Environmental Monitoring in New Zealand: Exploring Spatiotemporal Relationships

Marcos V. Ferreira[1], Luiz C. D. Cavalcanti[1], Tatiane N. Rios[1], Guilherme W. Cassales[2], Nick J. S. Lim[2], Albert Bifet[2], Ricardo A. Rios[1],

[1]Institute of Computing, Federal University of Bahia, Salvador, Brazil
[2]AI Institute, University of Waikato, Hamilton, New Zealand
marcosvsf@ufba.br, luizcdc@ufba.br, tatiane.nogueira@ufba.br guilherme.cassales@waikato.ac.nz, nick.lim@waikato.ac.nz, abifet@waikato.ac.nz, ricardoar@ufba.br

Abstract

The accelerating pace of global warming, confirmed by the Intergovernmental Panel on Climate Change (IPCC), underscores the urgent need for continuous environmental monitoring and adaptive modeling frameworks. Therefore, this study explores AI-driven temperature forecasting across New Zealand using real observations from the national MetService network within the TAIAO data science programme. We evaluate a diverse set of models, from statistical baselines to deep neural networks, foundation models, and graph-based architectures, to assess their capacity for adaptive, spatially aware prediction. Our results show that graph-based representations substantially improve the modeling of spatial and temporal dependencies, while foundation models demonstrate robust generalization across diverse climatic regions. The integration of these paradigms produces forecasts that are both more accurate and more interpretable. The findings highlight the potential of adaptive AI frameworks to improve environmental monitoring, detect regional anomalies, and strengthen climate resilience strategies in New Zealand.

Introduction

The Intergovernmental Panel on Climate Change (IPCC), a United Nations body for assessing the science related to climate change, concluded that human activities have unequivocally caused global warming, with global surface temperature about 1.0 °C above 1850–1900 in 2011–2020 (Change 2023). According to the IPCC, global greenhouse gas emissions continue to rise, with unequal historical and ongoing contributions driven by unsustainable energy use, land use and land-use change, and by lifestyles and patterns of consumption and production across regions, between and within countries, and among individuals. The IPCC further assesses that 1.5 °C of warming is expected to be reached in the near term, most likely in the early 2030s across considered scenarios. However, annual temperatures have recently approached or exceeded this level: 2023 averaged 1.45 °C above pre-industrial, and 2024 was confirmed as the warmest year on record at 1.55 °C. Specifically for New Zealand, the latest Climate Change Performance Index (CCPI) reports that the country dropped seven places to 41st. While it scores highly on renewable energy, its overall rating for climate policy is low [1].

Beyond strongly advising governments and the public about the risks associated with this issue, it is important to continuously monitor the environment through systematic data collection and adaptive Artificial Intelligence (AI) model adjustments, enabling a deeper understanding and more accurate prediction of the progression of temperature increase. In this context, recent advances in AI have profoundly transformed environmental data analysis, enabling the combination of Statistical Modeling and Machine Learning to better capture the complexity, uncertainty, and spatiotemporal dynamics of natural systems.

Traditional statistical methods, such as regression analysis, remain essential for correcting systematic biases in numerical weather prediction and quantifying forecast uncertainty. Building on these foundations, deep learning approaches have expanded the modeling capacity for complex environmental phenomena, allowing for the extraction of nonlinear relationships and high-dimensional feature representations from heterogeneous data sources. Recently, the emergence of Foundation Models has pushed this integration further, introducing general-purpose architectures capable of addressing multiple environmental and weather-related tasks through self-supervised and in-context learning. Models such as ClimaX (Nguyen et al. 2023) and WeatherGFM (Zhao et al. 2025) exemplify this trend by leveraging large-scale environmental datasets and adaptive fine-tuning strategies to enhance generalization across domains. In parallel, graph-based learning has brought a new layer of spatial reasoning to environmental modeling. Such methods integrate hierarchical graph structures and latent-variable formulations to produce spatially coherent ensemble forecasts with improved probabilistic accuracy (Oskarsson et al. 2024). Likewise, researchers have leveraged Graph Neural Networks (GNN) to handle irregularly sampled time series with missing values, achieving notable gains in predictive accuracy and computational efficiency (Yalavarthi et al. 2024). In the agricultural domain, graph-based recurrent neural networks have also been successfully applied to crop yield prediction, effectively incorporating geographical and temporal dependencies to outperform traditional statis-

[1]https://ccpi.org/country/nzl/

tical and deep learning baselines (Fan et al. 2022).

In this work, we conduct an exploration of AI models to forecast temperature evolution across New Zealand, using data collected by the national weather service (MetService) and shared through the TAIAO data science programme. As illustrated in Figure 1, the environmental monitoring network in New Zealand comprises multiple temperature measurement stations distributed across the country, each recording its own temporal dynamics influenced by local microclimates and regional circulation patterns. These stations form a spatial graph where nodes represent sensor locations and edges capture their geographical and climatic proximity, enabling the modeling of inter-station dependencies through GNNs and other spatial learning techniques. This spatiotemporal representation allows adaptive AI models to jointly learn spatial correlations and temporal evolutions, thereby improving short- and long-term temperature forecasting accuracy.

