Bridging Spatial and Temporal Contexts: Sparse Transfer Learning

Daniel Persson*, William Wahlberg*, Anna Vettoruzzo, and Sławomir Nowaczyk

Center for Applied Intelligent Systems Research, Halmstad University, Sweden

Abstract. This paper introduces a novel transfer learning adapter, the Bridged Attention Module (BAM), designed to enhance the performance of Spatial-Temporal Graph Convolutional Networks (ST-GCN) in data-limited forecasting scenarios. BAM improves fine-tuning efficiency by jointly capturing spatial and temporal dependencies, optimizing information flow, and significantly reducing the number of trainable parameters while preserving model accuracy. Experimental evaluations demonstrate that the BAM-enhanced ST-GCN consistently achieves competitive accuracy and, in some cases, surpasses traditional fine-tuning methods, even with limited data. The effectiveness of this approach is validated using electric vehicle (EV) charging station occupancy forecasting, highlighting the practical utility of BAM.

Keywords: Transfer learning · Electric vehicles · Time series · Deep learning · Parameter-efficient learning

1 Introduction

Accurate spatial-temporal forecasting is essential across various fields. However, achieving high accuracy in these models can be challenging, particularly in scenarios with limited access to comprehensive datasets. This paper addresses the need for parameter-efficient forecasting methods that can deliver accurate predictions while relying on only a small amount of data. To demonstrate the proposed approach, electric vehicle (EV) charging station occupancy forecasting is used as a test case, reflecting the practical importance of scalable solutions in data-sparse environments. In the EV industry, where infrastructure is still developing and historical data may be scarce, reliable occupancy forecasting is critical for optimizing charging station networks. In 2022, global EV sales surpassed 10 million, increasing their share of total vehicle sales from 4% in 2020 to 12% [16]. This shift, driven by policy initiatives like the European Green Deal [9], underscores the need for efficient, data-sparse forecasting methods that can adapt to growing demands.

Accurate forecasting in data-limited scenarios has broad relevance as it enables strategic decision-making and efficient resource management across various

* Equal contribution

fields. In cases where collecting extensive data is impractical or costly, parameter-efficient models can provide reliable predictions while minimizing data requirements [43] [14]. This capability supports more effective planning in areas such as transportation, energy consumption, urban development, and healthcare, making forecasting solutions adaptable and scalable in rapidly evolving domains.

Accurately forecasting in spatial-temporal contexts presents several unique challenges, particularly when aiming for parameter efficiency with limited data. Traditional models often struggle to maintain accuracy when data is sparse or non-continual, as they are not designed to handle incomplete datasets effectively [8]. Furthermore, spatial dependencies—where conditions or usage patterns at one location can impact nearby locations—add another layer of complexity, requiring models that can capture these interdependencies dynamically [38]. Temporal variations, influenced by patterns or unexpected external factors, further complicate reliable forecasting [34]. Additionally, scalability is crucial: in applications with numerous spatial points, retraining a separate model for each location is computationally prohibitive. Addressing these spatial and temporal complexities while maintaining parameter efficiency across large networks necessitates advanced solutions capable of achieving high accuracy without extensive retraining.

This paper presents a novel approach by introducing the Bridged Attention Module (BAM) to build upon a transformer-based Spatio-Temporal Graph Convolutional Network (ST-GCN) for efficient transfer learning. The ST-GCN framework addresses both spatial and temporal aspects of charging station occupancy, with a GCN [20] modeling spatial relationships between stations and a transformer network [37] capturing temporal information. The proposed BAM adapter improves the adaptability and accuracy of this architecture in data-sparse scenarios, making it particularly effective for time series forecasting of EV charging station occupancy. A key innovation of this approach is the integration of the BAM adapter, which facilitates efficient transfer learning, allowing the model to be fine-tuned for different charging networks without requiring complete retraining. By significantly reducing the number of trainable parameters, BAM improves the model's ability to generalize across stations with limited data, enhancing scalability and computational efficiency. The combination of the ST-GCN and BAM modules provides a highly effective solution for forecasting charging station occupancy across charging station networks.

