

Enhanced Load Forecasting with GAT-LSTM: Leveraging Grid and Temporal Features

Ugochukwu Orji
Jheronimus Academy of Data Science
(JADS), Tilburg University
Tilburg, The Netherlands
u.e.orji@tilburguniversity.edu

Çiçek Güven
Department of Cognitive Science and
Artificial Intelligence, Tilburg
University
Tilburg, The Netherlands
c.guven@tilburguniversity.edu

Dan Stowell
Department of Cognitive Science and
Artificial Intelligence, Tilburg
University
Tilburg, The Netherlands
Naturalis Biodiversity Center
Leiden, The Netherlands
d.stowell@tilburguniversity.edu

ABSTRACT

Accurate power load forecasting is essential for the efficient operation and planning of electrical grids, particularly given the increased variability and complexity introduced by renewable energy sources. This paper introduces GAT-LSTM, a hybrid model that combines Graph Attention Networks (GAT) and Long Short-Term Memory (LSTM) networks. A key innovation of the model is the incorporation of edge attributes, such as line capacities and efficiencies, into the attention mechanism, enabling it to dynamically capture spatial relationships grounded in grid-specific physical and operational constraints. Additionally, by employing an early fusion of spatial graph embeddings and temporal sequence features, the model effectively learns and predicts complex interactions between spatial dependencies and temporal patterns, providing a realistic representation of the dynamics of power grids. Experimental evaluations on the Brazilian Electricity System dataset demonstrate that the GAT-LSTM model significantly outperforms state-of-the-art models, achieving reductions of 21.8% in MAE, 15.9% in RMSE and 20.2% in MAPE. These results underscore the robustness and adaptability of the GAT-LSTM model, establishing it as a powerful tool for applications in grid management and energy planning.

KEYWORDS

Short-Term Load Forecasting, Spatial-Temporal Analysis, Graph Attention Network, Long Short-Term Memory, Hybrid models

1 INTRODUCTION

Modern power systems are intricate networks of interconnected components responsible for generating, transmitting, distributing, and utilizing electricity. A key challenge in these systems is maintaining the balance between supply and demand, especially with the rapid integration of renewable energy sources (RES) such as solar and wind into grid operations. The intermittent nature of RES introduces variability, which requires the deployment of new technologies that enhance grid flexibility and enable real-time responses.

Consequently, accurate demand estimation is fundamental for effective power planning as matching supply with demand is essential to maintain grid stability. Note that errors in demand forecasting can result in significant financial costs, with even a 1% forecasting error potentially leading to hundreds of thousands of dollars in losses per GWh [39]. Since forecasting plays a critical role in

numerous tasks, improving both the accuracy and efficiency of forecasting processes is crucial.

Traditionally, load forecasting has relied on statistical methods that estimate relationships within time-series data, sometimes incorporating external factors. Classical statistical models such as ARIMA [13] and exponential smoothing [36, 47], as well as various machine learning approaches [10], have been utilized. In addition, load forecasting is influenced by numerous factors, many of which are location-specific and depend on the equipment used [5]. Factors such as weather conditions, demographics, socioeconomic variables, and special events such as holidays that can impact power demand are often incorporated as input variables in forecast models [4].

In modern electricity markets, forecasts are generated for various time horizons, each supporting grid operations and market management. Very short-term forecasts, spanning seconds to minutes, are critical for emergency operations and optimizing Distributed Energy Resources (DER). Short-term forecasts, ranging from minutes to a day ahead, are vital for real-time market trading, power plant operations, grid balancing, and managing operating reserves. Medium-term forecasts, covering days to months, aid in pre-dispatch, unit commitment, and maintenance planning, while long-term forecasts, extending months to years, support system planning, investment decisions, and maintenance scheduling [14, 44, 54].

Among these, short-term load forecasting (STLF) is particularly crucial for daily balancing in grid operations and facilitates real-time decision-making for unit dispatching, peak load analysis, and automatic generation control, ensuring efficient grid operation under dynamic conditions, especially as RES are integrated [14, 44, 49, 54].

Artificial intelligence (AI) techniques and deep neural networks (DNNs) have gained popularity for load forecasting, including expert systems [45], support vector machines (SVM) [17], fuzzy logic [42], artificial neural networks (ANNs) [33] and long-short-term memory networks (LSTMs) [29]. However, despite their ability to incorporate external influences, these models have limitations, such as the risk of getting stuck in local minima, overfitting [55], and inability to fully capture complex spatial-temporal dependencies.

To address these challenges, hybrid models that combine multiple forecasting methods have been proposed to further enhance accuracy and reliability [5, 18, 37]. These hybrid models offer the advantage of capturing the spatial and temporal features of the electricity load while addressing the limitations of individual methods.

However, despite their advantages, hybrid models also introduce additional complexity in model design, implementation, and parameter optimization [5].

Key Challenges with Traditional and State-of-the-Art Load Forecasting Models

- (1) **Limited Incorporation of Grid-Specific Features:** Traditional load forecasting models, and even the current advanced methods, often overlook or inadequately handle grid-specific information (e.g. transmission capacities, efficiencies), which are essential for capturing power flow constraints and line losses in the grid. These models focus mainly on dynamic data (e.g., energy demand at each location, external covariates, etc.). Failing to incorporate these features limits the model's ability to accurately represent real-world grid dynamics, leading to less robust predictions.
- (2) **Limited Fusion of Spatial and Temporal Information:** Many load forecasting models treat spatial and temporal information separately, either through sequential models (like LSTMs) that ignore spatial context or graph-based models that lack robust temporal modeling. This split approach fails to capture how spatial dependencies and temporal patterns interact, limiting the model's ability to adapt to sudden changes in load patterns or energy flow dynamics.
- (3) **Lack of Integration of RES Data:** The intermittent and variable nature of RES, such as solar and wind, introduces significant uncertainty into load forecasts. Most current models do not account for RES data, leading to predictions that may not accurately reflect fluctuations in energy supply and demand.

