Enhancing Energy Demand Predictions with Seasonal Patterns and Dual Temporal Inputs in a Hybrid GPI-DSSM Model

Rishi Malyala*,
School of Electrical Engineering and Telecommunications
University of New South Wales, Sydney
Sydney, Australia
ORCID*: https://orcid.org/0009-0002-3502-8487

Kuthsav Thattai,
School of Electrical Engineering and Telecommunications
University of New South Wales, Sydney
Sydney, Australia
kuthsav.thattai@unsw.edu.au

Anam Malik,
School of Electrical Engineering and Telecommunications
University of New South Wales, Sydney
Sydney, Australia
anam.malik1@unsw.edu.au

Jayashri Ravishankar,
School of Electrical Engineering and Telecommunications
University of New South Wales, Sydney
Sydney, Australia
jayashri.ravishankar@unsw.edu.au

***Abstract*—This paper introduces a novel approach to energy demand forecasting by integrating dual-temporal analysis with weather-based features and annual lag sequences. Building upon the foundational Graph Patch Informer-Deep Sequential State Memory (GPI-DSSM), this framework evolves into hybrid Transformer-Gated Recurrent Unit (GRU) model that combines recent historical data with sequences from the same period in the prior year, to effectively capture both short-term trends and recurring seasonal patterns. Weather data, including temperature and humidity, is dynamically incorporated to adjust weights, enhancing forecast accuracy during high-impact periods. The model demonstrates high precision in predicting biweekly demand, balancing immediate fluctuations with long-term cyclic behavior. Extensive experiments on high-resolution datasets evaluate the impact of varying input sequence lengths, highlighting significant accuracy improvements as the sequence length increases. Leveraging multi-GPU optimization, the model achieves scalability and efficiency in processing large-scale data. Results indicate substantial reduction in forecasting errors compared to conventional methods, underscoring the efficacy of integrating dual-temporal and weather-driven features.**

Keywords— Hybrid GPI-DSSM Model, Seasonal Temporal Pattern Inputs, Dual Temporal Inputs, Graph Patch Informer, Deep Sequential State Memory, Weather-Driven Predictions

I. Introduction

Forecasting energy demand with high accuracy is crucial for ensuring reliability and operational efficiency in electricity markets. Accurate demand predictions enable optimal resource allocation, informed bidding strategies, and effective grid management. Traditional models often rely on weather data as key input variables to predict energy demand fluctuations. While this approach provides insights into local energy dynamics, it introduces spatial-temporal noise and computational complexity, particularly for large-scale systems spanning diverse climatic regions [1], [2].

In a previous study, a novel hybrid Graph Patch Informer-Deep Sequential State Memory (GPI-DSSM) model was developed to forecast energy demand independently of weather inputs, focusing solely on aggregate demand and market price data [3]. This approach achieved remarkable predictive accuracy by leveraging the intrinsic relationships between market variables, eliminating the uncertainties associated with weather data. However, as renewable energy integration expands and climate-driven policies gain traction, incorporating weather-driven features into forecasting models become very important for capturing renewable variability and demand dynamics.

This study extends the foundational principles of the GPI-DSSM model by integrating weather variables such as temperature and humidity alongside demand data. The electricity demand data used in this study is sourced from the Ausgrid Distribution Zone Substation dataset for the Sydney CBD, which provides granular energy consumption data across a 14-year timespan (2009–2023) [4]. Weather variables are obtained from Open-Meteo's Historical Weather API, which supplies high-resolution weather data, including temperature, relative humidity, and other meteorological parameters, for the same period and location [5]. The combination of these datasets provides a comprehensive view of energy demand and its weather dependencies in an urban setting. The enhanced hybrid model employs a Transformer-based GPI for capturing long-term temporal dependencies and DSSM architecture, utilizing GRU layers to model short-term variations in demand and weather interactions. By incorporating annual seasonality patterns into the model, this research aims to improve the accuracy of mid- to long-term demand forecasts while maintaining robustness against noisy weather fluctuations.

Compared to recent advancements in multivariate energy forecasting [6], spatio-temporal graph-based methods [7], and self-supervised transformer variants [8], this work

demonstrates the synergistic benefits of combining temporal, seasonal, and weather data in a hybrid architecture. Key contributions of this paper include the introduction of dual temporal inputs to capture both short-term and seasonal dependencies and the evaluation of weather features to dynamically adjust predictive weights, as proposed in previous works. The efficiency of this comprehensive approach is in both short-term and biweekly forecasting ranges while maintaining robustness to weather-induced demand variability. Section II discusses the dataset and preprocessing steps, Section III elaborates on the model architecture, and Section IV presents the performance analysis, highlighting the advantages of integrating weather data and seasonality features. This study positions the weather-enhanced GPI-DSSM model as a significant advancement in energy demand forecasting, addressing the limitations of both purely weather-independent and weather-reliant approaches.

