temp' in df.columns:
        df['temp'] = df['temp'].interpolate(method='time')
        if ndfs:
            temp_df = pd.concat([(n['temp'] if 'temp' in n.columns else pd.Series(index=df.index, dtype=float))
                                 for n in ndfs], axis=1)
            df['temp'] = df['temp'].fillna(temp_df.mean(axis=1, skipna=True))
        df['temp'] = df['temp'].ffill().bfill()

    # Wind speed: neighbor mean, then time interpolate, then ffill/bfill
    if 'wdsp' in df.columns:
        if ndfs:
            ws_df = pd.concat([(n['wdsp'] if 'wdsp' in n.columns else pd.Series(index=df.index, dtype=float))
                               for n in ndfs], axis=1)
            df['wdsp'] = df['wdsp'].fillna(ws_df.mean(axis=1, skipna=True))
        df['wdsp'] = df['wdsp'].interpolate(method='time').ffill().bfill()

    # Wind direction: circular neighbor mean, then ffill/bfill
    if 'wddir' in df.columns and ndfs:
        wd_df = pd.concat([(n['wddir'] if 'wddir' in n.columns else pd.Series(index=df.index, dtype=float))
                           for n in ndfs], axis=1)
        wd_filled = circular_mean_deg(wd_df)
        df['wddir'] = df['wddir'].fillna(wd_filled).ffill().bfill()

    return df

# Apply across all stations
for code, df in stations.items():
    stations[code] = fill_from_neighbors(code, df, stations, neighbours_map)
```

Appendix A.3 Resample to 15 Minutes (scalar means, circular mean for direction)

Creates 15-minute frames for each station. Scalars use arithmetic mean; directions use circular mean inside each 15-minute bin.

```
def resample_15min(df):
    # Identify which variables are present
    scalars = [c for c in df.columns if c in ['temp','wdsp']]
    dirs    = [c for c in df.columns if c == 'wddir']

    out = pd.DataFrame(index=pd.date_range(
        df.index.min().ceil('15T'),
        df.index.max().floor('15T'),
        freq='15T'
    ))

    if scalars:
        out[scalars] = df[scalars].resample('15T').mean()
```

```
    if dirs:
        # Circular mean per 15-min bin
        grouped = df['wddir'].resample('15T')
        out['wddir'] = grouped.apply(lambda g: _circular_mean_deg_series(g))

    return out

stations_15T = {code: resample_15min(df) for code, df in stations.items()}
```

Appendix A.4 Regional Aggregation (NW/NE/SW/SE)

Aggregates stations into four regional features using means for temp/wdsp and a circular mean for wddir. Uses an inner join on time so all regions share the same, complete 15-minute index.

```
clusters = {
    "NW": ["1175_NEWPORT", "1275_MARKREE", "2175_CLAREMORRIS",
           "2375_BELMULLET", "2727_CLAREMORRIS", "4935_KNOCK_AIRPORT"],
    "NE": ["532_DUBLIN_AIRPORT", "675_BALLYHAISE", "875_MULLINGAR",
           "1375_DUNSANY", "1575_MALIN_HEAD", "175_PHOENIX_PARK",
           "1975_MT_DILLON", "2075_FINNER"],
    "SW": ["275_MACE_HEAD", "518_SHANNON_AIRPORT", "775_SherkinIsland",
           "1875_ATHENRY", "2125_MACE_HEAD", "3402_SHERKIN_ISLAND",
           "3904_CORK_AIRPORT", "575_MOORE_PARK", "2275_VALENTIA_OBSERVATORY"],
    "SE": ["375_OAK_PARK", "1004_ROCHES_POINT", "1075_ROCHES_POINT",
           "1475_GURTEEN", "3723_CASEMENT"]
}

def code_from_tag(tag: str) -> int:
    return int(tag.split('_')[0])

# Common time index across all station 15T frames (inner join)
common_index = None
for df in stations_15T.values():
    common_index = df.index if common_index is None else common_index.intersection(df.index)
common_index = common_index.sort_values()

# Aggregate by region
region_15T = pd.DataFrame(index=common_index)

for region, station_tags in clusters.items():
    codes = [c for c in (code_from_tag(t) for t in station_tags) if c in stations_15T]
    if not codes:
        continue

    aligned = [stations_15T[c].reindex(common_index) for c in codes]