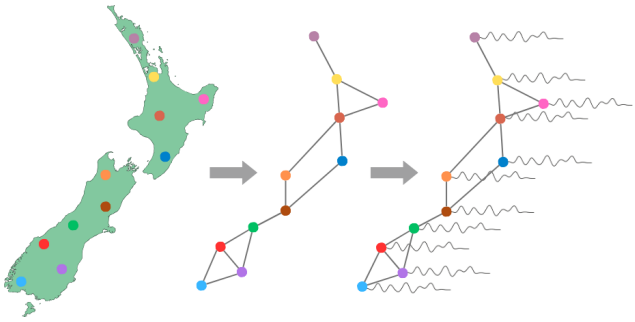


Figure 1: Environmental monitoring network in New Zealand

By integrating continuous data collection, statistical correction, and advanced AI-based inference, this framework supports the development of predictive systems capable of detecting regional anomalies, understanding the propagation of temperature variations, and ultimately contributing to more precise environmental monitoring and climate resilience strategies in New Zealand. To empirically assess these capabilities, we conducted a comprehensive series of experiments across multiple modeling paradigms, ranging from classical statistical baseline (SARIMA) to deep learning architectures (LSTM, GRU), foundation models (Chronos and Times), and graph-based neural networks (GCN, SAGE, Cheb, GAT and LEConv). The experiments were performed using real temperature data collected from monitoring stations distributed throughout New Zealand, as illustrated in Figure 1.

Overall, the main findings and contributions of this study can be summarized as follows:

- Graph-based representations enhanced the modeling of spatial and temporal dependencies, improving the understanding of temperature propagation across monitoring stations.
- Foundation Models showed strong generalization across diverse climatic regions through large-scale pretraining.
- A comparative evaluation of statistical, deep learning, foundation, and graph-based models was conducted using real temperature data from New Zealand.
- Graph-based models achieved the highest forecasting accuracy while maintaining spatial interpretability.
- The results highlight the potential of adaptive, data-driven systems to strengthen environmental monitoring and climate resilience.

TAIAO Dataset

TAIAO is a New Zealand data science programme funded by the Ministry of Business, Innovation and Employment. It aims to make environmental data more accessible and usable for researchers, scientists and the wider community. The main goal of TAIAO is to advance the state of the art in environmental data science by developing machine learning methods for time series and data streams that can process large-scale data in real time, tailored to data collected from New Zealand's environment.

The dataset used in this work was obtained from MetService (Te Ratonga Tirorangi), New Zealand's national meteorological service, and consists of hourly observations from weather stations across New Zealand covering January 1993 to August 2022. For each station, the available features include: wind direction over the previous 10 minutes, in degrees clockwise from geographic north ($^\circ$T); average wind speed over the previous 10 minutes (kt); maximum wind gust during the hour preceding the observation time (kt); average visibility during the previous 10 minutes (metres or kilometres); present weather; cloud cover; air temperature measured in a screen typically 1.5 m above ground level ($^\circ$C); dew-point temperature ($^\circ$C) and relative humidity (%), measured under the same screen conditions; atmospheric pressure corrected to mean sea level using ambient temperature (hPa); hourly rainfall (mm); and hourly solar radiation (MJ/m^2).

These data were collected from 51 stations distributed across New Zealand. Consequently, in addition to temporal information, the available geospatial detail supports methods that account for neighborhood influences on localized weather, enabling spatiotemporal modelling and analysis.

AI Models for Environmental Data

In the following subsections, we detail the main AI models applied to the TAIAO dataset, organized into four complementary perspectives: (i) Statistical approaches, which form the basis for uncertainty quantification and bias correction; (ii) Deep Neural Networks, which capture complex nonlinear relationships in environmental data; (iii) Foundation Models, which generalize across heterogeneous environmental tasks; and (iv) Graph Neural Networks, which explicitly represent spatial and relational dependencies in environmental systems.

Statistical approach

Statistical methods have long been used to model univariate time series by identifying key patterns such as trend, seasonality, and random noise. Among these, the Seasonal Autore-

gressive Integrated Moving Average (SARIMA) model (Box et al. 2015; Shumway and Stoffer 2011) is one of the most established approaches. SARIMA combines three main components: the Autoregressive (AR) process, which models dependencies on past values; the Moving Average (MA) process, which captures correlations with past forecast errors; and Integration (I), which removes long-term trends. Additionally, seasonal differencing allows the model to handle periodic patterns effectively.