In summary, the main contributions of this paper are as follows:

- We introduce the BAM adapter, which enhances the performance of ST-GCN for transfer learning in time series forecasting under data-scarce conditions, while also significantly reducing the number of trainable parameters.
- We demonstrate the effectiveness of BAM in transfer learning scenarios by enabling the model to transfer knowledge from one EV charging station network to another. This is particularly beneficial for newly established charging station networks with limited historical data, making BAM an ideal solution for large-scale applications characterized by data-limited conditions.

2 Related Work

The rapid growth in EV adoption and demand for efficient charging infrastructure have brought significant attention to forecasting EV charging station occupancy. Various approaches address challenges in predicting usage patterns and optimizing management.

Handling missing data is essential for time series forecasting with intermittent datasets. Simple univariate methods like Last Observation Carried Forward (LOCF) and Next Observation Carried Backwards (NOCB) are effective for sparse series [12], while multivariate approaches, such as k-Nearest Neighbors (k-NN), improve imputation accuracy by leveraging relationships between variables, resulting in smoother imputed data [39][23].

Previous literature often used foundational ML models for time series forecasting, such as Moving Average (MA), Autoregressive (AR), and Autoregressive Integrated Moving Average (ARIMA) [33][31][17]. While extensively explored, these models primarily handle univariate data and linear relationships, limiting their effectiveness with complex patterns [35][7][6][41][25].

Unlike traditional ML methods, deep learning approaches such as Long Short-Term Memory (LSTM) networks and transformers excel with complex, high-dimensional, nonlinear data [13][37][21]. LSTM networks effectively capture long-term dependencies in time series data, while transformer models leverage attention mechanisms to process sequential data in parallel [29][26][32]. Techniques like fractional positional encoding further enhance transformers' ability to handle intermittent series with irregular intervals, making them popular for time series forecasting [11][2][1]. While LSTMs and transformers capture temporal patterns, they struggle with spatial dependencies; GCNs address this by modeling spatial relationships in graph-structured data [20][24]. ST-GCNs extend GCNs by incorporating temporal dynamics, enabling simultaneous modeling of spatial and temporal dependencies for tasks like traffic prediction [27][19][18]. These architectures have also been employed to predict EV charging station occupancy, utilizing the spatial distribution of stations alongside their temporal usage patterns [28][36].

Another used technique in time series forecasting is transfer learning. Transfer learning enables models to generalize effectively across diverse scenarios (e.g., urban networks) even when data is limited [40][3]. By enhancing both scalability and adaptability, transfer learning makes it feasible to handle large-scale scenarios. Some widely used transfer learning techniques include fine-tuning the last layer [42] or fine-tuning all layers [15]. Recently, the focus of transfer learning was more directed towards the integrations of small adapters modules into pre-trained models for task-specific fine-tuning while keeping the original weights frozen. This approach reduces the number of trainable parameters and improves the efficiency of the model [14][43]. Adapter-based techniques, like MLP-based modules, have been successfully applied to transformers and extended to CNNs and RNNs [5][22][10]. A further improvement is obtained with the Tiny-Attention Adapter, which enhances model performance by utilizing a compact multi-head attention mechanism to focus on key features [43]. However, it does

not capture spatial dependencies, as it was originally designed for large language models (LLMs) rather than spatio-temporal tasks. Additionally, the retraining of a task-specific decoder increases the number of trainable parameters. To address these limitations, this paper introduces BAM, which integrates GCN and transformer components within the ST-GCN model, effectively capturing both spatial and temporal dependencies while minimizing the number of trainable parameters.

3 Method

3.1 Background

To address the spatial and temporal dependencies in EV charging station occupancy data, this paper employs an ST-GCN model [27][19][18][28][36]. The ST-GCN combines a GCN to capture spatial relationships among charging stations with a flexible temporal modeling component. The GCN output is represented in our paper as

$$H_S = \mathrm{GCN}(X, G)$$

where $G = (V, E)$ represents the graph structure of the stations, and $X = \{x_t\}_{t=1}^{T}$ denotes the sequence of occupancy data over time, with each $x_t \in \mathbb{R}^{|V'| \times d}$ capturing the occupancy state of all stations at time t. In particular, the nodes V and edges E in G represent the set of charging stations and the distance between these stations, respectively. Each edge $e_{ij} \in E$ between nodes v_i and v_j is weighted according to the distance between the stations by road, enabling the ST-GCN to incorporate spatial relationships. For the temporal component, instead, various models can be integrated into the ST-GCN framework. This study specifically employs a transformer model which applies self-attention to the spatial encoding H_S [37], resulting in a spatio-temporal representation H_{ST} that allows the ST-GCN to account for both inter-station dependencies and temporal occupancy patterns.