II. Data Collection and Preprocessing

The accurate forecasting of energy demand requires careful selection of variables that capture both immediate fluctuations and long-term patterns. This study builds on the foundational methods established in previous work [3] and integrates insights from mathematical principles of time-series analysis and meteorological factor modelling [9]. High resolution electricity demand data and weather variables are used as data inputs to the model—specifically, temperature (T_{2m}) and relative humidity (RH_{2m}) —for City Central, spanning 14 years (2009–2023) at 15-minute intervals [4], [5]. The inclusion of these variables is informed by their significant role in driving energy consumption trends, particularly through heating, cooling, and ventilation systems [7].

Electricity demand serves as the primary target variable, while temperature and relative humidity act as weather-driven features. The inclusion of these variables complements findings in previous research [6], [10], which highlight the impact of weather on energy consumption patterns. As illustrated in Fig. 1, temperature demonstrates a dynamic inverse relationship with demand, varying across seasons and times of day. For instance:

- $\rho_{\text{Summer Nights}} = -0.49$ (cooling demand decreases as temperatures fall)
- $\rho_{\text{Winter Mornings}} = -0.65$ (heating demand increases as temperatures fall)

Relative humidity (RH_{2m} , while exhibiting weaker correlations compared to temperature, complements the latter by capturing secondary weather effects, such as adjustments in Heating, Ventilation and air-conditioning (HVAC) operation under varying humidity levels. Together, these variables form a comprehensive representation of weather-driven demand patterns, justifying their inclusion in this study. The distribution of energy demand, temperature, and humidity, as illustrated in Fig. 2, highlights the inherent variability in these features. The significant range of energy demand values compared to temperature and humidity outlines the need for preprocessing steps, such as normalization, to ensure balanced feature distribution during model training.

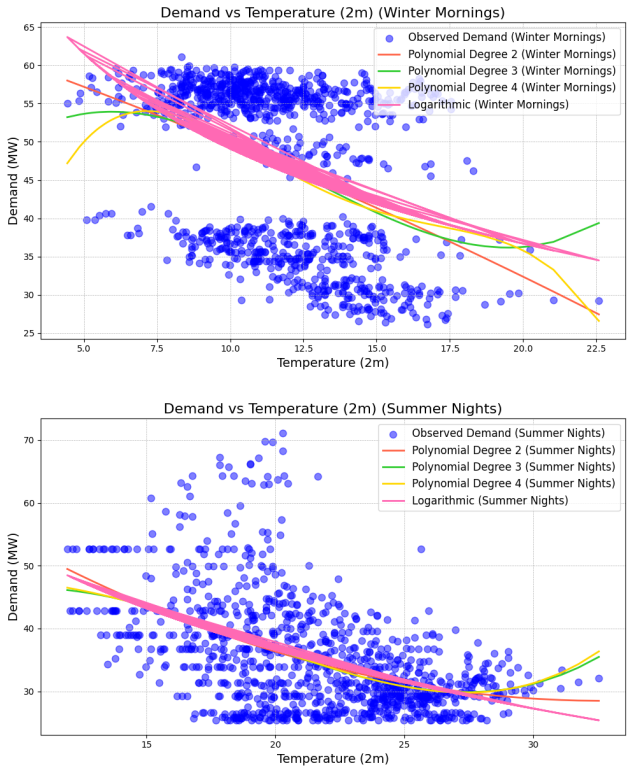

Fig. 1. Demand vs Temperature (2m above sea-level) Trend Analysis with Seasonal and Day Variations

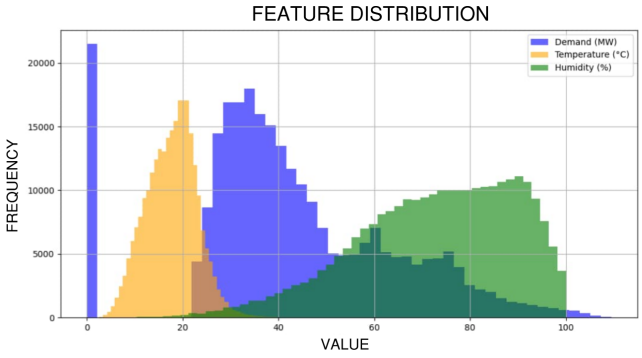

Fig. 2. Feature Distribution of Demand, Temperature and Humidity Variables across the input dataset

A. Preprocesing

To ensure temporal consistency and to prepare the data for model training, a systematic preprocessing approach is applied. Demand and weather variables are normalized using the Min-Max Normalization technique, which standardizes the inputs to the range [0, 1], and is shown in (1):

$$x_{scaled} = \frac{(x_t - x_{min})}{(x_{max} - x_{min})} \quad (1)$$

where x_t represents the raw value, and x_{min} and x_{max} are the variable's minimum and maximum observed values, respectively. This transformation prevents variables with

larger magnitudes from dominating the learning process while preserving their relative importance. Fig. 3 illustrates the normalized energy demand, temperature, and humidity over a representative 42-day period. The consistent range of these features after Min-Max scaling ensures their equitable contribution during model training while preserving critical temporal trends.