In this study, we present a hybrid GAT-LSTM fusion model for STLF that addresses key limitations of existing forecasting approaches. The proposed model integrates grid-specific attributes, handles spatial-temporal data fusion, and incorporates RES data to provide more realistic and reliable load predictions. By capturing complex interactions within modern power grids, our approach offers enhanced accuracy and robustness, making it well suited to the intricacies of today's dynamic energy landscape.

The rest of the paper is organized as follows; Section 2 presents reviews of key concepts, including energy forecasting techniques and graph neural networks, Section 3 describes the data and introduces our hybrid GAT-LSTM model, Section 4 presents the experimental results and discusses the efficacy of our approach, and finally, Section 5 concludes with future directions.

2 REVIEW OF KEY CONCEPTS

As discussed, forecasting models predict future energy demand and supply by analyzing historical time-series data and relevant covariates to uncover important dynamics within the data. This section provides a brief overview of some widely employed energy time-series forecasting techniques in the literature.

2.1 Traditional Energy Forecasting Models

Traditional statistical methods for energy forecasting, such as autoregressive models and exponential smoothing techniques, are foundational approaches that use historical data to predict future

energy demand. These models are widely used for linear and stationary time series data.

Autoregressive models, such as ARMA and its extensions (ARIMA and SARIMA), and exponential smoothing techniques, have been widely used for STLF. ARIMA models relationships among current values, past values, and previous errors, with SARIMA incorporating seasonal differencing for datasets with seasonality [14, 38].

Exponential smoothing methods, including Holt-linear and Holt-Winters extensions, use weighted averages of past observations to capture trends and seasonality in non-stationary data [36]. These approaches are valued for their simplicity and effectiveness in linear time series data [13, 15, 36]. However, they are limited in handling non-linear, high-dimensional data and fail to capture the complex spatial and temporal dependencies inherent in energy systems. These limitations are especially evident in modern energy systems, where demand patterns are shaped by dynamic factors such as weather, renewable energy integration, and network topology.

2.2 Deep Learning-Based Energy Forecasting Models

Deep learning, a subfield of machine learning, leverages neural networks to automatically learn patterns from data, making it particularly effective for modeling complex, non-linear relationships in large datasets [7]. Sequence processing models such as recurrent neural networks (RNNs) and long-short-term memory networks (LSTMs) are widely used for time-based predictions, as they retain information across time steps and capture temporal dependencies [50, 51].

RNNs model sequential data by updating a hidden state based on the current input and the previous hidden state. While effective for short-term dependencies, they struggle with vanishing or exploding gradients, limiting their ability to capture long-term dependencies [31, 40, 41].

LSTMs address these challenges through a more sophisticated architecture featuring a memory cell and gating mechanisms (forget, input, and output gates) that regulate information flow. This design enables LSTMs to capture both long-term and short-term dependencies in time-series data [21, 22, 26].

RNNs and LSTMs have been extensively applied to load forecasting, where they effectively model temporal dependencies and dynamic patterns in energy time-series data [8, 19, 29, 43]. However, these models are limited to temporal dependencies and do not account for the spatial interactions inherent in energy systems. To address this limitation, hybrid models such as the proposed GAT-LSTM combine the strengths of LSTMs for temporal modeling with GNNs for spatial dependencies. This integration provides a holistic representation of energy systems by capturing both time-series dynamics and grid topology.

2.3 Graph Neural Networks (GNNs) for Energy Forecasting

Recent advancements in deep neural networks (DNNs) have extended their applications to graph-structured (non-Euclidean) data, enabling the modeling of complex relationships inherent in graphs [23,

35, 46]. GNNs are particularly effective in capturing spatial dependencies in domains such as social networks, recommendation systems, and energy systems, where power grids naturally exhibit graph structures [27]. By combining graph topology with node and edge attributes, GNNs facilitate the modeling of spatial interactions critical for understanding the dynamics of energy networks.

Graph-Structured Data and GNN Architecture: A graph $G = (V, E)$ consists of nodes (V) and edges (E), which may be directed or undirected. The graph structure is represented by an adjacency matrix A , where $a_{ij} = 1$ if an edge exists between nodes i and j , and $a_{ij} = 0$ otherwise. Node features x_i are organized into a feature matrix $X \in \mathbb{R}^{N \times F}$, where N is the number of nodes and F is the number of features per node.

The core operation in GNNs is the graph convolution, where nodes aggregate information from their neighbors to update their representations. At layer l , this is defined as:

$$h_i^{(l+1)} = \text{Agg} \left(h_j^{(l)} : j \in N(i) \cup i \right) \quad (1)$$

where $h_i^{(l)}$ is the embedding of node i at layer l , $N(i)$ denotes the neighbors of node i , and Agg is an aggregation function (e.g., sum, mean, or max). After aggregation, embeddings are passed through a non-linear activation function, such as ReLU.

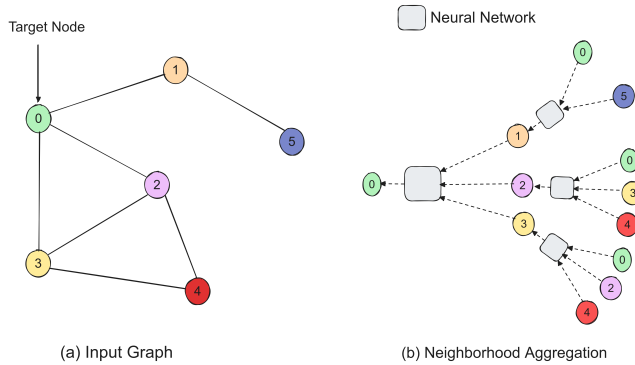


Figure 1: Neighborhood Aggregation in GNNs. Source [30].

Figure 1 illustrates neighborhood aggregation in GNNs. For energy systems, nodes may represent grid regions or substations, and edges capture relationships like shared demand or transmission lines. This process enables GNNs to learn spatial dependencies and provide context-aware predictions.

Spectral- and Spatial-Based GNNs: GNNs can be broadly categorized into spectral or spatial-based [9, 12, 24, 25, 48]. Spectral-based GNNs, such as Graph Convolutional Networks (GCNs), apply filters in the Fourier domain [11, 32, 53], leveraging the normalized adjacency matrix \hat{A} to update node embeddings:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}) \quad (2)$$

where $H^{(l)}$ is the feature matrix at layer l , $W^{(l)}$ is the weight matrix, and σ is the activation function. GCNs are effective for tasks requiring localized information aggregation [35, 39].