Furthermore, this paper introduces the BAM adapter, a novel enhancement that improves the model's adaptability and performance of transfer learning with datasets containing limited labeled samples. To evaluate its performance, several transfer learning strategies and data percentages are considered during the evaluation. Each combination is represented as

$$\mathcal{T}(M, D_{\text{percentage}})$$

where M denotes the learning strategy, and $D_{\text{percentage}}$ specifies the fraction of target data used for fine-tuning.

3.2 Bridged Attention Module (BAM)

The BAM adapter's architecture is designed to maximize adaptability in data-sparse scenarios while keeping trainable parameters to a minimum. It achieves

this by incorporating both spatial and temporal information, thereby improving the flow of data from the GCN to the final predictions. It begins with a downscaling bottleneck layer that reduces input feature dimensionality, helping the model focus on essential information while significantly reducing parameter count and limiting overfitting risk. Next, a single-head attention mechanism captures the dependencies within the input sequence, offering sufficient expressiveness for inter-layer communication without the computational cost of multi-head attention, as used by the Tiny Attention Adapter [43]. The attention layer is formally expressed as $z_t = \text{Attention}(W_q x_t, W_k x_t, W_v x_t)$, where W_q, W_k, and W_v are learnable weight matrices for queries, keys, and values, respectively [37].

A dropout layer follows, providing regularization by randomly deactivating neurons to prevent overfitting, which is especially critical for cases with extremely limited data. The output from the attention mechanism is subject to dropout with probability p, and is represented as $\hat{z}_t = z_t \cdot \text{Bernoulli}(1-p)$. The features are then upscaled back to their original feature space and combined with a skip connection from the initial input, ensuring the preservation of the original information. This procedure is expressed as $y_t = \text{Concat}(\hat{z}_t, x_t)$ and it enables the model to maintain both the original context and the additional insights learned during adaptation. After that, Layer Normalization is applied to stabilize training, as in the following equation $\text{LayerNorm}(y_t) = \frac{y_t - \mu}{\sigma + \epsilon} \cdot \gamma + \beta$, where μ and σ are the mean and standard deviation of y_t, and γ and β are learnable parameters. Finally, the output is scaled by a learnable parameter and is expressed as $\text{Output} = \text{Scale}(\text{LayerNorm}(y_t))$.

The BAM adapter is positioned both between H_S and the transformer, and within the transformer's encoder-decoder layers. When placed between H_S and the transformer, it acts as a "bridge" that harmonizes diverse representations, ensuring smooth information flow across frozen layers, and integrating features learned by H_S with those within the transformer. Inside the transformer architecture, BAM is placed at the end of each encoder layer to adapt the information flow from the encoder to the decoder part based on the target domain. Indeed, the BAM adapter is the sole component that is fine-tuned to the target scenario, highlighting its adaptation capabilities and ensuring a low number of trainable parameters.

4 Result

4.1 Experiments

The experiments were conducted across two main scenarios, each designed to evaluate the model's ability to generalize across different networks. Three distinct EV charging-station networks, each representing small charging station areas of three different cities in Sweden—Värnamo, Varberg, and Malmö—were utilized to create the two transfer learning scenarios: Värnamo to Varberg and Varberg to Malmö. In each scenario, the model was trained on data from the source network using three distinct random seeds to ensure variability in initialization. Following this, each of these pre-trained models was fine-tuned on the target