Mathematically, an AR model uses p past observations ($x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + \varepsilon_t$), while an MA model uses q past random shocks ($x_t = x_{t-1} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$). Combining both results in the ARMA model, and when extended with seasonal components, it forms SARIMA, a flexible framework capable of modeling diverse temporal behaviors including trends, cycles, and seasonality (Shumway and Stoffer 2011).

Deep Neural Network

Unlike classical statistical models that explicitly decompose a time series into deterministic and stochastic components, Deep Neural Networks (DNNs) learn temporal dependencies directly from data without predefined assumptions about trend or seasonality. Traditional Artificial Neural Networks (ANNs), however, are typically designed for independent and identically distributed (i.i.d.) data and thus cannot inherently capture the temporal correlations present in sequences x_t.

To overcome this limitation, Recurrent Neural Networks (RNNs) introduce a hidden state h_t that evolves over time, enabling the network to model dependencies between past and present observations, formally represented as $h_t = f(x_t, h_{t-1})$. Among RNN variants, the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Unit (GRU) (Cho et al. 2014) architectures are the most widely used. Both employ gating mechanisms (e.g., input, forget, and output gates) to control information flow across time steps and mitigate the vanishing and exploding gradient problems encountered during Backpropagation Through Time (BPTT) (Werbos 1990).

These architectures allow DNNs to effectively capture long-range temporal dependencies and nonlinear relationships in time series data, making them well-suited for complex environmental modeling tasks compared to traditional autoregressive approaches.

Foundation Model

Foundation Models (FMs) represent a new generation of deep learning architectures trained on massive, heterogeneous datasets to learn general temporal representations that can be transferred across forecasting tasks with minimal or no fine-tuning. Unlike models explicitly designed for a specific dataset or domain, FMs emphasize scalability, adaptability, and zero-shot generalization, enabling them to perform well even on unseen time series.

A representative example is CHRONOS (Ansari et al. 2024), which adapts techniques from language modeling to the time-series domain. To enable this transfer, continuous observations x_t are normalized and quantized into discrete tokens through a mapping $q(x_t) : R \to 1, 2, \ldots, B$, where B represents the number of bins. The resulting tokenized series $\mathbf{Z}t = z_1, \ldots, z_t$ is processed by a large language model that predicts the next token autoregressively. The model is trained by minimizing the negative log-likelihood over the token vocabulary $\mathcal{V}$ts, thus performing regression through classification. This approach enables CHRONOS to generate probabilistic forecasts, capturing multiple plausible future trajectories by sampling from its learned distributions.

From a complementary perspective, TimesFM (Time-Series Foundation Model) (Das et al. 2024) leverages a Transformer-based decoder directly trained on large-scale time-series data. Instead of tokenizing, TimesFM divides the input sequence x_t into patches, each representing contiguous temporal segments. These patches are masked during training to promote robust pattern inference and long-term forecasting. The model employs multi-head self-attention layers and residual connections to learn dependencies across patches while maintaining computational efficiency. During inference, TimesFM autoregressively predicts future patches based on the context, achieving zero-shot forecasting performance comparable to fully supervised models.

Overall, foundation models such as CHRONOS and TimesFM illustrate a paradigm shift from task-specific networks to general-purpose temporal learners capable of handling diverse environmental and forecasting scenarios, a direction that complements the statistical and deep learning methods previously discussed.

Graph Neural Network

Graph Neural Networks (GNNs) extend deep learning to relational and spatially structured data, enabling the modeling of complex dependencies beyond fixed time series. In our formulation, we consider a graph $G = (V, E)$ with node features $\mathbf{X} \in R^{d \times |V|}$, where each node $v \in V$ is associated with an embedding $\mathbf{h}_v^k$ updated through *Neural Message Passing* (NMP) (Hamilton 2020; Wu et al. 2021). At each iteration k, node representations are updated by aggregating information from their neighborhood $\mathcal{N}(v)$: $\mathbf{h}_v^k = \text{UPDATE}^{(k)}\Big(\mathbf{h}_v^{k-1}, \text{AGGREGATE}^{(k)}\big\{\mathbf{h}_u^{k-1} : u \in \mathcal{N}(v)\big\}\Big)$.

Here, the AGGREGATE function pools neighboring information and UPDATE integrates it into the node's current representation.

The following models are briefly described, emphasizing how each architecture extends or adapts the general message-passing formulation presented.

Graph Convolutional Network (GCN). The GCN (Kipf and Welling 2016) applies a symmetric-normalized convolution, sharing weights across nodes for scalability: $\mathbf{h}_v^k = \sigma\left(\mathbf{W}^k \sum_{u \in \mathcal{N}(v) \cup \{v\}} \frac{\mathbf{h}_u}{\sqrt{|\mathcal{N}(v)|\,|\mathcal{N}(u)|}}\right)$, where $\mathbf{W}^k$ is the trainable matrix at layer k and σ a nonlinear activation function, typically ReLU.