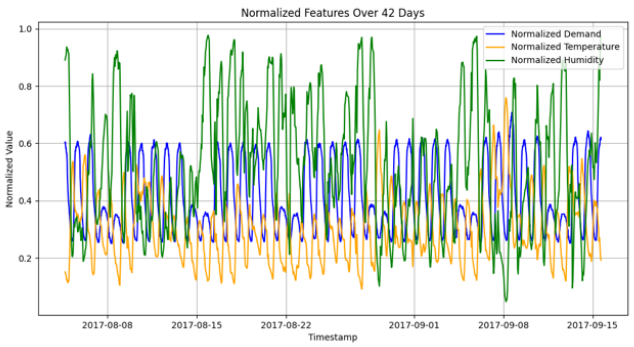

Fig. 3. Visualization of Normalized Features for Demand, Temperature and Humidity variables shown over 42 day period.

Gaps in weather data, which is a common issue in long-term datasets [8], are addressed using linear interpolation, ensuring temporal continuity across the dataset. Additionally, the weather data is resampled to align with the 15-minute resolution of the demand data, ensuring consistent temporal synchronization across all features. This step addresses the spatial-temporal variability highlighted in earlier works on meteorological factors in time-series analysis [9]. Outlier treatment is not separately performed beyond normalization and interpolation, as energy demand data is assumed to be reliable based on source quality.

B. Sequence Design

The sequence design in this study employs a dual temporal input structure to model both short-term fluctuations and long-term seasonal dependencies. The input sequence, denoted as $\boldsymbol{X}_{input}$, combines recent historical data $\boldsymbol{X}_t$ with annual lag data $\boldsymbol{X}_{t-1y}$, enabling the model to leverage temporal and seasonal patterns simultaneously, building further on the framework from previous study [3]. Mathematically, this is expressed as shown in (2).

$$\boldsymbol{X}_{input} = [\boldsymbol{X}_t, \quad \boldsymbol{X}_{t-1y}] \tag{2}$$

where $\boldsymbol{X}_t, \boldsymbol{X}_{t-1y} \in \boldsymbol{R}^{n\times d}$, n is the number of timesteps, and d represents the number of features.

1) Recent Historical Data

The recent historical data matrix, $\boldsymbol{X}_t$, spans n = 4032 timesteps (42 days at 15-minute intervals) and includes d = 3 features, which include energy demand (MW), temperature ($T_{2\mathrm{m}}$), and relative humidity ($RH_{2\mathrm{m}}$). The matrix is structured as shown in (3).

$$\boldsymbol{X}_t = \begin{bmatrix} \boldsymbol{x}_{1,1} & \boldsymbol{x}_{1,2} & \cdots & \boldsymbol{x}_{1,d} \\ \boldsymbol{x}_{2,1} & \boldsymbol{x}_{2,2} & \cdots & \boldsymbol{x}_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_{n,1} & \boldsymbol{x}_{n,2} & \vdots & \boldsymbol{x}_{n,d} \end{bmatrix} \tag{3}$$

where each row $\boldsymbol{x}_{i,:}$ contains values for the features at timestep $\boldsymbol{x}$, ensuring a granular representation of recent fluctuations.

2) Annual Lag Data

The annual lag data, $\boldsymbol{X}_{t-1y}$, spans n = 35040 timesteps (1 year at 15-minute intervals) and captures recurring seasonal patterns. The structure of $\boldsymbol{X}_{t-1y}$ mirrors $\boldsymbol{X}_t$ as shown in (4).

$$\boldsymbol{X}_t = \begin{bmatrix} \boldsymbol{x}_{1,1}{}^{lag} & \boldsymbol{x}_{1,2}{}^{lag} & \cdots & \boldsymbol{x}_{1,d}{}^{lag} \\ \boldsymbol{x}_{2,1}{}^{lag} & \boldsymbol{x}_{2,2}{}^{lag} & \cdots & \boldsymbol{x}_{2,d}{}^{lag} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_{n,1}{}^{lag} & \boldsymbol{x}_{n,2}{}^{lag} & \vdots & \boldsymbol{x}_{n,d}{}^{lag} \end{bmatrix} \tag{4}$$

where $\boldsymbol{x}_{i,j}{}^{lag}$ represents the lagged feature values from the same period in the previous year. This component explicitly models cyclical demand variations, such as increased cooling demand in summer or heating demand in winter.