Spatial-based GNNs, such as Graph Attention Networks (GATs), use attention mechanisms to assign importance to neighboring

nodes during aggregation [6, 20]. The attention coefficient α_{ij} is computed as:

$$\alpha_{ij} = \frac{\exp(\sigma(a^T [Wh_i || Wh_j]))}{\sum_{k \in N(i)} \exp(\sigma(a^T [Wh_i || Wh_k]))} \quad (3)$$

where a is a learnable attention vector and $||$ denotes concatenation. Multi-head attention refines this process further, enhancing robustness [28].

Applications and Limitations in Load Forecasting: GNNs, including GCNs and GATs, have been used to improve load forecasting by modeling grid topology and spatial dependencies [28, 35, 39]. However, GNNs alone cannot capture temporal dependencies inherent in energy systems, which are critical for forecasting dynamic load patterns.

2.4 Hybrid Models for Energy Forecasting

Hybrid models, such as the proposed GAT-LSTM, address the limitations of standalone GNNs and LSTMs by combining their strengths. While GATs model spatial dependencies in grid topology, LSTMs handle temporal dynamics by capturing sequential patterns in energy consumption. This integration enables a holistic representation of energy systems, accounting for both spatial interactions and time-series dynamics [18, 28, 37, 39].

The novelty of the GAT-LSTM lies in its ability to adapt graph-structured relationships and temporal patterns simultaneously. For example, GATs use attention mechanisms to focus on the most relevant grid regions, while LSTMs capture the evolving energy consumption trends. Together, they synergize to improve forecasting accuracy, particularly for complex, interconnected energy networks where traditional models and standalone deep learning methods fall short.

This hybrid approach has been extensively studied in domains like traffic forecasting [52, 56, 57], but remains underexplored in energy systems. By incorporating grid-specific features and leveraging the spatio-temporal interplay, the GAT-LSTM provides robust and context-aware predictions for energy time series forecasting, marking a significant advancement in the field.

3 METHODOLOGY

3.1 Data

3.1.1 Data Source and Description: For this study, we used the Brazilian power system as a case study with actual data on various aspects of the power system and covariate factors. The data sources and description is given in Tables 1 and 2, respectively. Except for the graph data, which are static, all other aspects of the dataset represent 2-years worth of data from 2019–2020.

3.1.2 Data Preprocessing: The data were cleaned and preprocessed for modeling through the following steps:

- (1) **Consolidation of Weather Data:** The weather data consisted of observations from various stations in different states of Brazil, with some states having multiple stations. To ensure consistent hourly data for each weather variable per state, we calculated the mean and standard deviation (STD) of each variable, grouped by state and datetime.

Table 1: Data Sources

Data	Details	Source
Electricity	Load, PV, wind, etc.	[16]
Grid	Line length, capacity, efficiency, etc.	[16]
Weather	Temperature, pressure, rainfall, etc.	[1]
Socio-economic	State-wise GDP	[3]
Population	State-wise population	[2]

Table 2: Data Description

Category	Details	Horizon
Sequence Data	Atmospheric pressure, Total hourly rain, Global radiation, Air temperature, Dew point temperature, Relative humidity, Wind direction, Wind maximum gust, Wind speed, PV generation, Onshore wind generation, Offshore wind generation, Load profile (by consumption)	1-hour
Graph Data	Node features: Source (state), Target (state), PV potential, Onshore wind potential, Offshore wind potential, geometry (longitude & latitude) Edge attributes: Line capacity, Line efficiency, Line length, Line carrier	Static
Socio-economic	Population, GDP, Total plant capacity	Annual
Calendar	Year, Quarter, Month, Day, Hour, Day-of-week, Week-of-Year, Holiday, Season	Annual

- (2) **Missing Value Imputation:** Analysis showed approximately 21.42% of missing data across the weather dataset, which was filled by interpolation of time series to maintain continuity in the data.
- (3) **Handling Insufficient Data:** One state had only a single weather station, resulting in NaNs for the STD values. We addressed this by setting the STD values for that state to zero.
- (4) **Negative Value Correction:** Negative values were observed in the PV variable, which were corrected by replacing them with zero.
- (5) **Data Scaling:** To handle variance and outliers, we scaled the dataset using RobustScaler from the Python Scikit-learn package.
- (6) **Data Splitting:** A dynamic split was performed to create training, validation, and test sets as follows:
 - Training set: January–December 2019
 - Validation set: January–June 2020

- Test set: July–December 2020

- (7) **Target Variable Creation:** Target values for prediction were created by shifting the current load by one hour to facilitate next-hour load forecasting.
- (8) **State-wise Sequencing:** Finally, we generated state-wise sequences, aligning data by state and timestamp to feed into the model.

3.2 Problem Definition and Model Architecture

Let $X = \{x_1, x_2, \dots, x_p\}$ represent a historical sequence of multidimensional variables in p time steps, where each $x_t \in \mathbb{R}^N$ is a vector of N features at time t . Each vector x_t includes the power load at time t and other covariates that influence future load (e.g., temperature, seasonal and holiday variables, etc.). The objective of this study is to forecast the next hour's power load, y_{p+1} , by leveraging these historical data alongside the spatial grid information and RES data.