GraphSAGE. The GraphSAGE model (Hamilton, Ying, and Leskovec 2017) generalizes this process by learning aggregation functions (MEAN, SUM, or MAX) that com-

bine neighbor features with the node's own representation: $\mathbf{h}_v^k = \sigma\Big(\mathbf{W}^k \cdot \text{CONCAT}\Big(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k\Big)\Big)$.

Chebyshev Convolution (CHEB). The CHEB model (Defferrard, Bresson, and Vandergheynst 2016) defines graph filters as Chebyshev polynomials of the Laplacian $\mathbf{L}$, enabling efficient multi-hop information propagation without explicit eigen decomposition: $\mathbf{X} *_{\mathcal{G}} g = p_M(\mathbf{L})\mathbf{X}$.

Graph Attention Network (GAT). The GAT (Veličković et al. 2017) introduces attention coefficients $\alpha_{(v,u)}$ to weight the relevance of neighbors, yielding adaptive feature aggregation: $\mathbf{h}'_v = \sum_{u \in \mathcal{N}(v) \cup \{v\}} \alpha_{(v,u)}\, \mathbf{W}\mathbf{h}_u$.

Local Extremum Convolution (LEConv). The LEConv (Ranjan, Sanyal, and Talukdar 2020) enhances expressiveness by emphasizing local feature contrasts through a difference operator: $\mathbf{h}'_v = \mathbf{h}_v\mathbf{W}_1 + \sum_{u \in \mathcal{N}(v)} e_{(u,v)} \left(\mathbf{W}_2\mathbf{h}_v - \mathbf{W}_3\mathbf{h}_u\right)$, where $e_{(u,v)}$ denotes the edge weight between u and v, and $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$ are trainable parameter matrices.

Together, these architectures provide complementary perspectives on how spatial, structural, and relational information can be integrated into temporal modeling. Within the TAIAO dataset context, GNNs allow representing environmental variables as interconnected nodes, capturing dependencies among sensors, regions, or climate variables, thus extending traditional time-series modeling to a fully graph-based spatiotemporal learning framework.

Experimental Setup

Data Modeling

The dataset contains a total of 259,453 observations, which were partitioned into 80% (207,562 observations) for training and 20% (51,891 observations) for testing.

A preprocessing step was then carried out to handle missing values using interpolation. Linear interpolation was applied exclusively to the training set in order to prevent data leakage. Missing values at the edges of the series were filled using adjacent observations through forward and backward filling techniques. The interpolated series were then reinserted solely into the training partition.

For the test partition, any remaining missing values were replaced with the mean of the corresponding variable calculated from the training data. This procedure effectively prevents data leakage, since no information from the test set is used to estimate its own values. The same process was applied to all variables in the dataset.

After preprocessing, the data were further prepared to support multiple forecasting models: SARIMA, recurrent neural networks (RNNs), foundation models, and graph neural networks (GNNs), ensuring that the target variable, temperature, was properly modeled for each experimental setup.

The entire dataset was structured using a sliding-window strategy to capture temporal dependencies and enable multi-step forecasting. Each node or time series was represented as a sequence of observations sampled at regular intervals, preserving the temporal order of events. Two hyperparameters defined the temporal resolution of all experiments: the window size ($n_{\text{lags}} = 100$), representing the number of past observations used as input, and the forecasting horizon ($n_{\text{steps}} = 100$), corresponding to the number of future values predicted at each iteration.

To employ the GNNs, a graph was modeled in which each station represents a vertex, and an edge was established whenever a feasible connection existed in the terrestrial road network. In total, the resulting graph comprised 51 nodes and 1.177 edges, with distance used as the edge attribute.

The graph was constructed from road-network distances between stations, obtained through the Open Source Routing Machine (OSRM) service. Each node represents a station with its geographic coordinates, and an edge is created only when OSRM returns a valid route between two points. The edge weight corresponds to the route distance in kilometers, reflecting the actual connectivity of the road network.

The resulting adjacency matrix is symmetric and weighted, with infinite values indicating the absence of connections between node pairs. To optimize the process, a caching mechanism was implemented to avoid repeated OSRM queries. This procedure produces a simple and undirected graph, suitable for spatial analyses and modeling of dependencies between locations.

To maintain a consistent experimental scenario across training and testing, a subset was selected consisting of the last 1,000 observations from the training set and the subsequent 500 observations from the test set.

During the training phase, a grid search was performed for the SARIMA model using the training data. For the RNN and GNN models, a sliding-window approach was applied with a window shift of one observation, considering 100 past observations as input and a 100-step forecasting horizon. For testing, all models were provided with the first 100 observations of the test set as context, from which the next 400 observations were predicted.

In all models, the input feature consisted of past temperature values used to predict future values. For each time series, a separate model was trained for SARIMA, LSTM, GRU, and Foundation Models. For the GNNs, the input was a temporal graph, that is, a sequence of 100 graphs representing past time steps, used to predict the next 100 time steps (corresponding to 100 future graphs).