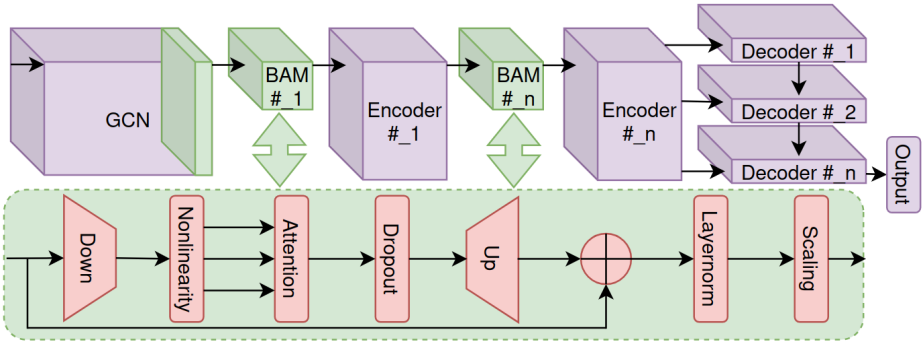

Fig. 1. Architecture of the proposed BAM adapter integrated into the ST-GCN. Green and red represent trainable weights, while purple indicates frozen weights.

network using three new seeds, creating a total of 9 unique seed combinations per scenario, allowing a comprehensive assessment of the model's robustness and performance consistency. The same experimental setup was repeated for the comparative study against baseline models (Sect. 4.2), and for the ablation study (Sect. 4.3) to analyze the contributions of individual components in the BAM adapter.

Although experiments were conducted on both scenarios (i.e., Värnamo to Varberg and Varberg to Malmö), only the results of the Värnamo to Varberg scenario are presented in this paper due to the page limit since the outcomes across both were consistent.

Data The data for this project was provided by ChargeFinder [4], and it consists of occupancy information from 29 EV charging stations across three Swedish cities: Värnamo, Varberg, and Malmö. Each dataset includes essential features: station ID, outlet count, occupied count, available count, offline count, and timestamps for each data point. As the data was logged intermittently based on user queries, an imputation method was applied to resample it into a continuous time series, making it suitable for time series forecasting.

To address this, the k-NN imputation method [30] was implemented for its capability to capture complex relationships among multiple features. By identifying patterns based on the similarity of data points in the temporal space, k-NN effectively fills in missing values.

Following the transformation from an intermittent to a continuous time series, the data was structured as a graph, denoted by $G = (V, E)$, as detailed in Section 3.1. For this study, a subset of the data consisting of five strategically selected stations was used for each city scenario, chosen based on their high usage and spatial distribution. These five stations together form a network of charging stations, denoted by $N = (V', E')$, where $V' \subset V$ and $E' \subset E$. This subset enables an efficient demonstration of the model's ability to leverage spatial relationships while managing computational complexity.

Table 1. Performance of various models at different data usage percentages (1%, 5%, 20%, and 50%) for the Värnamo to Varberg scenario. The "Parameters" column displays the percentage of trainable parameters within each model. All values are multiplied by 10^4.

Värnamo to Varberg					
Model	1%	5%	20%	50%	Parameters
No fine-tuning	126.4 ± 99.08	126.4 ± 99.08	126.4 ± 99.08	126.4 ± 99.08	0%
Traiend from scratch	122.49 ± 28.16	82.59 ± 14.1	55.75 ± 1.73	52.55 ± 2.68	100%
Fine-tune all	58.37 ± 5.27	54.45 ± 2.21	50.65 ± 1.26	50.2 ± 1.21	100%
Last GCN Layer	95.89 ± 49.74	68.25 ± 16.17	57.56 ± 5.50	56.94 ± 6.16	0.22%
Last transformer Layer	62.26 ± 1.05	57.26 ± 4.53	53.17 ± 2.80	53.49 ± 2.74	20.99%
MLP Adapter	71.53 ± 10.70	58.03 ± 3.88	52.44 ± 2.01	51.65 ± 1.53	2.69%
Tiny-Attention Adapter	63.07 ± 5.39	58.24 ± 3.22	52.49 ± 1.59	51.43 ± 2.78	57.01%
BAM	63.14 ± 7.63	53.7 ± 1.81	51.22 ± 1.39	50.09 ± 1.31	3.02%

4.2 Comparative Study

The results of the comparative study, presented in Table 1, highlight the strong performance of the BAM adapter. BAM's performance, with different percentage of data used, is comparable, or even outperform (in $\mathcal{T}(\text{BAM}, D_{5\%})$ and $\mathcal{T}(\text{BAM}, D_{50\%})$) the *fine-tune all* baseline, which require fine-tuning all model parameters. BAM on the other hand is very parameter-efficient, as reported in the last column of the table.