To achieve this, we propose a hybrid GAT-LSTM model¹ as shown in Figure 2 and described below:

Phase 1 (Get Node Embeddings): The model employs 2 parallel GAT layers to compute node embeddings that reflect spatial dependencies and grid-specific constraints. Unlike traditional attention mechanisms, which typically rely only on node features, our approach also incorporates edge attributes (i.e., line capacities, efficiencies, etc.) into the calculation of attention coefficients. Thus, we modify Equation 3 to incorporate edge attributes such that, for a pair of connected nodes i and j with respective node features h_i and h_j and edge attribute e_{ij} , the attention coefficient α_{ij} becomes:

$$\alpha_{ij}^k = \frac{\exp\left(\sigma\left(a^k \cdot [W^k h_i \parallel W^k h_j \parallel U^k e'_{ij}]\right)\right)}{\sum_{k \in N(i)} \exp\left(\sigma\left(a^k \cdot [W^k h_i \parallel W^k h_k \parallel U^k e'_{ik}]\right)\right)} \quad (4)$$

Where:

- W^k = learnable transformation matrix specific to attention head k of node features h_i
- U^k = learnable transformation matrix specific to attention head k of edge attributes e'_{ij}
- a^k = learnable vector that projects the concatenated inputs into a scalar (used for attention scoring)
- σ = LeakyReLU activation function

Based on the multi-head attention mechanism, the aggregation and update function for each head is given as:

$$h'_i = \text{Concat}\left(\sigma\left(\sum_{j \in N(i)} \alpha_{ij}^k W^k h_j\right)\right), \forall k \in \{1, \dots, K\} \quad (5)$$

This formulation allows the model to dynamically learn the importance of each connection based on both node and edge attributes, yielding node embeddings that better capture the underlying power grid structure and constraints.

¹Our model code is available under an open license: <https://github.com/ugoorji12/Load-Forecasting-using-GAT-LSTM>

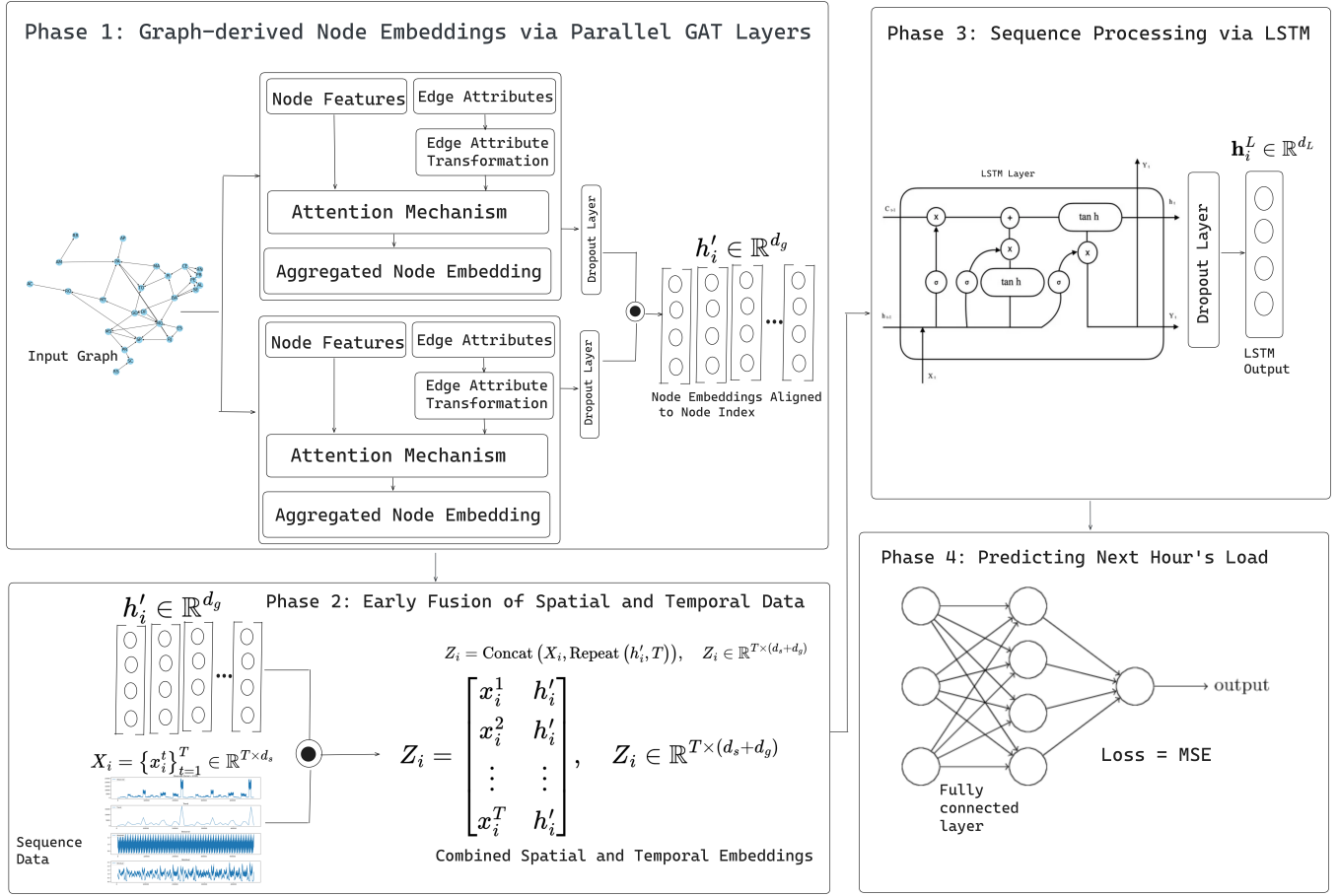


Figure 2: Model Architecture

Phase 2 (Early Fusion of Spatial and Temporal Data): The goal of this phase is to combine the graph-derived node embeddings h'_i from the GAT layer (Phase 1) with temporal sequence data. We achieve this by first expanding the h'_i along the temporal dimension to match the sequence data as represented in Equation 6.

$$Z_i = \begin{bmatrix} x_i^1 & h'_i \\ x_i^2 & h'_i \\ \vdots & \vdots \\ x_i^T & h'_i \end{bmatrix}, \quad Z_i \in \mathbb{R}^{T \times (d_s + d_g)} \quad (6)$$

The final concatenation is given as:

$$Z_i = \text{Concat}(X_i, \text{Repeat}(h'_i, T)), \quad Z_i \in \mathbb{R}^{T \times (d_s + d_g)} \quad (7)$$

Where:

- $h'_i \in \mathbb{R}^{d_g}$ = The graph-derived embedding for node i , where d_g is the GAT output dimension (after concatenation across heads).
- $X_i = \{x_i^t\}_{t=1}^T \in \mathbb{R}^{T \times d_s}$ = The temporal sequence data for node i , T is the sequence length (e.g., 24 hours), and d_s is the feature dimension of the sequence data.