Two variations of the TimesFM model were evaluated: TimesFM 1.0 200m and TimesFM 2.0 500m. These versions differ in their parameter count (500M vs. 200M), number of transformer layers (50 vs. 20), and maximum context length (512 vs. 2048), respectively. In addition, TimesFM 2.0 supports longer temporal dependencies, enabling more accurate forecasts for extended horizons while maintaining efficiency across multiple data frequencies (Aksu et al. 2024).

All experiments were conducted on a machine with the following configuration: **CPU:** 2× Intel Xeon Gold 6326 @ 2.90GHz (32 physical cores, 64 threads total; **GPU:** 4× NVIDIA A16 (16GiB each); **RAM:** 125GiB; **Swap Memory:** 8GiB; and **Operating System:** Ubuntu 22.04 with kernel 5.15.0-144-generic. The libraries, code, and preprocessed dataset are available at https://github.com/LabIA-

UFBA/TAIAO-forecasting.

Metrics

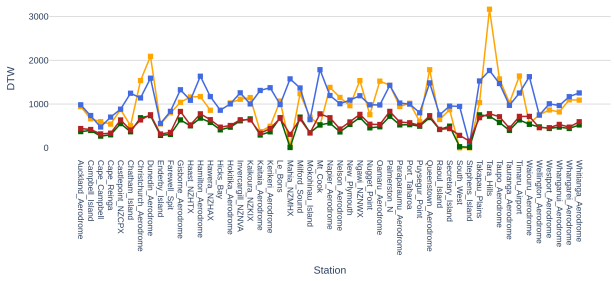
The first metric considered in our experiments was the mean squared error, MSE $= \frac{1}{n}\sum^{n}(y_i - \hat{y}_i)^2$, computing the differences between expected (y_i) and predicted ($\hat{y}_i$) values. We also used two related metrics: Root-Mean-Squared Error, RMSE $= \sqrt{\frac{1}{n}\sum^{n}(y_i - \hat{y}_i)^2}$, and Mean Absolute Error, MAE $= \frac{1}{n}\sum^{n}|y_i - \hat{y}_i|$. Aiming to perform different interpretations, we have analyzed our results using the mean absolute percentage error (MAPE) and the Dynamic Time Warping (DTW). MAPE $= \frac{1}{n}\sum^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$ expresses the prediction errors in terms of percentages of actual values. Traditional error metrics compute pairwise differences between predicted and ground truth values. Consequently, they often penalize models that strive to capture the true dynamics of the time series more than those that simply approximate a static mean. This problem is addressed using the DTW distance, which seeks the optimal alignment between two time series before computing their dissimilarity (Ding et al. 2008). DTW is particularly effective for comparing time series that may be out of phase or exhibit temporal distortions, as it allows non-linear warping along the time axis. The DTW distance between two time series, in our scenario h expected and predicted observations, Y_h and $\hat{Y}_h$, respectively, is recursively defined by Equation $DTW(Y_h, \hat{Y}_h) = \sqrt{dist(y_h - \hat{y}_h)}$, in which $dist(y_i, \hat{y}_j) = (y_i, \hat{y}_j)^2 + \min\left(dist(y_{i-1}, \hat{y}_j), dist(y_i, \hat{y}_{j-1}), dist(y_{i-1}, \hat{y}_{j-1})\right)$. The optimal alignment is represented by a sequence of matched index pairs known as the warping path.

Results

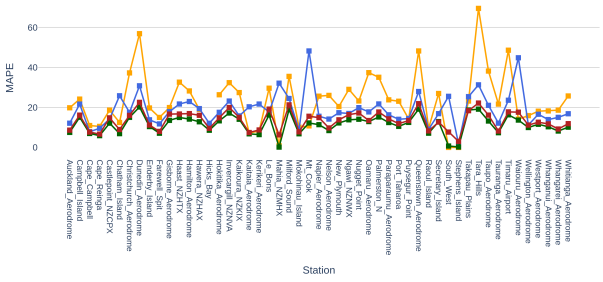
After fitting and training all models, and evaluating the foundation models in a zero-shot setting, we summarize predictive performance in Table 1, reporting metric-wise means and standard deviations over the 51 stations.

Based on these results, TimesFM achieved the lowest prediction errors for temperature over time. Beyond its strong performance on classical metrics (MAE, MSE, RMSE, and MAPE), the DTW results indicate that it closely follows the expected temporal pattern of the series.