3) Output Representation

The target sequence $\boldsymbol{Y}_t$ predicts energy demand over $\boldsymbol{m} = 1344$ timesteps (2 weeks, 15-minute intervals) as shown in (5).

$$\boldsymbol{Y}_t = \begin{bmatrix} \boldsymbol{y}_{t+1} \\ \boldsymbol{y}_{t+2} \\ \vdots \\ \boldsymbol{y}_{t+m} \end{bmatrix}, \ \boldsymbol{Y}_{t+i} = \boldsymbol{D}_{t+i}, \ \boldsymbol{i} \in \{\boldsymbol{1}, \quad \ldots, \quad \boldsymbol{m}\} \tag{5}$$

With the combined output input matrix, $\boldsymbol{Z} \in \boldsymbol{R}^{m\times 2d}$ is the output weight matrix, $\boldsymbol{h} \in \boldsymbol{R}^{2d}$ is the encoded feature representation of $\boldsymbol{Z}$, and $\boldsymbol{D}$ is the energy demand decomposition, which is represented as shown in (6).

$$\boldsymbol{D}_t = f(\boldsymbol{W}_t, \boldsymbol{S}_t, \boldsymbol{\varepsilon}_t) \tag{6}$$

where $\boldsymbol{W}_t$ is the short-term variations from $\boldsymbol{X}_t$, $\boldsymbol{S}_t = \boldsymbol{S}_{t-1}$ is the seasonal patterns from $\boldsymbol{X}_{t-1y}$, and $\boldsymbol{\varepsilon}_t$ is the residual noise. The combined input sequence $\boldsymbol{Z}_t$, integrates these components as shown in (7).

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{x}_{1,1} & \cdots & \boldsymbol{x}_{1,d} & \boldsymbol{x}_{1,1}{}^{lag} & \cdots & \boldsymbol{x}_{1,d}{}^{lag} \\ \boldsymbol{x}_{2,1} & \cdots & \boldsymbol{x}_{2,d} & \boldsymbol{x}_{2,1}{}^{lag} & \cdots & \boldsymbol{x}_{2,d}{}^{lag} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_{n,1} & \cdots & \boldsymbol{x}_{n,d} & \boldsymbol{x}_{n,1}{}^{lag} & \cdots & \boldsymbol{x}_{n,d}{}^{lag} \end{bmatrix}, \boldsymbol{Z} \in \boldsymbol{R}^{n\times 2d} \tag{7}$$

This methodology processes combined input sequence $\boldsymbol{Z}_t$ to minimize residual variability $\boldsymbol{\varepsilon}_t$, improving prediction accuracy. The sequence assembly is therefore, represented as the concatenated output shown in (8).

$$\boldsymbol{Z} = [\boldsymbol{X}_t, \quad \boldsymbol{X}_{t-1y}], \ \boldsymbol{Z} \in \boldsymbol{R}^{n\times 2d} \tag{8}$$

III. Methodology

The proposed methodology integrates a multi-layered architecture that leverages the strengths of temporal encoding and sequential memory mechanisms. This hybrid model combines Transformer-based encoders and GRU-based decoders to address the challenges of forecasting energy

demand influenced by weather and seasonal patterns. GRUs, also compared to other models, offer faster training, fewer parameters, and better performance for mid-length sequence modelling compared to other models, making them particularly suitable for the dual temporal input structure.

A. *Transformer-Based Encoder*

The Transformer encoder processes the input sequence $\boldsymbol{Z}$ to learn temporal and cross-feature dependencies. Each encoder layer applies the multi-head self-attention mechanism and a feedforward network.

1) *Multi-Head Self-Attention*

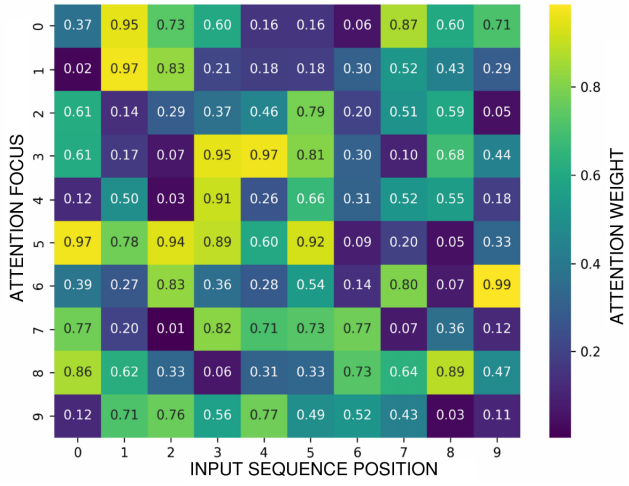

Fig. 4. Attention Positional Heatmap for Demand and Weather Input Sequence Positions