This early fusion strategy ensures that both spatial and temporal information are jointly modeled in the downstream LSTM layer.

Phase 3 (LSTM Layer for Sequential Processing): The combined spatial-temporal data are fed into an LSTM layer, which learns the combined dependencies, capturing how past load trends and covariates evolve over time to influence future loads. This layer enables the model to retain and process the long-term spatio-temporal dependencies in the data.

Phase 4 (Final Prediction Layer): The output from the LSTM layer is fed into a fully connected layer that generates the forecast for the next hour's load, y_{p+1} .

3.3 Training Process

Loss Function and Optimization: The training process used the mean squared error (MSE) as the loss function for both training and validation. MSE is particularly suited for regression tasks as it heavily penalizes larger errors, ensuring a focus on minimizing significant deviations in predictions.

For optimization, the Adam optimizer was chosen due to its effective combination of momentum and adaptive learning rates, which makes it well suited for handling complex models. Additionally, a

ReduceLROnPlateau learning rate scheduler was used to dynamically adjust the learning rate. If the validation loss did not improve for five consecutive epochs, the learning rate was reduced by a factor of 0.1 (90% decrease), promoting more effective convergence.

To prevent overfitting, Early Stopping was applied, halting the training process if validation loss showed no improvement for 10 consecutive epochs. This ensured that the training process was stopped once convergence was achieved.

These techniques work together to ensure efficient learning, smoother convergence, and better generalization.

4 EXPERIMENTS AND RESULTS

4.1 Experimental Setup

The experiments were carried out in a high performance computing environment featuring a dual-socket architecture with 12 physical cores (24 logical CPUs) and 192 GB of memory. The system includes 3 NVIDIA Tesla GPUs and ran on Linux (Debian 6.1), with Slurm (version 22.05.8) used for job scheduling. The hyperparameters used in the model were carefully selected, with details provided in Table 3.

Table 3: Description of Model Parameters

Parameter	Value
Sequence length	24
Batch size	27
GAT output layer	64
GAT attention heads	8
LSTM hidden-state	128
Number of LSTM layers	4
Learning rate	0.0001
Weight decay	1×10^{-5}
GAT Dropout	0.2
LSTM Dropout	0.3
Epochs	200

4.2 Baseline Models

The following baseline models were used to compare the performance of our proposed approach:

GCN-LSTM: This hybrid model combines Graph Convolutional Network (GCN) layers with a Long Short-Term Memory (LSTM) network for load forecasting. It mirrors the GAT-LSTM architecture by applying two parallel 1-hop GCN layers, each performing a 1-hop convolution on the node features, followed by dropout for regularization. The outputs of these GCN layers are concatenated and indexed for the nodes corresponding to the input sequences. These combined GCN features are expanded to match the temporal sequence length and concatenated with the sequence data. The LSTM layer captures temporal dependencies, and a fully connected layer generates the final output. Unlike GAT-LSTM, this model does not incorporate edge attributes, as standard GCNs focus exclusively on node features and graph structure.

EdgeGCN-LSTM: An extension of GCN-LSTM, this model integrates edge attributes into the message-passing process using a

custom EdgeAttrGCNConv layer. Edge attributes are transformed via a linear layer before aggregation, allowing the model to utilize both node and edge information effectively. The aggregated features are passed to an LSTM layer to capture temporal dynamics and generate predictions.

LSTM: Serving as a sequence forecasting baseline, this model focuses solely on temporal data without incorporating graph-based features. It uses an LSTM network to learn temporal dependencies and produce forecasts based on dynamic time-series data.

XGBoost: A popular tree-based regression model, XGBoost is used here as a baseline for evaluating performance on time-series data. Like the LSTM model, it does not include graph-based features and relies entirely on dynamic data for forecasting.

4.3 Evaluation Metrics

Our evaluation metrics include Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

Concretely, these metrics are widely used to evaluate the accuracy of single or hybrid load forecasting models [5, 44].

4.4 Experiment Results and Discussion

Training Performance and Early Stopping Analysis

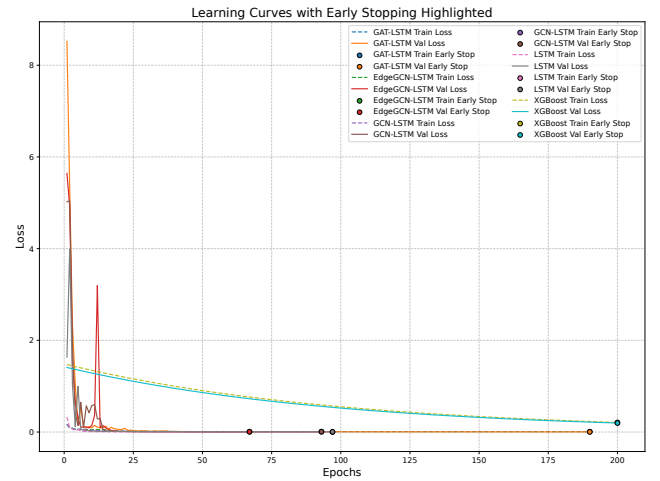


Figure 3: Learning curve for all models.

Figure 3 presents the learning curves for all models during training, highlighting their convergence behavior and potential overfitting risks. GAT-LSTM demonstrates rapid convergence with stable training and validation losses, suggesting efficient training and generalization. Early stopping was applied around optimal epochs to prevent overfitting. In contrast, EdgeGCN-LSTM and GCN-LSTM show steady declines in training loss but exhibit fluctuations in validation loss, indicating minor risks of overfitting. LSTM converges more slowly, with its validation loss stabilizing at a higher value, reflecting its weaker capacity to model the data comprehensively. Lastly, XGBoost displays the poorest training performance, with consistently high errors throughout, underscoring its limitations in capturing temporal dependencies.