It is also worth noting the high standard deviation observed in several models, particularly at 1% data usage in Table 1. Models such as $\mathcal{T}(\text{No fine-tuning}, D_{1-50\%})$, $\mathcal{T}(\text{Trained from scratch}, D_{1\%})$, and $\mathcal{T}(\text{Last GCN layer}, D_{1\%})$ exhibit significant variability, also due to the scaling factor introduced for results presentation (10^4), which further amplify relative fluctuations. Furthermore, the low data regime causes overfitting and poor generalization, as demonstrated by the performance of $\mathcal{T}(\text{Trained from scratch}, D_{1-5\%})$. Also fine-tuning only the last GCN layer is insufficient for stable performance, especially with limited data (see $\mathcal{T}(\text{Last GCN layer}, D_{1-50\%})$). In contrast, $\mathcal{T}(\text{BAM}, D_{1-50\%})$ maintains consistent performance with low variability, showcasing its adaptability across different data percentages.

To qualitatively assess the efficiency of the BAM adapter compared to previous baselines in different data regimes, Figure 2 shows a Pareto plot with the MSE and the number of parameters trained (or fine/tuned) when transferring knowledge among different stations. The performance of the BAM adapter (blue color) outstands when considering its efficiency in utilizing fewer trainable parameters compared to other models. The only comparable baselines in terms of parameter count are the MLP adapter, which operates in a similar parameter range, and the fine-tuning of the last GCN layer, which uses even fewer parameters. However, when considering together both the MSE and the number of parameters, the BAM adapter outperforms all other methods when using 5% and 50% of the data. Notably, BAM achieves these results with only 3.02% of trainable parameters, compared to 100%, 100%, 20.99%, and 57.01% used by Trained from scratch, Fine-tune All, Fine-tune Last transformer, and

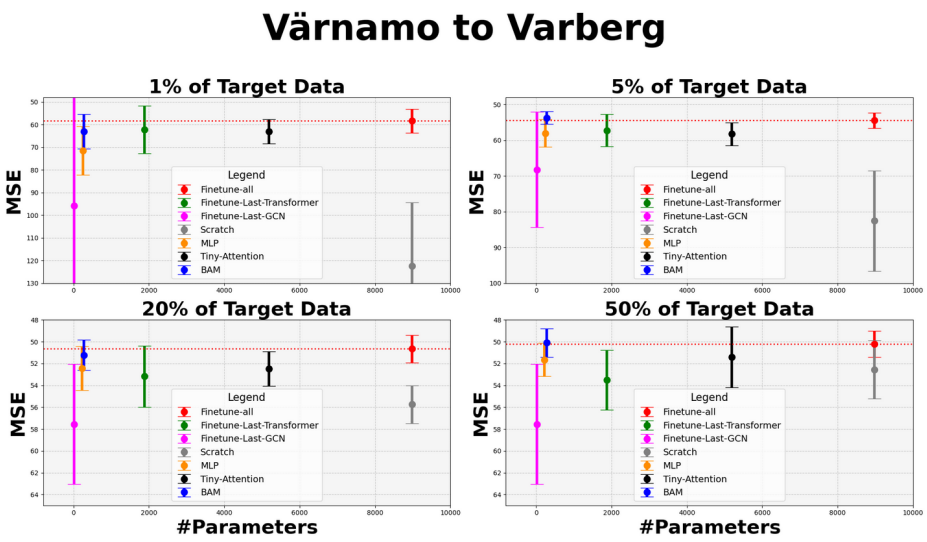

Fig. 2. MSE scores for different models at 1%, 5%, 20%, and 50% data usage, displayed across four subplots as a Pareto plot. Each subplot plots the number of trainable parameters on the X-axis and the MSE values on the Y-axis, with error bars representing the standard deviations. Results are shown for the Värnamo to Varberg scenario. All MSE values are multiplied by 10^4.