The attention score between two timesteps $\boldsymbol{i}$ and $\boldsymbol{j}$ is computed as shown in (9) and (10), where Query is represented as $\boldsymbol{Q} = \boldsymbol{ZW_Q}$, Key as $\boldsymbol{K} = \boldsymbol{ZW_k}$, Value as $\boldsymbol{V} = \boldsymbol{ZW_V}$ and $\boldsymbol{d_k}$ is the dimensionality of the attention space.

$$\text{Score}\ (\boldsymbol{i},\boldsymbol{j}) = \frac{\boldsymbol{Q_i K_j}^{\mathbf{T}}}{\sqrt{\boldsymbol{d_k}}} \tag{9}$$

$$\boldsymbol{W_Q}, \boldsymbol{W_K}, \boldsymbol{W_V} \in \boldsymbol{R^{d \times d_k}} \tag{10}$$

The attention weights $\boldsymbol{\alpha_{ij}}$ are computed by applying the softmax function to normalize features as shown in (11).

$$\boldsymbol{\alpha_{ij}} = \frac{\exp\ (\textbf{Score}\ (\boldsymbol{i},\boldsymbol{j}))}{\sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \exp\ (\textbf{Score}\ (\boldsymbol{i},\boldsymbol{j}))} \tag{11}$$

The Output for each timestep is then computed as shown in (12).

$$\mathbf{O}_{\boldsymbol{i}} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{\alpha_{ij}} \mathbf{V}_{\boldsymbol{j}} \tag{12}$$

Fig. 4 illustrates the attention weights for input features (demand, temperature, and humidity) and timesteps. It is a representative example using a selected input segment for interpretability and is not intended to generalize across the entire dataset. For instance, a high attention weight $\boldsymbol{\alpha_{1,1}} = 0.97$ (from Query Attention Position 1-1) in the heatmap indicates that demand at the second timestep strongly correlates with temperature at the first timestep. This shows the importance of earlier temperature patterns in influencing future demand. A moderate attention weight $\boldsymbol{\alpha_{3,0}} = 0.61$ (From Query Position 2 to Key Position 1) highlights intra-feature dependencies, such as humidity referencing itself across consecutive timesteps. These attention weights dynamically adjust based on the input features and timesteps, allowing the model to prioritize relationships critical for energy demand forecasting. This mechanism ensures that both short-term and long-term dependencies are effectively captured, as visualized in Fig. 4.

2) *Feed-Forward Network*

Each timestep representation is further refined using a feedforward network as shown in (13),

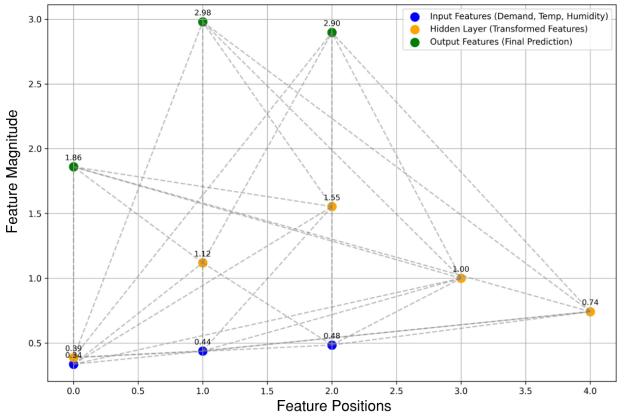

Fig. 5. Hidden representation of feedforward network transformation between Demand and Weather Variables

$$\boldsymbol{H} = \text{Relu}(\boldsymbol{OW_1} + \boldsymbol{b_1})\boldsymbol{W_2} + \boldsymbol{b_2} \tag{13}$$

where $\boldsymbol{W_1}, \boldsymbol{W_2} \in \boldsymbol{R^{d \times d_f}}$ and $\boldsymbol{b_1}, \boldsymbol{b_2} \in \boldsymbol{R^{d_f}}$ are learnable parameters. The feedforward network transforms demand and weather features into a hidden representation, as shown in Fig. 5.

The figure visually captures the flow of input features (demand, temperature, and humidity) into hidden layers and their final transformation into output predictions, emphasizing the mapping across layers. The encoded sequence $\boldsymbol{H} \in \boldsymbol{R^{n \times d}}$ is passed to the GRU based decoder for final processing.

B. *GRU-Based Decoder*

The GRU-based decoder processes the encoded sequence $\boldsymbol{H}$ to predict energy demand by dynamically balancing information and new input features, complementing work previously done in hybrid DSSM models [3].