Model Comparison Based on Accuracy Metrics

Table 4: Experiment Results

Model	MAE (MW)	RMSE (MW)	MAPE (%)
GAT-LSTM	64.64	119.06	4.59
LSTM	82.68	141.55	5.75
EdgeGCN-LSTM	84.63	148.09	7.24
GCN-LSTM	89.11	184.12	5.72
XGBoost	297.47	517.69	40.50

Table 4 summarizes the performance of all models based on MAE, RMSE, and MAPE. GAT-LSTM achieves the best overall accuracy, outperforming other models across all metrics. Specifically, it shows a 21.82% improvement in MAE compared to LSTM and a 23.62% improvement over EdgeGCN-LSTM. This highlights the ability of GAT-LSTM to effectively capture both spatial and temporal dependencies, leveraging the graph-attention mechanism. LSTM, despite lacking spatial awareness, performs better than both EdgeGCN-LSTM and GCN-LSTM due to its robust temporal modeling capabilities. The underperformance of EdgeGCN-LSTM relative to LSTM suggests that incorporating spatial relationships without attention mechanisms might introduce irrelevant or noisy information, hindering forecasting accuracy. Similarly, GCN-LSTM performs the worst among GNN-based models, indicating that suboptimal spatial features can reduce model effectiveness. Lastly, XGBoost demonstrates the highest errors across all metrics, reflecting its inability to model non-linear temporal dependencies and spatial relationships critical for load forecasting.

Analysis of Mean Actual vs Predicted Load Curves

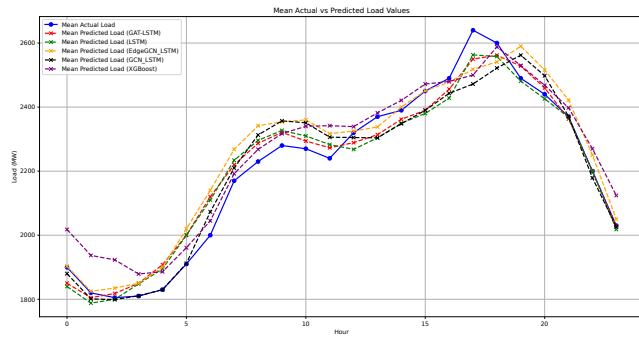


Figure 4: Mean Actual vs Predicted Load Values for all Models.

Figure 4 compares the mean actual and predicted load profiles across all models. GAT-LSTM closely follows the actual load curve throughout the day, reflecting its superior accuracy and ability to generalize across varying load conditions. LSTM and EdgeGCN-LSTM show reasonable alignment with the actual load curve but exhibit larger deviations during peak and off-peak transitions, particularly in high-gradient periods. GCN-LSTM struggles more during peak hours, with noticeable deviations during high-load periods. In contrast, XGBoost displays significant errors, including peak and

off-peak times, confirming its limited ability to generalize load dynamics effectively.

Performance of GAT-LSTM During Peak vs Off-peak Hours

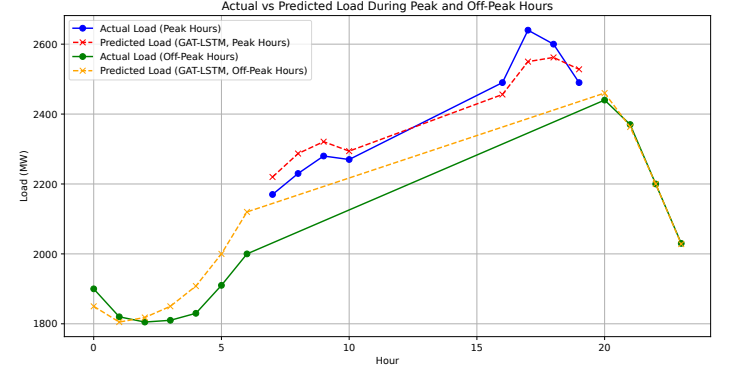


Figure 5: Peak vs Off-peak Performance for GAT-LSTM.

Figure 5 illustrates the average performance of GAT-LSTM during peak and off-peak periods across the test set. During peak hours (7:00 AM–10:00 AM and 4:00 PM–7:00 PM), GAT-LSTM consistently tracks actual load trends but exhibits slight deviations during sharp increases, such as the morning ramp-up. Evening peaks are generally better captured, although minor overestimations occur after the peak (e.g., hours 18–19). During off-peak hours (12:00 AM–6:00 AM and 8:00 PM–11:00 PM), predictions align well with actual loads, although a pattern of overestimation is observed during early off-peak periods (e.g., hours 2–5). This performance demonstrates the model’s robustness, but also highlights areas for further improvement, such as better handling of high-gradient patterns.

4.5 Key Insights and Discussion

The experimental results underscore several critical insights:

Effectiveness of Attention Mechanisms: The GAT-LSTM model demonstrates superior accuracy by effectively capturing spatial and temporal dependencies using attention mechanisms. Its ability to prioritize relevant node and edge attributes enables robust modeling of grid-specific constraints, such as line capacities. However, the model struggles slightly during rapid transitions, such as morning ramp-ups, suggesting that the temporal component could benefit from further fine-tuning. The integration of spatial embeddings and temporal features remains a key strength, enabling GAT-LSTM to model complex interactions with high robustness for real-world electricity forecasting.

Limitations of Basic Graph Architectures: Both EdgeGCN-LSTM and GCN-LSTM perform worse than LSTM, indicating that spatial features, when modeled without attention mechanisms, may introduce noise or irrelevant information, reducing forecasting accuracy.

Temporal Strength of LSTM: Despite its lack of spatial awareness, LSTM’s strong temporal modeling capabilities allow it to outperform EdgeGCN-LSTM and GCN-LSTM. This highlights the importance of robust temporal modeling in energy forecasting tasks.

Limitations of XGBoost: XGBoost exhibits the highest errors, confirming its inability to model sequential relationships and temporal dependencies effectively. Its tree-based approach further limits its ability to capture the non-linear temporal dynamics critical for load forecasting.

5 CONCLUSION, LIMITATIONS, AND FUTURE WORK

In this paper, we introduce and evaluate the GAT-LSTM model for hourly power load forecasting, combining GAT and LSTM to effectively capture spatial and temporal dependencies in electricity grids. Our results demonstrate that GAT-LSTM consistently outperforms state-of-the-art models, across key metrics such as MAE, RMSE, and MAPE. This superior performance arises from the model's ability to leverage graph-based attention mechanisms to extract meaningful spatial features while utilizing the LSTM's strength in modeling sequential patterns.