    # temp & wdsp: arithmetic mean
    if any('temp' in a.columns for a in aligned):
        region_15T[f'{region}_temp'] = pd.concat([a['temp'] for a in aligned if 'temp' in a.columns],
axis=1).mean(axis=1)
    if any('wdsp' in a.columns for a in aligned):
        region_15T[f'{region}_wdsp'] = pd.concat([a['wdsp'] for a in aligned if 'wdsp' in a.columns],
axis=1).mean(axis=1)

    # wddir: circular mean across stations at each timestamp
    if any('wddir' in a.columns for a in aligned):
        wddir_df = pd.concat([a['wddir'] for a in aligned if 'wddir' in a.columns], axis=1)
```

```
    region_15T[f'{region}_wddir'] = circular_mean_deg(wddir_df)
```

Appendix A.5 Windowing, Rounding, and Integrity Checks

Restricts the time span, rounds values, and verifies basic integrity.

```
start = pd.Timestamp("2014-01-01 00:00:00")
end   = pd.Timestamp("2025-12-31 23:59:59")

region_15T = region_15T.loc[(region_15T.index >= start) & (region_15T.index <= end)].copy()

# Round: keep 5 dp for comparability; adjust if downstream tooling prefers fewer
region_15T = region_15T.round(5)

# Quick checks
print("Index freq approx:", pd.infer_freq(region_15T.index[:100]) or "irregular")
print("Any NaNs?", region_15T.isna().any().any())
```

Appendix A.6 Outputs.

region_15T contains 15-minute regional features: {NW, NE, SW, SE}_temp, {NW, NE, SW, SE}_wdsp, and {NW, NE, SW, SE}_wddir. These feed directly into the forecasting and anomaly-detection notebooks as quadrant-level meteorological predictors.

Appendix B Time-Index Regularization & Duplicate Handling

This appendix documents the routine that standardizes the time axis to a strict 15-minute cadence, eliminates duplicate timestamps, and produces a clean, gap-aware index for all downstream models. It also records the validation checks, known edge cases (notably daylight-saving time (DST)), and the rationale for each step to show how to reproduce the splits and avoid target leakage.

Appendix B.1 Definitions

- Cadence (15-minute cadence): every row is exactly 15 minutes after the previous one no irregular steps.
- Duplicate timestamp: two or more rows that share the same DateTime value; models and resampling break if these aren't resolved.
- DST forward/back: clock jumps (e.g., spring forward skips times; autumn back repeats an hour) that can create missing or duplicated local timestamps.
- Target leakage: using information from the future when building features for the present; irregular or duplicated time axes can accidentally cause this.

Appendix B.2 Inputs & Outputs

- Input: a DataFrame with a DateTime column (local naive timestamps) and all feature/target columns.
- Output (Stage 1): the same rows, but with a de-duplicated, strictly increasing DateTime (no duplicates).
- Output (Stage 2): a full 15-minute grid from min to max time (missing slots materialized) so later models can safely interpolate/fill as needed.

Appendix B.3 Procedure

Appendix B.3.1 De-duplicate while preserving row count

1. Detect duplicates: Mark duplicates in DateTime (duplicated(keep=False)) to quantify and review.
2. Sort & normalize type: Cast DateTime to datetime64[ns], sort ascending.
3. Construct a clean sequential index: Walk the sorted rows with a pointer current_expected_timestamp:
    - If the original time is ahead of the pointer, reset the pointer to that time (don't synthesize gaps here).
    - Assign DateTime1 = current_expected_timestamp.
    - Increment pointer by 15 minutes and continue.

4. Swap in the clean index: Replace original DateTime with DateTime1 and set it as the index.
5. Verify invariants: No duplicates; strictly increasing; asfreq('15min') preserves length.

Appendix B.3.2 Build a full 15-minute grid (gap-aware)

1. Create full grid date_range(min_time, max_time, freq='15min'), then reindex onto it.
2. Fill policy
    - Meteorology (e.g., *_temp, *_wdsp, *_wddir): time interpolation (with circular mean for directions), then forward/back fill for isolated holes.
    - Lags/rolling stats: short runs back-filled; longer runs interpolated to avoid biasing early windows.
3. Track synthesized values Keep a boolean mask of filled points to inform uncertainty analysis and model diagnostics.

Appendix B.4 Quality checks (run and record)

- Duplicates: index.duplicated().sum() == 0.
- Frequency: asfreq('15min') succeeds; infer_freq on head/tail ≈ '15T'.
- Window/target bounds: with seq_len and horizon applied, t + horizon ≤ max_ts.
- Known expected NaNs before filling:
    - Start-of-series: first hour lacks lags/persistence → expected NaNs in pers_1h at the first 4 slots.
    - DST spring-forward: missing 01:00–01:45 local times create 4 more NaNs.