The decoder achieves this through its internal mechanisms, such as reset gate $\mathbf{r_t}$, update gate $\boldsymbol{z_t}$, candidate hidden gate $\boldsymbol{\tilde{h}_t}$, and the final hidden state $\boldsymbol{h_{t-1}}$. At each timestep $\boldsymbol{t}$, the reset gate $\mathbf{r_t}$ controls how much of the previous hidden state $\mathbf{h_{t-1}}$ is used as shown in (14), where $\boldsymbol{W_r}, \boldsymbol{U_r}, \boldsymbol{b_r}$ are learnable parameters.

$$\boldsymbol{r_t} = \sigma(\boldsymbol{W_r H_t} + \boldsymbol{U_r h_{t-1}} + \boldsymbol{b_r}) \tag{14}$$

The update gate $\boldsymbol{z_t}$ determines the trade-off between retaining past information and using new input as shown in (15).

$$\boldsymbol{z_t} = \sigma(\boldsymbol{W_z H_t} + \boldsymbol{U_z h_{t-1}} + \boldsymbol{b_z}) \quad (15)$$

The candidate hidden state $\widetilde{\boldsymbol{h}}_t$ and final hidden state $\boldsymbol{h_{t-1}}$ are computed as shown in (16) and (17).

$$\widetilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W_h H_t} + \boldsymbol{U_h}(\boldsymbol{r_t} \odot \boldsymbol{h_{t-1}}) + \boldsymbol{b_h} \quad (16)$$

$$\boldsymbol{h_t} = (\boldsymbol{1} - \boldsymbol{z_t}) \odot \boldsymbol{h_{t-1}} + \boldsymbol{z_t} \odot \widetilde{\boldsymbol{h}}_t \quad (17)$$

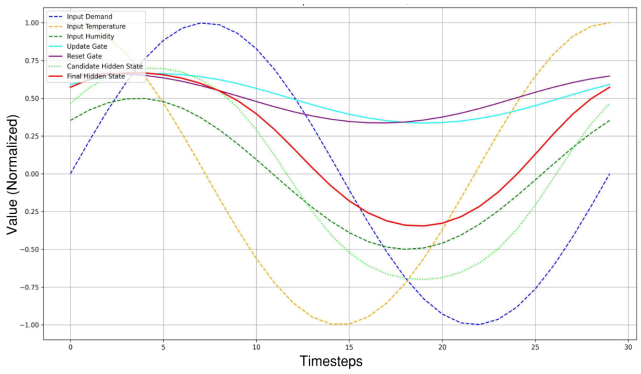

Fig. 6. Visual Representation of Synthetic interplay between Demand, Temperature and Humidity with Transformations showing Hidden States in the GRU Encoder

Fig. 6 visualizes the synthetic interplay (not based on actual data timesteps) between the input features (demand, temperature, humidity) and the GRU gates (reset, update) over 30 timesteps. This alignment of gates with input features demonstrates the GRU's adaptability, effectively filtering historical patterns and incorporating critical weather-driven variations for accurate demand forecasting.

C. Training Loss

The model uses Huber Loss as the training function, which effectively balances sensitivity to outliers and ensures smooth gradient updates. The Huber Loss employed is mathematically represented as shown in (18).

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \begin{cases} 0.5(\boldsymbol{y} - \hat{\boldsymbol{y}})^2 & if\ |\boldsymbol{y} - \hat{\boldsymbol{y}}| \le \delta \\ \delta(|\boldsymbol{y} - \hat{\boldsymbol{y}}| - 0.5\delta) & otherwise \end{cases} \quad (18)$$

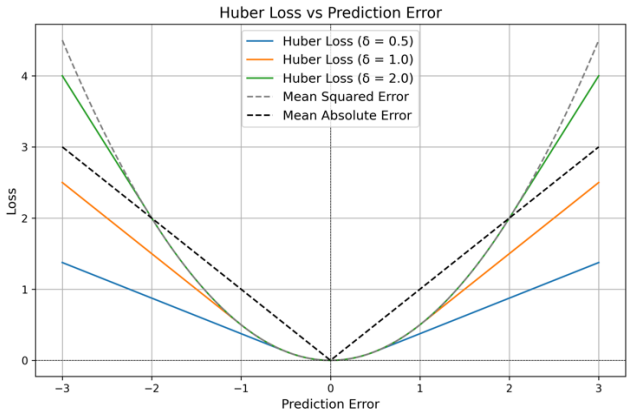

Fig. 7. Synthetic Training Loss representation of Hubber loss and Adam Optimizer compared to MAE and MSE Loss

where $\boldsymbol{y}$ is the true demand value, $\hat{\boldsymbol{y}}$ is the predicted demand value, and δ is a threshold value, which defaults to 1 in this implementation.