Despite its strong performance, the model has notable limitations. The integration of graph attention mechanisms and LSTM layers introduces significant computational complexity, potentially hindering scalability for large datasets or resource-constrained environments. Furthermore, the model's accuracy is influenced by the quality of the graph structure; incomplete or inaccurate node connections, often due to missing or imprecise data in electrical grid representations, can reduce its effectiveness. These issues are relatively common in real-world grids due to challenges such as missing data, approximations in topology, and outdated infrastructure records. Additionally, while GAT-LSTM performs well during stable off-peak periods, it struggles to fully capture rapid transitions, such as morning ramp-ups, leading to deviations from actual load values. This limitation is not unique to GAT-LSTM, but reflects a broader challenge in forecasting highly dynamic events, where past information may not adequately represent future behavior.

Future work should address these challenges by enhancing the model's ability to handle rapid load transitions during peak hours and reducing sensitivity to low-magnitude variations in off-peak periods. Adaptive learning techniques, automated graph refinement, and the inclusion of additional external covariates, such as market data and maintenance records, could improve the model's robustness and contextual understanding. Leveraging attention mechanisms to identify and prioritize critical graph elements, such as influential nodes or edges, can further enhance interpretability and provide deeper insights into the spatial-temporal factors driving predictions. Additionally, integrating uncertainty quantification methods alongside these mechanisms would improve decision-making reliability and broaden the model's applicability in real-world scenarios.

In conclusion, the GAT-LSTM model represents a significant advancement in power load forecasting by effectively modeling spatial-temporal relationships in electricity grids. Its ability to identify and prioritize informative graph components allows it to capture critical spatial dependencies that underpin accurate predictions. Addressing its limitations related to computational complexity and peak-hour accuracy, along with further refinements for scalability and reliability, will enhance its practicality. Visualizing the graph structure to identify central or influential nodes could also provide

valuable insights, improving both interpretability and optimization. These advancements will solidify GAT-LSTM's role as a powerful tool for grid management, demand response, and energy planning in dynamic and interconnected energy systems.

ACKNOWLEDGMENTS

This research is part of the project Innovation Lab for Utilities on Sustainable Technology and Renewable Energy (ILUSTRE), No.KICH3.LTP.20.006 of the research program LTP ROBUST which is partly financed by the Dutch Research Council (NWO).

REFERENCES

- [1] 2024. Brazil Weather Information by INMET. <https://www.kaggle.com/datasets/gregoryoliveira/brazil-weather-information-by-inmet?resource=download>. Accessed: 14/06/2024.
- [2] 2024. Estimates of Resident Population for Municipalities and Federation Units | IBGE. <https://www.ibge.gov.br/en/statistics/social/population/18448-estimates-of-resident-population-for-municipalities-and-federation-units.html?edicao=28688>. Accessed: 14/06/2024.
- [3] 2024. Regional Accounts of Brazil | IBGE. <https://www.ibge.gov.br/en/statistics/economic/national-accounts/16855-regional-accounts-of-brazil.html>. Accessed: 14/06/2024.
- [4] R. Ahmed, V. Sreeram, Y. Mishra, and M. D. Arif. 2020. A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization. *Renewable and Sustainable Energy Reviews* 124 (2020), 109792.
- [5] A. Al Mamun, M. Sohel, N. Mohammad, M. S. H. Sunny, D. R. Dipta, and E. Hossain. 2020. A comprehensive review of the load forecasting techniques using single and hybrid predictive models. *IEEE Access* 8 (2020), 134911–134939.
- [6] Dzmitry Bahdanau. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
- [7] R. F. Berriel, A. T. Lopes, A. Rodrigues, F. M. Varejao, and T. Oliveira-Santos. 2017. Monthly energy consumption forecast: A deep learning approach. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 4283–4290. <https://doi.org/10.1109/IJCNN.2017.7966398>
- [8] Filippo Maria Bianchi, Enrico Maiorino, Michael C Kampffmeyer, Antonello Rizzi, and Robert Jenssen. 2017. Recurrent neural networks for short-term load forecasting: an overview and comparative analysis. (2017).
- [9] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).
- [10] P. Bunnoon, K. Chalermyanont, and C. Limsakul. 2010. A computing model of artificial intelligent approaches to mid-term load forecasting: a state-of-the-art survey for the researcher. *International Journal of Engineering and Technology* 2, 1 (2010), 94–100.
- [11] Pingping Cao, Zeqi Zhu, Ziyuan Wang, Yanping Zhu, and Qiang Niu. 2022. Applications of graph convolutional networks in computer vision. *Neural computing and applications* 34, 16 (2022), 13387–13405.
- [12] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 257–266.
- [13] E. Chodakowska, J. Nazarko, and Ł. Nazarko. 2021. Arima models in electrical load forecasting and their robustness to noise. *Energies* 14, 23 (2021). <https://doi.org/10.3390/en14237952>
- [14] L. Dannecker. 2015. *Energy time series forecasting: Efficient and accurate forecasting of evolving time series from the energy domain*. Springer.
- [15] C. Deb, F. Zhang, J. Yang, S. E. Lee, and K. W. Shah. 2017. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews* 74 (2017), 902–924.
- [16] Ying Deng, Karl-Kiên Cao, Wenxuan Hu, Ronald Stegen, Kai von Krbek, Rafael Soria, Pedro Rua Rodriguez Rochedo, and Patrick Jochem. 2023. Harmonized and open energy dataset for modeling a highly renewable Brazilian power system. *Scientific Data* 10, 1 (2023), 103.
- [17] X. Dong, S. Deng, and D. Wang. 2022. A short-term power load forecasting method based on k-means and SVM. *Journal of Ambient Intelligence and Humanized Computing* 13, 11 (2022), 5253–5267.
- [18] S. Eandi, A. Cini, S. Lukovic, and C. Alippi. 2022. Spatio-temporal graph neural networks for aggregate load forecasting. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
- [19] L. Fang and B. He. 2023. A deep learning framework using multi-feature fusion recurrent neural networks for energy consumption forecasting. *Applied Energy* 348 (2023), 121563.