Tiny-Attention, respectively. This analysis emphasizes the BAM adapter's ability to maintain strong performance while using a minimal number of trainable parameters, demonstrating the model's efficiency and effectiveness in leveraging limited data and hardware resources to obtain competitive results.

4.3 Ablation Study

The ablation study was conducted to assess the impact of each component of the BAM adapter. By systematically removing or altering individual elements, the study identifies which components are essential for optimal performance. The results, presented in Table 2, show that each component is a necessary contribution to the BAM architecture, and its removal would lead to an increase in MSE.

The only exception is with $\mathcal{T}(\text{BAM}, D_{1\%})$. This might be attributed to the data-hungry nature of attention mechanisms, which may overfit and amplify irrelevant features at lower data percentages. Additionally, the improved performance when removing the layer normalization could be due to the fact that, with limited data, the model may benefit from the instability of training, which allows for occasional "lucky jumps" in optimization that lead to better performance. Similarly, removing the ST-GCN bridge reduces the model's complexity, which can be advantageous when working with very limited data, as it simplifies

Table 2. Ablation study on this study's proposed BAM for the Värnamo to Varberg scenario. Each row of the table reports the results when one component is selectively removed from the original BAM architecture. All values are multiplied by 10^4.

Ablation study (Värnamo to Varberg)				
Model	1%	5%	20%	50%
BAM	63.14 ± 7.63	53.7 ± 1.81	51.22 ± 1.39	50.09 ± 1.31
- GCN Fine-tuning	65, 28 ± 9, 98	55, 46 ± 1, 90	52, 04 ± 1, 43	50, 51 ± 1, 25
- ST-GCN Bridge	55, 39 ± 3, 86	54, 53 ± 1, 91	53, 23 ± 1, 10	51, 77 ± 1, 63
- Transformer Bridge	67, 03± 7, 12	57, 67 ± 2, 84	53, 83 ± 2, 97	52, 41 ± 2, 16
- Attention	61, 55 ± 7, 33	54, 42 ± 2, 01	52, 54 ± 1, 33	51, 18 ± 1, 44
- Dropout	64, 77 ± 7, 54	54, 88 ± 2.05	51, 30 ± 1, 44	50, 53 ± 1, 39
- Layer normalization	61, 47 ± 9, 36	54, 80 ± 2, 17	52, 79 ± 1, 36	51, 46 ± 1, 62
- Scalar	65, 82± 9, 37	55, 45 ± 1, 93	52, 35 ± 1, 22	50, 21 ± 2, 34

the feature transformations and reduces the risk of overfitting, thus improving performance at 1%.

$\mathcal{T}$(-Transformer Bridge, $D_{1-50\%}$), which removes the bridges inside the transformer, exhibits the greatest performance loss overall. Additionally, $\mathcal{T}$(-Scalar, $D_{1-50\%}$) demonstrated that the scalar component has a high parameter efficiency, offering notable performance improvements with minimal additional parameters.

These findings highlight how different components contribute to the overall effectiveness of the BAM adapter and assure high performance in a low data regime with a low number of trainable parameters, which is crucial in many industrial applications.

5 Conclusion

This paper introduces the BAM adapter, a novel and highly effective solution for overcoming the challenges associated with data-sparse time-series prediction tasks. By enhancing the performance of ST-GCN through efficient transfer learning, BAM demonstrates a unique ability to capture both spatial and temporal dependencies while maintaining exceptional parameter efficiency. The results highlight the scalability and efficiency of the BAM adapter, which, together with an ST-GCN model, outperforms or matches the performance of previous baselines while significantly reducing the number of trainable parameters. Extensive experiments were conducted on various EV charging station networks in Sweden, demonstrating that BAM offers an efficient transfer learning solution when new charging stations are introduced with only a few days of historical data available. This adaptability underscores BAM's potential to address critical forecasting needs across a wide range of domains.

Looking ahead, future work is expected to focus on extending the BAM adapter's application to larger and more diverse datasets, providing a deeper evaluation of its scalability and robustness across varied contexts. Furthermore, investigating BAM's potential in a wider array of domains, such as healthcare,

energy management, and urban planning, could uncover valuable opportunities to apply its parameter-efficient design to domain-specific forecasting challenges. Finally, integrating BAM with other advanced spatio-temporal architectures holds promise for unlocking additional performance improvements and expanding its applicability to more complex and demanding scenarios.

Acknowledgment: The work was carried out with support from The Knowledge Foundation and from Vinnova (Sweden's innovation agency) through the Vehicle Strategic Research and Innovation Programme, FFI.

Disclosure of Interests: The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Akita, R., Yoshihara, A., Matsubara, T., Uehara, K.: Deep learning for stock prediction using numerical and textual information. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). pp. 1–6. IEEE (2016)
2. Boulanger-Lewandowski, N., Bengio, Y., Vincent, P.: Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In: Proceedings of the 29th International Conference on Machine Learning (ICML). pp. 1159–1166 (2012)
3. Caruana, R., Silver, D.L., Baxter, J., Mitchell, T.M., Pratt, L.Y., Thrun, S.: Learning to learn: knowledge consolidation and transfer in inductive systems (1995)
4. ChargeFinder: Data Provided for this project. https://chargefinder.com/se
5. Chen, H., Tao, R., Zhang, H., Wang, Y., Li, X., Ye, W., Wang, J., Hu, G., Savvides, M.: Conv-adapter: Exploring parameter efficient transfer learning for convnets. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 1551–1561 (June 2024)
6. Clark, S., Rodriguez, P.: Limitations of autoregressive models for nonlinear time series data. Journal of Applied Statistics **42**(3), 341–355 (2019)
7. Clark, S., Rodriguez, P.: Challenges in modeling nonlinear patterns in time series data. Journal of Applied Statistics **41**(9), 1230–1245 (2021)
8. Dou, B., Zhu, Z., Merkurjev, E., Ke, L., Chen, L., Jiang, J., Zhu, Y., Liu, J., Zhang, B., Wei, G.W.: Machine learning methods for small data challenges in molecular science. Chemical Reviews **123**(13), 8736–8780 (2023). https://doi.org/10.1021/acs.chemrev.3c00189, https://doi.org/10.1021/acs.chemrev.3c00189, pMID: 37384816
9. European Union: The European Green Deal (11 December 2019), https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal_en, Accessed on 2023-11-24
10. Guo, D., Rush, A., Kim, Y.: Parameter-efficient transfer learning with diff pruning. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4884–4896. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.378, https://aclanthology.org/2021.acl-long.378

11. Harzig, P., Einfalt, M., Lienhart, R.: Synchronized audio-visual frames with fractional positional encoding for transformers in video-to-text translation. In: 2022 IEEE International Conference on Image Processing (ICIP). pp. 2041–2045. IEEE (2022)
12. Hassani, H., Kalantari, M., Ghodsi, Z.: Evaluating the performance of multiple imputation methods for handling missing values in time series data: A study focused on east africa, soil-carbonate-stable isotope data. Stats **2**(4), 457–467 (Dec 2019)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
14. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp (2019), https://arxiv.org/abs/1902.00751
15. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 (2018)
16. IEA: Electric Vehicles (2023), https://www.iea.org/energy-system/transport/electric-vehicles, Accessed on 2023-11-24
17. Johnson, K., Lee, M.: The use of arima models in time series forecasting. Journal of Forecasting **40**(4), 401–420 (2021)
18. Johnson, M., Wang, L.: Applications of st-gcns in urban planning and transportation. Journal of Urban Mobility **25**(3), 101–115 (2022)
19. Kim, A., Lee, S.: Advantages of spatio-temporal graph convolutional networks in forecasting. Journal of Machine Learning Research **34**(4), 345–359 (2021)
20. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
21. Lara-Benítez, P., Gallego-Ledesma, L., Carranza-García, M., Luna-Romera, J.M.: Evaluation of the transformer architecture for univariate time series forecasting. In: Alba, E., Luque, G., Chicano, F., Cotta, C., Camacho, D., Ojeda-Aciego, M., Montes, S., Troncoso, A., Riquelme, J., Gil-Merino, R. (eds.) Advances in Artificial Intelligence. pp. 106–115. Springer International Publishing, Cham (2021)
22. Le, H., Pino, J., Wang, C., Gu, J., Schwab, D., Besacier, L.: Lightweight adapter tuning for multilingual speech translation. arXiv preprint arXiv:2106.01463 (2021)
23. Lee, B., Lee, H., Ahn, H.: Improving load forecasting of electric vehicle charging stations through missing data imputation. Energies **13**(18), 4893 (Sep 2020)
24. Lee, J., Zhang, M.: Advantages of graph convolutional networks in capturing local neighborhood information. Journal of Graph Data Analysis **11**(3), 123–136 (2018)
25. Lee, M., Johnson, K.: Limitations of moving average models in time series prediction. Journal of Machine Learning **50**(4), 567–580 (2020)
26. Leeraksakiat, P., Pora, W.: Occupancy forecasting using LSTM neural networkand transfer learning. In: 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE (Jun 2020)
27. Lovanshi, M., Tiwari, V.: Human skeleton pose and spatio-temporal feature-based activity recognition using st-gcn. Multimedia Tools and Applications **83**(5), 12705–12730 (2024)
28. Luo, R., Song, Y., Huang, L., Zhang, Y., Su, R.: AST-GIN: Attribute-Augmented spatiotemporal graph informer network for electric vehicle charging station availability forecasting. Sensors (Basel) **23**(4) (Feb 2023)
29. Ma, T.Y., Faye, S.: Multistep electric vehicle charging station occupancy prediction using hybrid lstm neural networks. Energy **244**, 123217 (2022). https://doi.org/https://doi.org/10.1016/j.energy.2022.123217, https://www.sciencedirect.com/science/article/pii/S0360544222001207

30. Murti, D.M.P., Pujianto, U., Wibawa, A.P., Akbar, M.I.: K-nearest neighbor (k-nn) based missing data imputation. In: 2019 5th International Conference on Science in Information Technology (ICSITech). pp. 83–88 (2019). https://doi.org/10.1109/ICSITech46713.2019.8987530
31. Nie, S.y., Wu, X.q.: A historical study about the developing process of the classical linear time series models. In: Yin, Z., Pan, L., Fang, X. (eds.) Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 2013. pp. 425–433. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
32. Parikh, A.P., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933 (2016)
33. S., H.L.: A study in the analysis of stationary time series. by herman wold. [pp. 214 + viii. almqvist and wiksells boktryckeri-a.-b., uppsala. 1938. price kr. 6.]. Journal of the Institute of Actuaries **70**(1), 113–115 (1939). https://doi.org/10.1017/S0020268100011574
34. Saxena, D., Cao, J.: D-gan: Deep generative adversarial nets for spatio-temporal prediction (2021), https://arxiv.org/abs/1907.08556
35. Smith, R., Green, A.: Strengths of autoregressive models in time series analysis. Journal of Statistical Modeling **30**(1), 45–57 (2018)
36. Su, S., Li, Y., Chen, Q., Xia, M., Yamashita, K., Jurasz, J.: Operating status prediction model at EV charging stations with fusing spatiotemporal graph convolutional network. IEEE Trans. Transp. Electrif. pp. 1–1 (2022)
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems (2017)
38. Vyas, A., Bandyopadhyay, S.: Dynamic structure learning through graph neural network for forecasting soil moisture in precision agriculture (2022), https://arxiv.org/abs/2012.03506
39. Wafaa Mustafa Hameed, N.A.A.: Comparison of seventeen missing value imputation techniques. Journal of Hunan University Natural Sciences **49**(7) (2022)
40. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. Journal of Big data **3**, 1–40 (2016)
41. White, L., Black, G.: Challenges and limitations of arima models in time series prediction. Journal of Machine Learning **51**(5), 601–615 (2022)
42. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? (2014), https://arxiv.org/abs/1411.1792
43. Zhao, H., Tan, H., Mei, H.: Tiny-attention adapter: Contexts are more important than the number of parameters. arXiv preprint arXiv:2211.01979 (2022)