Fig. 7 illustrates the relationship between prediction error and loss for the Huber Loss at varying $\boldsymbol{\delta}$ values, as well as its comparison with Mean Squared Error and Mean Absolute Error. The plot highlights how smaller $\boldsymbol{\delta}$ values increase the transition sensitivity between MSE and MAE, making the loss function more adaptive for different error scales. This adaptability ensures the model can effectively optimize under diverse error distributions encountered in the dataset.

IV. Effectiveness of the Model and Analysis of Results

The GPI-DSSM model's performance is evaluated under two configurations: with and without seasonality, incorporating annual lag features captures recurring patterns, improving forecasting accuracy, as highlighted in prior works on seasonal demand forecasting[2], [6], [10]. The input sequence $\boldsymbol{Z}_t$ is defined as shown in (19).

$$\boldsymbol{Z}_t = \begin{cases} [\boldsymbol{X}_t\,], & \text{if seasonality is excluded} \\ [\boldsymbol{X}_t\,,\ \boldsymbol{X}_{t-1}\,], & \text{if seasonality is included} \end{cases} \quad (19)$$

Fig. 8 illustrates the performance comparison, where the model achieves a Mean Absolute Error (MAE) of 4.90 with seasonality, compared to 5.06 without. The incorporation of annual lag features enables the model to capture long-term periodic trends, improving forecasting accuracy.

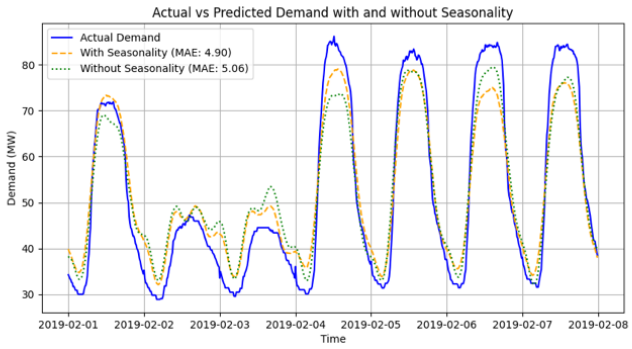

Fig. 8. Output Analysis of Demand With and Without Seaonality Input Sequences

In addition, Fig. 9 analyzes the impact of varying input sequence lengths on model performance. Sequence lengths range from 96 to 4032 timesteps, corresponding to approximately 24 hours to 10 days of historical data. The MAE decreases significantly with longer input sequences, improving from 5.32 (96 timesteps) to 3.78 (672 timesteps). Beyond 672 timesteps, the MAE stabilizes around 3.95, indicating diminishing returns. This trend highlights the balance between leveraging sufficient historical context and avoiding redundant or noisy data.

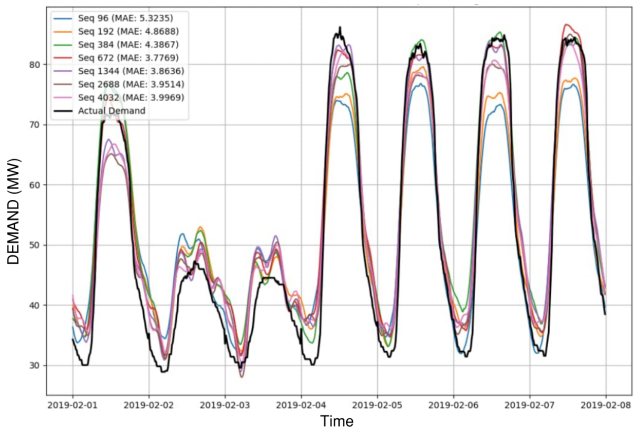

Fig. 9. Output Analysis for Increasing Sequence Length of Demand and Weather Input Sequences

Fig. 10 demonstrates the training loss convergence over 20 epochs. The loss function's rapid decline highlights the efficiency of the Huber Loss and the Adam optimizer in achieving convergence. The smooth and stable reduction of the Mean Squared Error (MSE) emphasizes the model's ability to handle complex patterns in the demand and weather data without overfitting. Together with the MAE results, the loss convergence underscores the robustness and efficiency of the GPI-DSSM architecture.