- [20] Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. 2016. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344* (2016).
- [21] F. A. Gers, J. Schmidhuber, and F. Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural Computation* 12, 10 (2000), 2451–2471.
- [22] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. 2002. Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research* 3 (2002), 115–143.
- [23] M. Gori, G. Monfardini, and F. Scarselli. 2005. A new model for learning in graph domains. In *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*. IEEE, 729–734.
- [24] Will Hamilton, Zitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).
- [25] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* (2015).
- [26] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
- [27] X. Hu, J. Yang, Y. Gao, M. Zhu, Q. Zhang, H. Chen, and J. Zhao. 2024. Adaptive power flow analysis for power system operation based on graph deep learning. *International Journal of Electrical Power & Energy Systems* 161 (2024), 110166.
- [28] Chaokai Huang, Ning Du, Jiahao He, Na Li, Yifan Feng, and Weihong Cai. 2023. Multidimensional Feature-Based Graph Attention Networks and Dynamic Learning for Electricity Load Forecasting. *Energies* 16, 18 (2023), 6443.
- [29] N. Jin, F. Yang, Y. Mo, Y. Zeng, X. Zhou, K. Yan, and X. Ma. 2022. Highly accurate energy consumption forecasting model based on parallel LSTM neural networks. *Advanced Engineering Informatics* 51 (2022), 101442.
- [30] Z. Jin, Y. Wang, Q. Wang, Y. Ming, T. Ma, and H. Qu. 2022. Gnnlens: A visual analytics approach for prediction error diagnosis of graph neural networks. *IEEE Transactions on Visualization and Computer Graphics* 29, 6 (2022), 3024–3038.
- [31] A. Kag and V. Saligrama. 2021. Training recurrent neural networks via forward propagation through time. In *International Conference on Machine Learning*. PMLR, 5189–5200.
- [32] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [33] S. Kouhi and F. Keynia. 2013. A new cascade NN based method to short-term load forecast in deregulated electricity market. *Energy Conversion and Management* 71 (2013), 76–83.
- [34] E. Kyriakides and M. Polycarpou. 2007. Short-term electric load forecasting: A tutorial. *Trends in Neural Computation, Studies in Computational Intelligence* 35 (2007), 391–418.
- [35] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, and Y. Wang. 2021. A review of graph neural networks and their applications in power systems. *Journal of Modern Power Systems and Clean Energy* 10, 2 (2021), 345–360.
- [36] S. Lima, A. M. Gonçalves, and M. Costa. 2019. Time series forecasting using Holt-Winters exponential smoothing: An application to economic data. In *AIP Conference Proceedings*, Vol. 2186. AIP Publishing. <https://doi.org/10.1063/1.5137999>
- [37] W. Lin, D. Wu, and B. Boulet. 2021. Spatial-temporal residential short-term load forecasting via graph neural networks. *IEEE Transactions on Smart Grid* 12, 6 (2021), 5373–5384.
- [38] S. Makridakis and M. Hibon. 1997. ARMA models and the Box-Jenkins methodology. *Journal of Forecasting* 16, 3 (1997), 147–163.
- [39] H. Mansoor, M. S. Gull, H. Rauf, M. Khalid, and N. Arshad. 2024. Graph Convolutional Networks based short-term load forecasting: Leveraging spatial information for improved accuracy. *Electric Power Systems Research* 230 (2024), 110263.
- [40] M. C. Mozer. 1991. Induction of multiscale temporal structure. In *Advances in Neural Information Processing Systems*, Vol. 4.
- [41] Understanding LSTM Networks. 2015. *colah's blog*. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> Accessed: 13/09/2024.
- [42] S. C. Pandian, K. Duraiswamy, C. C. A. Rajan, and N. Kanagaraj. 2006. Fuzzy approach for short term load forecasting. *Electric Power Systems Research* 76, 6–7 (2006), 541–548.
- [43] J. A. Pérez-Ortiz, F. A. Gers, D. Eck, and J. Schmidhuber. 2003. Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets. *Neural Networks* 16, 2 (2003), 241–250.
- [44] P. Pinson. 2013. Wind energy: Forecasting challenges for its operational management. *Statist. Sci.* 28, 4 (2013), 564–585.
- [45] S. Rahman and R. Bhatnagar. 1988. An expert system based algorithm for short term load forecast. *IEEE Transactions on Power Systems* 3, 2 (1988), 392–399.
- [46] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2008), 61–80.
- [47] J.W. Taylor. 2009. Triple seasonal methods for short-term electricity demand forecasting. *European Journal of Operational Research* 204, 1 (2009), 139–152.
- [48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [49] G. K. Venayagamoorthy, K. Rohrig, and I. Erlich. 2012. One step ahead: Short-term wind power forecasting and intelligent predictive control based on data analytics. *IEEE Power and Energy Magazine* 10, 5 (2012), 70–78.
- [50] P. J. Werbos. 1990. Backpropagation through time: What it does and how to do it. *Proc. IEEE* 78, 10 (1990), 1550–1560.
- [51] R. J. Williams and D. Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* 1, 2 (1989), 270–280.
- [52] Tianlong Wu, Feng Chen, and Yun Wan. 2018. Graph attention LSTM network: A new model for traffic flow forecasting. In *2018 5th international conference on information science and control engineering (ICISCE)*. IEEE, 241–245.
- [53] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 7370–7377.
- [54] R. Zavadil. 2013. Renewable generation forecasting: The science, applications, and outlook. In *Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS)*. 2252–2260.
- [55] J. Zhang, Y. M. Wei, D. Li, Z. Tan, and J. Zhou. 2018. Short term electricity load forecasting using a hybrid model. *Energy* 158 (2018), 774–781.
- [56] Tianqi Zhang and Ge Guo. 2020. Graph attention LSTM: A spatiotemporal approach for traffic flow forecasting. *IEEE Intelligent Transportation Systems Magazine* 14, 2 (2020), 190–196.
- [57] Taomei Zhu, Maria Jesus Lopez Boada, and Beatriz Lopez Boada. 2024. Adaptive Graph Attention and Long Short-Term Memory-Based Networks for Traffic Prediction. *Mathematics* 12, 2 (2024), 255.