      In the data these occurred at:
    - 2014-01-01 00:00, 00:15, 00:30, 00:45
    - 2014-03-30 01:00, 01:15, 01:30, 01:45

Appendix B.5 Edge cases & guidance

- DST: Using naive local timestamps treats DST gaps/overlaps as ordinary missing/duplicate times handled by the two-stage flow. If you require strict timezone semantics, convert to UTC before Stage 1 and keep a mapping table.
- Outages vs. interpolation: Stage 1 never fabricates rows; Stage 2 does, under an explicit fill policy. Document which columns were interpolated vs. carried forward.
- Leakage guard: After Stage 2, ensure features are built only from past rows relative to each timestamp and that targets are strictly horizon-ahead.

Appendix C HDF5-Backed Windowed Dataset (HDF5WindowDataset & v2)

This appendix explains how long, multivariate time series are converted into an HDF5 store and then streamed to the model as fixed-length windows with horizon-ahead targets. It documents the storage layout (columns, index, chunking), the dataset interfaces (v1 vs v2), and the safeguards that keep training scalable and reproducible.

Appendix C.1 Definitions

- HDF5 store: a single on-disk file that holds arrays and metadata, allowing fast partial reads (no need to load everything into RAM).
- Window (sequence): the past seq_len time steps used as model input.
- Horizon: how far into the future the target lies (e.g., 4 steps = 1 hour at 15-min cadence).
- Chunking: how the data are split internally so that reading a window is efficient.
- Lazy open (per worker): each DataLoader worker opens the file only when it actually needs it, reducing contention and memory use.

Appendix C.2 Inputs & Outputs

- Input: a feature-engineered DataFrame with a regular 15-minute index and numeric columns; plus the name of the target column.
- Output (v1): HDF5 file with three datasets:
    - "data" → 2-D float array shaped (rows, n_features)
    - "index_ns" → 1-D int64 array of timestamps in nanoseconds since epoch
    - "columns" → list of feature names (the target is still among these columns)
- Output (v2): HDF5 file with separate target:
    - "data" → features only
    - "target" → 1-D float array for the target
    - "index_ns", "columns" → as above, with "columns" listing feature names (target name provided externally)

Appendix C.3 Procedure

Phase A creates an efficient on-disk representation (downcasted dtypes, window-friendly chunking). Phase B consumes that store via dataset classes that assemble windows and horizon targets on the fly. Use v2 when you want the cleanest separation of features and target or when swapping targets without rewriting the whole store.

Appendix C.3.1 Phase A Build the HDF5 store (export)

1. Downcast dtypes: convert float64 → float32 and large integers to compact ints to shrink file size and speed I/O.
2. Preserve the time axis: save index_ns (int64) so any position ↔ timestamp mapping is exact.
3. Choose chunking that matches windows: set HDF5 chunks to (seq_len, n_features) so each window read hits as few chunks as possible.
4. Record schema: write a "columns" dataset (UTF-8 strings) for full provenance; in v2, write "target" as its own dataset.

Appendix C.3.2 Phase B Stream windows efficiently (training/inference)

1. Map timestamps to positions: convert each fold's timestamp indices to integer positions using the saved index; drop positions too close to the start/end where a full window or horizon is impossible.
2. Lazy open per worker: in __getitem__, open the file on first use inside each worker; keep a handle for subsequent reads.
3. Assemble window & target: for position p, read [p−seq_len, …, p−1] from "data" as X; read the target at p + horizon (from the target column in v1 or from "target" in **v2`).
4. Batching & padding policy: iterate with a sequential sampler (time order preserved), then shuffle batches (not samples) to retain locality while improving GPU utilization. Only pad if you intentionally allow variable seq_len.
5. GPU-friendly I/O: enable pinned memory and prefetch in the DataLoader; keep batch shapes (B, seq_len, n_features).

Appendix C.4 Quality checks (run and record)

- Schema parity: the number and order of "columns" match the feature matrix; in v2, verify the target's length equals rows.
- Index integrity: len(index_ns) == rows, strictly increasing, and round-trips to timestamps.
- Chunk alignment: seq_len used at training time matches the HDF5 chunk's first dimension.
- Window bounds: for every sampled position p, ensure p−seq_len ≥ 0 and p + horizon < rows.
- Determinism: given the same split indices, the dataset yields identical windows/targets across runs.