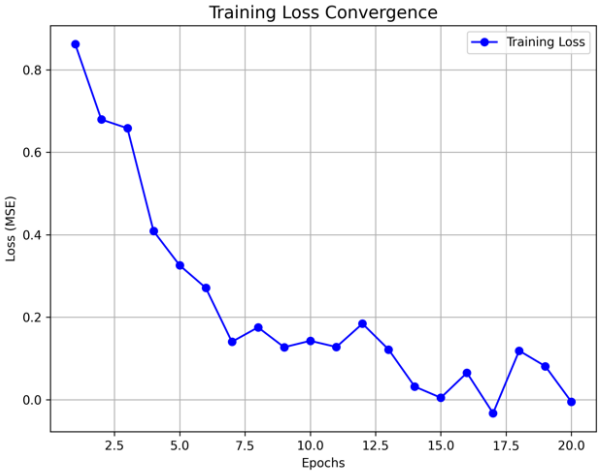

Fig. 10. Training Loss MSE Convergence with Increasing Training Epochs

Compared to state-of-the-art models, such as the Transformer-GRU hybrid [11] and spatio-temporal graph neural networks [7], the GPI-DSSM model achieves superior performance in both short-term and biweekly forecasting horizons. It should be noted that the previous studies [11] [7] reported general findings, whereas the work in this paper is based on direct benchmarking performance with the same dataset. These improvements highlights the benefits of combining seasonal patterns with weather features. The results underscore the importance of model scalability and dynamic feature integration.

V. Conclusion

This study presents an efficient GPI-DSSM hybrid model that utilizes dual temporal inputs to capture both short-term fluctuations and long-term seasonal patterns in energy demand. By combining weather-driven features with advanced transformer and GRU-based architectures, the model demonstrates rapid convergence and high accuracy, making it a scalable and adaptable solution for dynamic energy markets. Its robust performance highlights the importance of integrating temporal dependencies for reliable demand forecasting. Future work may explore localized spatial features and renewable energy variability conditions such as low demand on power systems to further enhance its applicability.

VI. References

[1] C. Maduabuchi *et al.*, "Renewable Energy Potential Estimation Using Climatic-Weather-Forecasting Machine Learning Algorithms," *Energies (Basel)*, vol. 16, no. 4, Feb. 2023, doi: 10.3390/en16041603.

[2] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, "A review of deep learning for renewable energy forecasting," Oct. 15, 2019, *Elsevier Ltd*. doi: 10.1016/j.enconman.2019.111799.

[3] R. Malyala, K. Thattai, A. Malik, and J. Ravishankar, "Weather-Independent Forecasting for State-Wide Energy Markets Using Hybrid GPI-DSSM Model," Sydney, Australia, Nov. 2024. [Online]. Available: https://orcid.org/0009-0002-3502-8487

[4] "Distribution zone substation data - Ausgrid." Accessed: Jan. 20, 2025. [Online]. Available: https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Distribution-zone-substation-data

[5] "🏛 Historical Weather API | Open-Meteo.com." Accessed: Jan. 20, 2025. [Online]. Available: https://open-meteo.com/en/docs/historical-weather-api#hourly=temperature_2m,relative_humidity_2m,apparent_temperature,precipitation,cloud_cover,wind_speed_10m,sunshine_duration,shortwave_radiation_instant,direct_radiation_instant&csv_coordinates=Sydney

[6] N. Bacanin *et al.*, "Multivariate energy forecasting via metaheuristic tuned long-short term memory and gated recurrent unit neural networks," *Inf Sci (N Y)*, vol. 642, Sep. 2023, doi: 10.1016/j.ins.2023.119122.

[7] Z. Su, G. Zheng, M. Hu, L. Kong, and G. Wang, "Short-term load forecasting of regional integrated energy system based on spatio-temporal convolutional graph neural network," *Electric Power Systems Research*, vol. 232, Jul. 2024, doi: 10.1016/j.epsr.2024.110427.

[8] J. Liu and Y. Fu, "Renewable energy forecasting: A self-supervised learning-based transformer variant," *Energy*, vol. 284, Dec. 2023, doi: 10.1016/j.energy.2023.128730.

[9] B. Jie, J. Baba, and A. Kumada, "Mathematical Analysis of Time-Series & Meteorological Factors in Electricity Demand Forecasting Based on Carbon Neutrality," in *2023 IEEE IAS Industrial and Commercial Power System Asia, I and CPS Asia 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 137–142. doi: 10.1109/ICPSAsia58343.2023.10294862.

[10] A. Srinivasan, R. Wu, P. Heer, and G. Sansavini, "Impact of forecast uncertainty and electricity markets on the flexibility provision and economic performance of highly-decarbonized multi-energy systems," *Appl Energy*, vol. 338, May 2023, doi: 10.1016/j.apenergy.2023.120825.

[11] Q. Zhang, J. Chen, G. Xiao, S. He, and K. Deng, "TransformGraph: A novel short-term electricity net load forecasting model," *Energy Reports*, vol. 9, pp. 2705–2717, Dec. 2023, doi: 10.1016/j.egyr.2023.01.050.