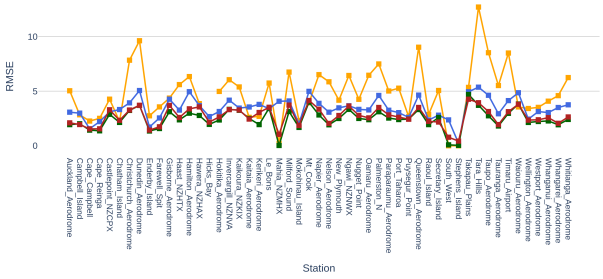
It is important to highlight the performance of the GNN models, especially CHEB, which achieved results statistically close to the foundation models. Recently, these large models have attracted attention for being pretrained on large corpora of time series, including climate data with similar dynamics. Recognizing that the graph structures used are an initial design with room for improvement, the results open avenues for further gains by incorporating additional features, refining preprocessing, and, most importantly, adopting meteorologically grounded strategies to define connections between nodes (stations). Regarding the statistical and RNN models, they delivered the poorest results, underscoring their limitations relative to the approaches discussed above. To better understand the results, we evaluate performance per station. Owing to space limits, we report a subset of metrics and only the top model from each learning family (Statistical, RNN, Foundation, GNN). We include RMSE, MAPE, and DTW, as they provide complementary views: scale-dependent error, scale-free interpretability, and alignment error, respectively. The best models in each family were SARIMA, GRU, TimesFM (500), and CHEB.



(a)



(b)



(c)

Figure 2: Per-station performance for the top best models from each class (SARIMA, GRU, TimesFM, CHEB) across three metrics (DTW, MAPE, and RMSE). Models: SARIMA GRU TIMES_500 GNN CHEB.

From these plots in Figure 2, it is apparent that the lower mean performance of SARIMA and GRU is not driven by specific stations but reflects a general pattern, as their curves remain markedly above those of TimesFM (500) and CHEB across all metrics and stations. By contrast, the plots emphasize the similarity between the foundation models and the GNNs. Focusing on the TimesFM and CHEB curves, we observe not only a clear convergence but also a shared overall pattern. In our final analysis (Figure 3), we selected time series from regions across both the North and South Islands of New Zealand, chosen for their environmental and population

Table 1: Performance comparison of models across evaluation metrics.

Approach	Model	DTW	MAE	MSE	RMSE	MAPE
Statistical	SARIMA	966.36 ± 550.11	3.83 ± 2.21	23.37 ± 13.71	29.15 ± 28.80	23.37 ± 13.73
RNN	LSTM	1122.15 ± 384.60	2.83 ± 0.97	13.04 ± 7.75	3.45 ± 1.07	19.35 ± 9.31
	GRU	1102.54 ± 339.99	2.79 ± 0.85	12.68 ± 6.65	3.41 ± 1.00	19.03 ± 8.38
Foundation Model	Chronos	661.14 ± 311.04	3.13 ± 1.16	19.48 ± 12.00	4.16 ± 1.48	18.71 ± 7.21
	TimesFM (200)	523.72 ± 191.89	2.03 ± 0.77	8.89 ± 6.76	2.76 ± 1.11	13.05 ± 5.51
	TimesFM (500)	**485.54 ± 178.56**	**1.80 ± 0.68**	**6.89 ± 4.39**	**2.45 ± 0.93**	**11.49 ± 4.69**
GNN	GCN	587.05 ± 154.45	2.30 ± 0.63	9.88 ± 5.17	3.01 ± 0.90	14.29 ± 4.23
	SAGE	563.39 ± 166.41	2.16 ± 0.62	8.83 ± 4.49	2.86 ± 0.80	14.14 ± 4.42
	CHEB	546.58 ± 164.21	2.03 ± 0.65	7.98 ± 4.48	2.69 ± 0.87	13.46 ± 4.69
	GAT	1085.75 ± 427.09	3.61 ± 1.16	19.18 ± 9.81	4.23 ± 1.13	24.21 ± 13.28
	LEConv	1447.62 ± 482.37	3.64 ± 1.20	19.84 ± 11.15	4.30 ± 1.16	26.22 ± 17.25

significance. To better assess the top models, we present the observed series alongside predictions from TimesFM and CHEB only.

The plots show that both TimesFM and CHEB capture cycles and seasonality well. Neighbourhood effects learned by the GNN contribute to improved temperature forecasts, narrowing the gap to the foundation models. However, anomalies remain hard to detect for both approaches, underscoring the need for better graph design, since incorporating meteorologically meaningful links could propagate anomaly signals from nearby stations, reveal causal structure, and reduce these errors overall.

Final Remarks

This study presented a comprehensive evaluation of statistical, recurrent, foundation, and graph-based models for temperature forecasting across New Zealand. Among all tested approaches, the foundation model TimesFM (500) achieved the lowest overall errors, while the CHEB graph neural network reached statistically comparable results, demonstrating the effectiveness of spatial reasoning in capturing inter-station dependencies. The analysis of per-station and per-region forecasts further revealed consistent patterns across climatic zones, confirming that both models can generalize well to heterogeneous environmental conditions.