Appendix C.5 Edge cases & guidance

- Changing seq_len after export: reading remains correct but may span multiple chunks; for peak performance, re-export with the new seq_len.

- Feature drift: if you add/remove columns, regenerate the HDF5 file to keep "columns" authoritative.
- Multiple targets: prefer v2 and store additional targets in separate datasets to avoid rewriting "data".
- Cross-platform paths: keep POSIX-style paths when working under WSL; document mount points used during experiments.

Appendix D Anomaly-Safe Time-Series Cross-Validation Splitter

This appendix describes the custom CV splitter used to evaluate anomaly detectors on a chronological series while guarding against information leakage and anomaly scarcity. It enforces time order, applies a gap/embargo around split boundaries, and guarantees a minimum number of anomalies in every validation fold so metrics are stable and comparable.

Appendix D.1 Definitions

- Anomaly (rare event): a timestamp flagged by the reference rule (e.g., σ-threshold on residuals) or curated labels used as ground truth.
- Rolling-origin CV: training grows forward in time; each fold validates on the next block of future data.
- Embargo / gap: samples near the split boundary that are removed to prevent leakage via overlapping windows.
- seq_len / horizon: past window length used as model input, and the number of steps ahead the prediction targets.
- Minimum anomaly count: the least number of positives each validation fold must contain (e.g., ≥ N) for reliable precision/recall/F1.

Appendix D.2 Inputs & Outputs

- Input: a time-indexed DataFrame; a boolean anomaly label series (or derivation rule); cadence (15-min); seq_len and horizon; desired number of folds; min anomalies per fold; optional seasonal/regime tags.
- Output: a list of folds, each a pair of timestamp indices (train_idx, val_idx), plus fold metadata (date ranges, anomaly counts, class balance).

Appendix D.3 Procedure

Phase A plans fold boundaries and feasibility (do we have enough anomalies per window?), while Phase B materializes valid timestamp indices that respect seq_len, horizon, and leakage constraints. Phase A's plan drives Phase B's index construction; if counts fall short, Phase B triggers a controlled adjustment and feeds back to Phase A until constraints are met.

Appendix D.3.1 Phase A: Scan & Plan (feasibility & boundaries)

1. Chronological scaffold: propose rolling-origin folds (e.g., 2-year expanding train → 1-year validation), aligned to the 15-min cadence and DST-safe index.
2. Anomaly inventory: compute anomaly counts per candidate validation block; note seasonal/regime coverage.
3. Set safeguards: define an embargo equal to max(seq_len, horizon) (in time) to be removed around each boundary.

4. Feasibility check: ensure each validation block has ≥ min anomalies after embargo; if not, widen the block, shift its start, or merge adjacent periods per a deterministic rule (e.g., extend by months until satisfied).

Appendix D.3.2 Phase B: Materialize Indices (leak-free, window-ready)

1. Edge filtering: remove timestamps that cannot form a full window or target (t - seq_len ≥ start, t + horizon ≤ end).
2. Apply embargo: drop boundary-adjacent samples from both train and validation sides to avoid overlap of source windows.
3. Anomaly guarantee: re-count; if a fold dips below the threshold (e.g., because edge filtering removed positives), adjust deterministically (extend the validation tail or shrink the embargo if safe) and re-emit indices.
4. Stratified sanity (optional): check seasonal/regime tags; if a fold is extremely skewed, allow Phase A to rebalance by shifting its window within defined limits.
5. Emit fold pack: (train_idx, val_idx) plus metadata (date span, counts, prevalence).

Appendix D.4 Quality checks

- No leakage: training windows never touch validation windows; embargo ≥ seq_len.
- Time order preserved: all validation timestamps are strictly after training.
- Anomaly sufficiency: each validation fold meets or exceeds the configured minimum.
- Window validity: every val_idx supports a full seq_len history and horizon target.
- Coverage balance: folds collectively cover seasons/regimes seen in the hold-out period.
- Determinism: same inputs → identical folds (no RNG dependence).

Appendix D.5 Edge cases & guidance

- Very sparse anomalies: prefer extend-forward of validation windows over shrinking the embargo; document final widths.
- Burst anomalies (clusters): allow slightly wider gaps to stop cluster tails from leaking into training via windows.
- Label revisions: if the ground truth rule changes (e.g., different σ), regenerate folds so counts and embargo are consistent.
- Different horizons: larger horizon increases the embargo; re-plan Phase A accordingly.
- Irregular clock (DST): keep a naive, regularized 15-minute index upstream; never split across missing/duplicated wall-times.

Appendix E MC-Dropout Inference & Uncertainty Aggregation

This appendix explains how Monte-Carlo (MC) dropout is used at inference time to obtain predictive means and uncertainty bands from the LSTM forecaster. Instead of a single deterministic pass, the model performs multiple stochastic passes with dropout active; the collection of outputs is then summarized into a mean forecast and a dispersion estimate used for interval coverage.

Appendix E.1 Definitions

- Dropout (at test time): randomly deactivating a fraction of units during inference to sample from an approximate posterior over model weights.
- MC samples: the number of stochastic forward passes (e.g., 50) used to estimate the predictive distribution.
- Predictive mean / std: the average and standard deviation of the MC sample predictions for each timestamp.
- Coverage (80% / 95%): fraction of true outcomes that fall inside the nominal prediction interval (e.g., mean ± z·std with z≈1.282/1.960 for 80%/95%).
- Aleatoric vs. epistemic: data noise vs. model uncertainty; MC-dropout primarily captures epistemic uncertainty.

Appendix E.2 Inputs & Outputs

- Input: trained LSTM with dropout layers; evaluation windows (seq_len) and targets (horizon); chosen MC sample size; z-scores for desired intervals; calibration set.
- Output: per-timestamp mean prediction, std deviation, and intervals (e.g., 80%, 95%); summary metrics such as RMSE and empirical coverage.

Appendix E.3 Procedure

Phase A makes the model stochastic at inference and generates a consistent set of MC predictions. Phase B turns those samples into actionable uncertainty summaries and evaluates their reliability (coverage). Phase A feeds a matrix of samples into Phase B; if calibration is off, Phase B can feed back scaling guidance to Phase A.

Appendix E.3.1 Phase A Enable stochastic inference & sample consistently

1. Activate dropout at test time: place all dropout layers in training mode only during inference passes while keeping the rest of the model evaluation-safe.
2. Fix randomness where needed: set seeds (framework + NumPy) for reproducibility across runs; keep batch size stable to avoid nondeterministic kernels.
3. Choose MC budget: select n_samples to balance variance of the std estimate and compute cost (typical: 30–100).

4. Batch the windows: run the validation/test windows in batches; for each batch, perform n_samples stochastic passes and collect the predictions into a samples matrix S of shape (n_samples, batch).
5. Persist minimal diagnostics: store per-timestamp sample mean, sample std, and skew/kurtosis for later analysis.

Appendix E.3.2 Phase B Aggregate, form intervals, and assess calibration

1. Compute summaries: per timestamp, mean = mean(S) and std = std(S).
2. Form nominal intervals: for target coverage α, set zα (e.g., 1.282 for 80%, 1.960 for 95%) and compute [mean − zα·std, mean + zα·std].
3. Evaluate accuracy & reliability: compute RMSE for point accuracy; compute empirical coverage by checking whether the true value lies inside each interval.
4. Calibration: if empirical coverage is below nominal, apply a global scale factor to std (temperature scaling for variance) estimated on a calibration split, then recompute intervals.
5. Report: provide per-split coverage tables and simple reliability plots (nominal vs. empirical).

Appendix E.4 Quality checks

- Stochasticity verified: repeated single-point inference yields different draws but similar means.
- Convergence of std: increasing n_samples changes std estimates only marginally (law of large numbers).
- Coverage sanity: 95% band is not narrower than 80%; empirical coverage rises with band width.
- Reproducibility: same seed, same inputs ⇒ identical mean/std results.

Appendix E.5 Edge cases & guidance

- Heteroskedastic targets: if dispersion varies with regimes (e.g., season), consider regime-specific calibration factors.
- Distributional shape: if sample distributions are heavy-tailed or skewed, prefer empirical quantiles from S over Gaussian z-scaling for intervals.
- Compute budget: when throughput is constrained, reduce n_samples but monitor the stability of coverage; alternatively cache per-batch dropout masks for reuse.
- Combination with variance proxies: std from MC-dropout reflects model uncertainty; if you also estimate noise via residual GARCH-like proxies, document whether bands are model-only or combined (e.g., sum in variance space).