These findings highlight that foundation models and graph-based architectures are complementary: while the former leverage large-scale pretraining to learn broad climatic representations, the latter incorporate explicit spatial structure that enhances local interpretability and robustness. Integrating these paradigms offers a promising direction for future research, combining the adaptability of foundation models with the relational expressiveness of graph learning.

Ultimately, it is worth emphasizing that the GNN results substantiate neighborhood influences in modeling weather conditions. In this initial study, graph edges were defined by inter-station distance. To better leverage GNNs, additional meteorologically meaningful features should be incorporated. For example, rather than using wind direction and speed as node attributes, they could weight the edges to represent advective flow. Our results suggest that this modification, together with the inclusion of other meteorologically grounded information, is likely to yield improved predictive performance.

Acknowledgements

This work was supported by CNPq (Brazilian National Council for Scientific and Technological Development) grants [404771/2024-6, 406354/2023-5, 312755/2023-6, 313053/2023-5], UFBA/CNPq 68/2022 - MAI/DAI, Maria Emilia Foundation grant to 01/2023, INCITE FAPESB (Bahia Research Foundation) grant TO PIE0002/2022, CAPES (Coordination for the Improvement of Higher Education Personnel – Brazil), and FAPESB grant [1589/2021].

References

Aksu, T.; Woo, G.; Liu, J.; Liu, X.; Liu, C.; Savarese, S.; Xiong, C.; and Sahoo, D. 2024. GIFT-Eval: A Benchmark For General Time Series Forecasting Model Evaluation. *arxiv preprint arxiv:2410.10393*.

Ansari, A. F.; Stella, L.; Turkmen, A. C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Arango, S. P.; Kapoor, S.; Zschiegner, J.; Maddix, D. C.; Wang, H.; Mahoney, M. W.; Torkkola, K.; Wilson, A. G.; Bohlke-Schneider, M.; and Wang, B. 2024. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research*.

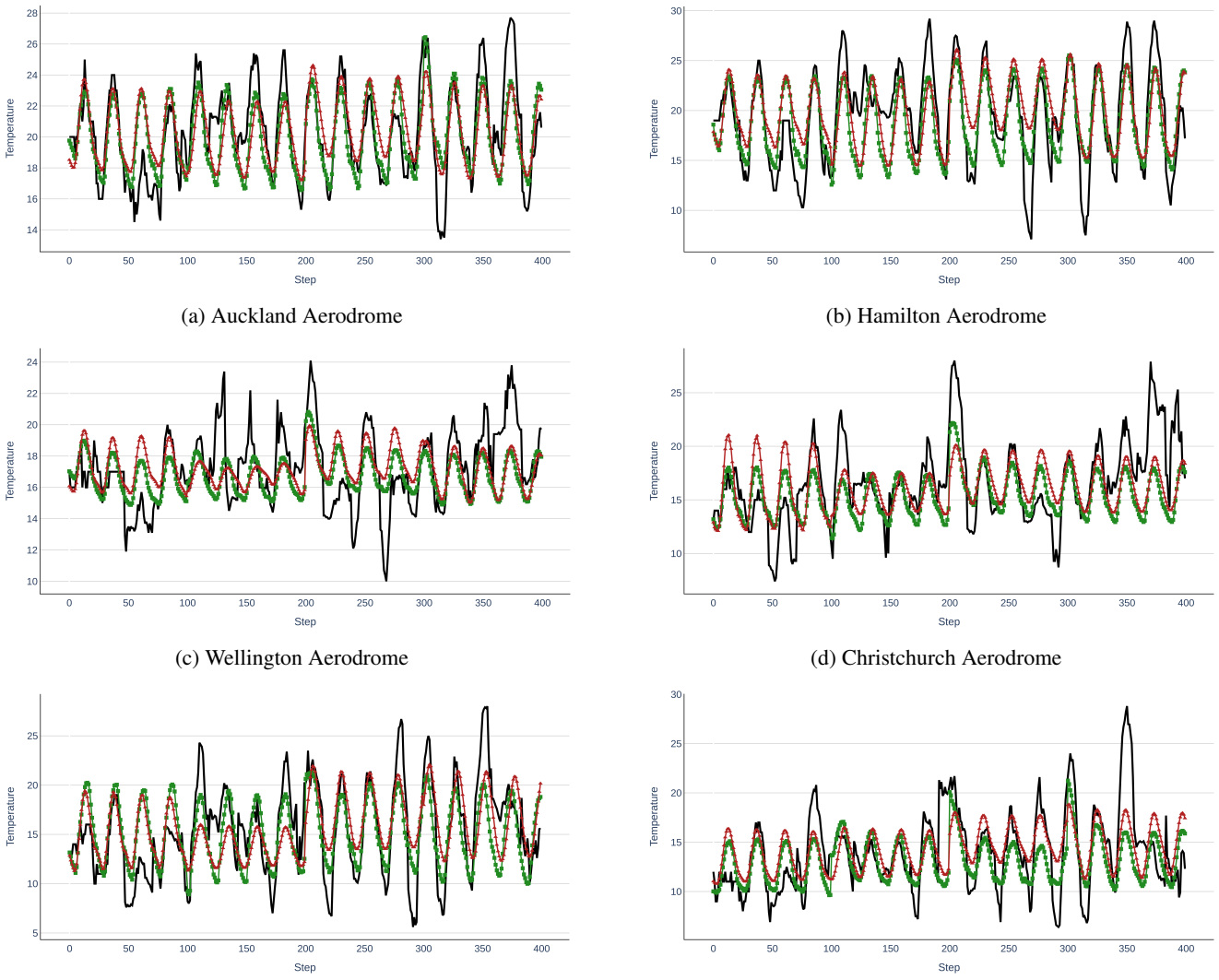
Box, G. E.; Jenkins, G. M.; Reinsel, G. C.; and Ljung, G. M. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Change, I. P. O. C. 2023. Buildings. In *Climate Change 2022-Mitigation of Climate Change*, 953–1048. Cambridge University Press.

Cho, K.; Van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2024. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized



(a) Auckland Aerodrome

(b) Hamilton Aerodrome

(c) Wellington Aerodrome

(d) Christchurch Aerodrome

(e) Queenstown Aerodrome

(f) Invercargill

Figure 3: Forecast visualization for the top two models (TimesFM and CHEB) against ground truth at six selected stations. **Model** — Real — Times_500 — GNN_CHEB.

spectral filtering. *Advances in neural information processing systems (NeurIPS)*, 29.

Ding, H.; Trajcevski, G.; Scheuermann, P.; Wang, X.; and Keogh, E. 2008. Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. *Proc. VLDB Endow.*, 1(2): 1542–1552.

Fan, J.; Bai, J.; Li, Z.; Ortiz-Bobea, A.; and Gomes, C. P. 2022. A GNN-RNN approach for harnessing geospatial and temporal information: application to crop yield prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 11873–11881.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems (NeurIPS)*, 30.

Hamilton, W. L. 2020. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3): 1.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Nguyen, T.; Brandstetter, J.; Kapoor, A.; Gupta, J. K.; and Grover, A. 2023. ClimaX: a25 foundation model for weather and climate. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23.

Oskarsson, J.; Landelius, T.; Deisenroth, M. P.; and Lindsten, F. 2024. Probabilistic Weather Forecasting with Hierarchical Graph Neural Networks. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; and Zhang, C., eds., *Advances in Neural Information Process-*

ing Systems, volume 37, 41577–41648. Curran Associates, Inc.

Ranjan, E.; Sanyal, S.; and Talukdar, P. 2020. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 5470–5477.

Shumway, R. H.; and Stoffer, D. S. 2011. *Time series analysis and its applications*, volume 3. Springer.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Werbos, P. J. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10): 1550–1560.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4.

Yalavarthi, V. K.; Madhusudhanan, K.; Scholz, R.; Ahmed, N.; Burchert, J.; Jawed, S.; Born, S.; and Schmidt-Thieme, L. 2024. Grafiti: Graphs for forecasting irregularly sampled time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16255–16263.

Zhao, X.; Zhou, Z.; zhangwenlong; Liu, Y.; Chen, X.; Gong, J.; Chen, H.; Fei, B.; Chen, S.; Ouyang, W.; Wu, X.-M.; and BAI, L. 2025. WeatherGFM: Learning a Weather Generalist Foundation Model via In-context Learning. In *The Thirteenth International Conference on Learning Representations*.

Reproducibility Checklist

Instructions for Authors:
This document outlines key aspects for assessing reproducibility. Please provide your input by editing this `.tex` file directly. For each question (that applies), replace the "Type your response here" text with your answer.

Example: If a question appears as

```
\question{Proofs of all novel claims
are included} {(yes/partial/no)}
Type your response here
```

you would change it to:

```
\question{Proofs of all novel claims
are included} {(yes/partial/no)}
yes
```

Please make sure to:

- Replace ONLY the "Type your response here" text and nothing else.
- Use one of the options listed for that question (e.g., **yes**, **no**, **partial**, or **NA**).
- **Not** modify any other part of the `\question` command or any other lines in this document.

You can `\input` this .tex file right before `\end{document}` of your main file or compile it as a stand-alone document. Check the instructions on your conference's website to see if you will be asked to provide this checklist with your paper or separately.

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) NA

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) no

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) Type your response here

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) Type your response here

2.4. Proofs of all novel claims are included (yes/partial/no) Type your response here

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) Type your response here

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) Type your response here

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) Type your response here

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) Type your response here

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) NA

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) NA

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) yes

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) yes

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) yes

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) yes

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) yes

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) yes

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) no

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes