
Causal Graph Neural Networks for Wildfire Prediction

Final Report (Group 23)

Wenting Yue
wyue2@andrew.cmu.edu

Wenyu Liu
wenyuliu@andrew.cmu.edu

Youyou Huang
youyouh@andrew.cmu.edu

Abstract

Wildfires pose significant threats to our ecological sustainability and socioeconomic stability throughout the world. The increasing frequency and intensity of these events, mainly driven by climate change, underscores the urgent need for more reliable predictive technologies to strengthen the response capacity to disasters and mitigation of communities [3]. In this report, we reimplement the approach proposed in *Causal Graph Neural Networks for Wildfire Danger Prediction* [12], which introduces a causal GNN framework to model spatiotemporal dependencies in the prediction of wildfires. Building on the SeasFire dataset, we replicate the original architecture and experimental setup, evaluate its performance on held-out seasons, and explore extensions to better understand causal drivers. The report includes a review of the literature on deep learning work related to wildfires, architectural details of the reimplementation, evaluation strategies, and future directions.

1 Introduction

Recent advances in artificial intelligence and deep learning have revolutionized our ability to model complex environmental systems, often outperforming traditional empirical or physics-based approaches. Deep neural networks excel at extracting subtle, high-dimensional patterns from diverse inputs, making them increasingly popular in Earth system and climate research. Wildfire prediction in particular stands to benefit, as climate change drives ever more frequent and severe fire events.

In this project, we reimplement the causal graph neural network framework of Zhao et al. [12], which unites causal discovery with spatiotemporal modeling to forecast seasonal wildfire danger. Using the global SeasFire datacube, a rich fusion of fire observations, meteorological variables, and climate indices, we not only reproduce baseline results, but also provide the missing details of our data processing pipeline and model implementation. By doing so, we enhance both the interpretability and robustness of the original approach.

Finally, we confront one of the field’s greatest challenges: data sparsity and scarcity. Limited and narrowly specialized observations in many regions impede the generalizability of deep learning-based wildfire-prediction frameworks, and overcoming this obstacle will be critical to deploying reliable forecasting tools worldwide.

2 Literature Review

2.1 Traditional Approach

Traditional research in wildfire prediction has relied heavily on statistical and classical machine learning (ML) techniques, including AdaBoost, Random Forests, and Support Vector Machines (SVMs). Although these models have shown effectiveness in handling localized and relatively simple

prediction tasks, they often struggle to capture complex spatio-temporal dependencies and interactions among environmental variables. Moreover, they typically require high-quality, well-curated input data and rely on extensive manual feature engineering, which limits their scalability and generalizability across diverse geographic regions and extended forecast horizons [1]. However, these conventional approaches offer valuable insights, particularly in identifying key predictive variables, and often provide greater interpretability and transparency compared to deep learning (DL) models, which are frequently criticized as "black-box" systems. Such interpretability serves as a critical reference point for informing and evaluating the structure and behavior of more complex DL-based models.

2.2 Deep Learning Approach

Recent studies highlight the superiority of DL methods in predicting wildfires, particularly convolutional neural networks (CNNs), convolutional recurrent networks (CRNs) and transformers. CNNs and its variants like U-Net are highly effective in extracting spatial features from images, while recurrent networks, such as LSTMs and GRUs, can capture temporal dependencies and patterns essential for model the dynamics of wildfires [7] [9]. Models integrating CNNs with recurrent modules, such as ConvLSTM, achieve notable performance in spatio-temporal data processing and become one of the predominant architectures used in this field [1].

Furthermore, the incorporation of attention mechanisms and reward systems have improved model precision and learning over a longer horizon [1] [9] [11]. Graph neural networks (GNNs), including STGNNs and T-ZCN, have provided alternative approaches to reconstruct textual and numerical data in a graphic representation, for a more granular extraction of features and dependencies between variables [10] [2].

The researchers also proposed techniques to address challenges in harmonizing noisy inputs and enhancing model performance. Marjani, Ahmadi, and Mahdianpari suggested a strategy that separates data based on temporal resolutions (e.g., hourly, daily, and static) into distinct parallel processing blocks, which are later combined into a unified dense layer. This method aims to mitigate issues related to data duplication, information redundancy, and premature dominance of particular data channels [8]. Additionally, some studies have integrated interpretability and explainability techniques, such as SHapley Additive exPlanation (SHAP) and Gradient-weighted Class Activation Mapping (Grad-CAM). These approaches facilitate post hoc analyzes, allowing researchers to evaluate the impact of individual variables on model predictions and identify potential areas for improvement [1] [6].

Despite the advantages of DL models, they continue to face challenges related to data quality, input harmonization, generalizability, and computational resource constraints. The resolution disparities and noise inherent in satellite imagery and weather data create pre-processing burdens, often compromising model performance and deployment [4]. Furthermore, many studies focus on a narrow geographical specification, probably due to data scarcity, which limits the transferability of the model as a trade-off for a higher accuracy rate, highlighting the importance of transfer learning or domain adaptation [1].

2.3 Baseline Selection

Among recent advances in wildfire prediction, the framework proposed by Zhao et al. [12] presents a novel integration of causal discovery with graph-based spatiotemporal modeling. The model employs the PCMCi (Peter-Clark Momentary Conditional Independence) algorithm to derive a causally informed adjacency matrix from climate and fire variables, which is subsequently used to guide a graph neural network (GNN) for binary wildfire danger classification.

We selected this work for reproduction due to its methodological innovation and its potential to enhance interpretability and generalizability in long-range wildfire forecasting. The authors' use of the SeasFire datacube, which harmonizes local meteorological measurements with global Oceanic Climate Indices (OCIs), aligns well with our objective of incorporating both regional and teleconnected climatic drivers. Through reimplementing, we aim to validate the original findings, evaluate robustness across spatio-temporal splits, and examine the utility of causal GNNs under varying biomes and pre-processing configurations.

3 Data Preparation

3.1 Data Source Overview

The SeasFire dataset is a publicly available spatio-temporal datacube designed for global, seasonal wildfire forecasting. Developed by Karasante et al. [5], the data set spans from 2001 to 2021 and offers harmonized environmental and climatic variables at a temporal resolution of 8 days and a spatial resolution of 0.25 degree. The data is structured as a three-dimensional array with dimensions $time \times latitude \times longitude$, where each voxel contains over 30 geophysical variables derived from multiple sources.

The dataset integrates local meteorological variables from ERA5-Land reanalysis data (e.g., temperature at 2m, total precipitation, and vapor pressure deficit) with broader climate drivers such as Oceanic Climate Indices (OCIs), including the North Atlantic Oscillation (NAO) and Niño 3.4 anomaly. The target variable, burned area, is sourced from the Global Wildfire Information System (GWIS), which compiles MODIS-based fire products to provide ground-truth fire occurrence information at the same spatiotemporal resolution.

In addition to global coverage, the dataset also includes biome-level masks, enabling targeted forecasting tasks in specific ecological zones. In our work, we focus on two fire-prone biomes: the Mediterranean and Boreal zones, following the experimental setup in Zhao et al. [12]. These biomes exhibit distinct fire regimes and climatic influences, making them well-suited for evaluating the generalizability of causal models across varying geospatial contexts.

3.2 Data Processing Pipeline

The preprocessing pipeline employs a carefully structured filtering and sampling methodology to extract meaningful data instances from a large-scale spatiotemporal Earth observation dataset. The strategy is grounded in domain-specific knowledge, focusing on biomes particularly prone to wildfires, while adhering to the temporal and spatial constraints essential for training predictive models.

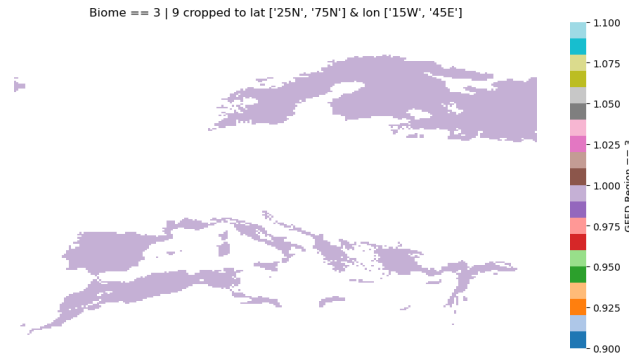
3.2.1 Biome and Geographic Filtering

To narrow down the spatial scope of analysis, the data is filtered based on:

- **Biome classification:** Only observations from specific biome classes (Europe’s Mediterranean and Boreal biomes) are retained. This mask is applied across the spatial grid using the biome variable from the dataset.
- **Latitude and Longitude Bounds:** The data is further cropped to a specified geographic window (between 25°N–75°N latitude and 15°W–45°E longitude) using coordinate-to-index conversion. This ensures computational tractability and region-specific learning.

The resulting subset is a spatial mask applied to all time steps of the dataset.

Figure 1: Sampled Region



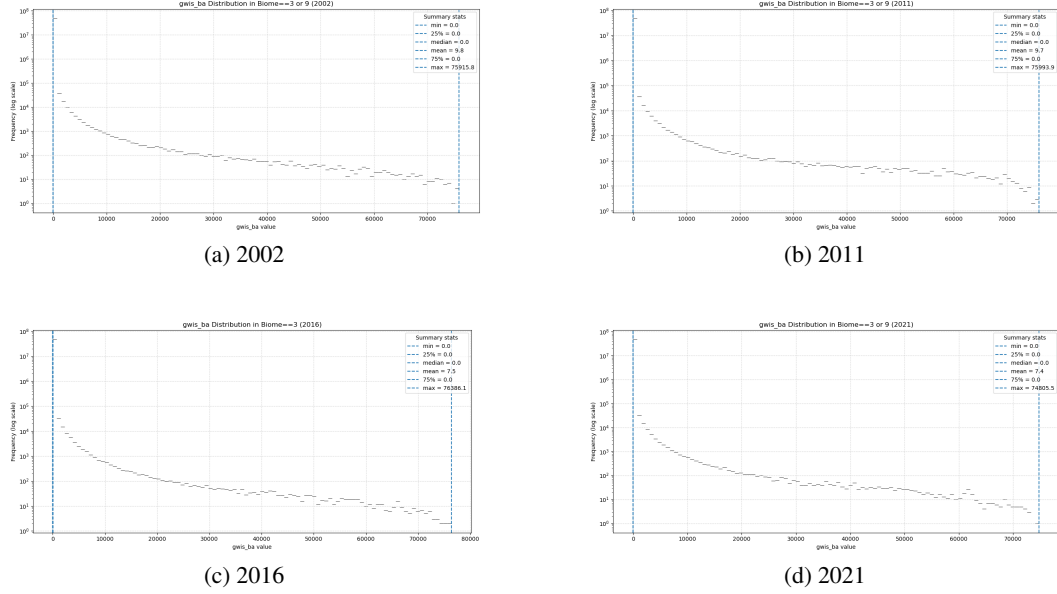
Note: Figure illustrates the regions with Europe’s Mediterranean and Boreal biomes.

3.2.2 Lack of Sampling Clarity in Baseline

According to Zhao et al. [12], each data sample is defined as a 39-step time series, where each step corresponds to an 8-day interval, accompanied by a binary “target” label indicating the presence or absence of wildfire activity. However, the original paper omits key implementation details regarding the precise definition of the “target” and the methodology used for sampling instances.

In our reimplementation, we infer wildfire occurrence using the variable "GWIS_BA" from the SeasFire dataset, a continuous indicator of burned area aggregated at a spatial resolution of 0.25 degrees. A non-zero value in this variable signifies the occurrence of fire within the corresponding spatial-temporal cell.

Figure 2: GWIS_BA Distribution By Year



Note: Graph shows distribution of the reported GWIS_BA (burned area) by in 4 years. The overall distribution stays the same over time and is heavily left-skewed, meaning that there are significantly more observations with no or low-level fire occurrence in the dataset.

A naive approach to labeling the “target” as the fire status at a given time step and using the preceding 39 timesteps as input could lead to two major issues:

- **Temporal overlap between samples:** Repeated use of overlapping windows around the same event may introduce bias or lead to data leakage during training.
- **Contamination of predictive history:** If fire activity is present in the input sequence leading up to a labeled fire event, the model may learn to recognize ongoing fires rather than predicting initial occurrences.

To address these challenges while still generating a sufficient number of training instances, we adopt a more principled sampling strategy, detailed as follows:

3.2.3 Positive and Negative Sample Extraction

A binary classification task is formulated by:

- **Positive Samples:** A spatial-temporal point is considered "positive" if it exceeds a user-defined wildfire activity threshold ($\text{fire_threshold} = 5$) at time step t , and all values in the preceding time_lag steps are below a safety threshold ($\text{safe_threshold} = 1$). Based on our observed distribution of GWIS_BA values across years (shown in figure 2), we believe the proposed thresholds are appropriate to generate balanced samples. This criterion ensures the sample captures sudden fire emergence after a calm period.

- **Negative Samples:** These are drawn from regions and time points where no fire activity occurred for the entire time_lag duration. A specified negative_ratio (5:1) governs the number of negative examples per positive one.

This dichotomy forms the core of the supervised learning labels.

3.2.4 Temporal Overlap Constraint

To reduce data leakage and redundancy, a temporal overlap constraint is applied:

- A maximum overlap ratio (e.g., max_overlap = 0.3) restricts how many time steps any two samples may share for the same spatial location.
- A greedy filtering algorithm ensures each sample’s temporal window is sufficiently isolated from others in the same cell, enhancing training robustness and avoiding autocorrelation bias.

3.2.5 Rolling Window Aggregation and Feature Extraction

Each selected coordinate triple (t, y, x) is transformed into a JSON object containing:

- **Local Variables:** Multivariate time series features (e.g., temperature, vapor pressure deficit, precipitation) over the time_lag window.
- **Global Climate Indices (OCI variables):** Exogenous covariates that are either global or purely temporal (e.g., North Atlantic Oscillation, El Niño 3.4 anomaly), included to enrich model context.
- **Target Label:** A binary indicator (0 or 1) of fire occurrence at time t.

These preprocessed instances are saved in JSON format, facilitating modular usage across training, validation, and test modes.

3.2.6 Matrix Specific Dataset

Different from the datasets constructed for model training, the matrix dataset captures the entire fire activity time series from 2001 to 2019 for each spatial cell over a multiyear period. The design is structured as follows:

- **No Sampling:** All cells within the geographic and biome filter mask are retained, rather than sampling positive or negative cases.
- **Temporal Coverage:** Each spatial cell (latitude, longitude) is associated with a full sequence (e.g., 20 years of 8-day intervals) of fire activity labels.
- **Multi-label Target:** The target is a sequence of binary indicators for each timestep, where each value denotes whether a fire occurred at that time.

We believe that the separate matrix dataset format better supports the causal analysis of the PCMCI algorithm by preserving uninterrupted temporal continuity, in contrast to the segmented windows used in supervised datasets.

Table 1: Sample counts by partition

Partition	Year Coverage	Positive	Negative	Total
Train	2001-2017	7,016	33,728	40,744
Validation	2018-2019	488	2,378	2,866
Test	2020-2021	475	2,318	2,793
Matrix	2001-2019	–	–	7,623

4 Model Description

The implemented model consists of three important components: a **Temporal LSTM** that encodes each node’s univariate time series into a fixed-length embedding, and a **Graph Neural Network** that propagates and refines these embeddings according to a learned, **directed adjacency** structure.

4.1 Adjacency Matrix

To construct interpretable causal graphs for wildfire danger prediction, we leverage the PCMCI algorithm from the Tigramite causal discovery framework. This methodology aims to infer lagged causal dependencies among multivariate time-series variables extracted from the matrix-format dataset. The central algorithm is configured as follows:

- **Independence Test:** Partial correlation (ParCorr).
- **Maximum Lag:** $\text{tau_max} = 23$, allowing for the detection of dependencies over approximately 6 months ($23 * 8\text{-day intervals}$).
- **Significance Threshold:** $\text{pc_alpha} = 0.05$ filters out statistically insignificant edges.

After running PCMCI, we produce three matrices:

- **P-value matrix:** p-values for each potential lagged edge.
- **Coefficient matrix:** estimated causal strength values.
- **Link matrix:** Boolean mask of statistically significant edges ($p \leq \alpha$).

We synthesize a final causal adjacency matrix through the following steps:

- **Significance Filtering:** We use data from the coefficient matrix, masked by the link matrix, meaning that only significant values are selected.
- **Temporal Collapse:** The 3D matrix ($V, V, \text{tau_max}$) is collapsed along the lag dimension, aggregated by mean, to yield a static 2D adjacency matrix ($V * V$).
- **Normalization:** The resulting matrix is optionally row-normalized such that each row sums to 1, with self-loops removed (zero-ed diagonal). This facilitates its use in downstream GNNs as a more stable weighted message-passing matrix.
- **Target Masking:** Finally, the "target" row and column are removed from the matrix to prevent label leakage during training while preserving causal inference for feature variables.

The final output is a normalized, interpretable, and task-specific causal graph over observed features, capturing statistical dependencies among variables in the wildfire dataset. Due to the large computational burden to run PCMCI calculations on the full matrix sample, we randomly selected 10% of the observations to generate the adjacency matrix.

$$\mathbf{A} = [A_{ij}]_{i,j=1}^V \in \mathbb{R}^{V \times V} \quad (1)$$

Table 2: Adjacency Matrix

	AO	NAO	NINA34_ANOM	T2M_MEAN	TP	VPD
AO		0.479	0.041	0.045	-0.037	-0.039
NAO	0.425		0.051	0.008	-0.026	-0.029
NINA34_ANOM	-0.001	-0.065		0.093	-0.045	0.075
T2M_MEAN	0.076	-0.082	-0.034		-0.001	0.317
TP	-0.020	-0.049	-0.011	-0.037		-0.331
VPD	0.010	-0.016	-0.013	0.282	-0.247	

Note: Table shows the aggregated, normalized, masked adjacency matrix derived from PCMCI.

4.2 LSTM

A one-layer, unidirectional LSTM is used to summarize each node’s history into a single hidden state of dimension H (256). Given an input tensor $\mathbf{x}$:

$$\mathbf{x} \in \mathbb{R}^{B \times V \times T} \quad (2)$$

where B is batch size, V the number of variables (nodes), and T the temporal length. The LSTM will yield per-node embedding $\mathbf{e}$ ready for graph convolution.

$$\mathbf{e} = \text{LSTM}(\text{reshape}(\mathbf{x})) \in \mathbb{R}^{B \times V \times H} \quad (3)$$

4.3 GNN

The core spatiotemporal reasoning is performed by a two-layer DenseGCNConv network (from PyTorch Geometric), which takes the node embeddings and a fixed, normalized adjacency matrix $\mathbf{A}$.

Then, the weighted features are passed through the following hidden layers, each followed by *GraphNorm*.

$$h^{(1)} = \text{LeakyReLU}(\text{DenseGCNConv}(\mathbf{e}, \mathbf{A})) \in \mathbb{R}^{B \times V \times 2H} \quad (4)$$

$$h^{(2)} = \text{LeakyReLU}(\text{DenseGCNConv}(h^{(1)}, \mathbf{A})) \in \mathbb{R}^{B \times V \times H} \quad (5)$$

A global pooling and binary classification are applied at the end.

$$\bar{h} = \frac{1}{V} \sum_{v=1}^V h_{b,v,:}^{(2)} \in \mathbb{R}^H \quad (6)$$

$$\mathbf{z} = W \bar{h}_b + \mathbf{b} \in \mathbb{R}^2 \quad (7)$$

Softmax is applied to calculate the confidence about the prediction.

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{z}_b) \quad (8)$$

All learnable weights (in particular the final linear layer) are initialized by Xavier normal, and non-learnable structures (the adjacency matrix) are registered as buffers for flexible device placement.

Table 3: Model Summary of CausalGNN

Layer (type)	Output Shape	Param #
CausalGNN	[256, 2]	–
TemporalLSTM	[256, 6, 256]	–
LSTM	[1536, 256, 256]	265,216
DenseGCNConv (1)	[256, 6, 256]	256
Linear (1)	[256, 6, 256]	65,536
GraphNorm (1)	[256, 6, 256]	768
DenseGCNConv (2)	[256, 6, 256]	256
Linear (2)	[256, 6, 256]	65,536
GraphNorm (2)	[256, 6, 256]	768
AdaptiveAvgPool1d	[256, 256, 1]	–
Linear (output)	[256, 2]	514
Total Parameters		398,850

4.4 Evaluation Metrics

To assess model performance on the binary wildfire prediction task, we use two widely adopted classification metrics: **Area Under the Receiver Operating Characteristic Curve (AUROC)** and **Area Under the Precision-Recall Curve (AUPRC)**. These metrics are particularly suitable for imbalanced datasets, where accuracy may be misleading.

AUROC measures a model’s ability to distinguish between the positive and negative classes across all classification thresholds. In our case, it evaluates how well the model distinguishes between burned and unburned grid cells.

The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR), defined as:

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN}$$

AUROC is the area under this curve. A value of 1.0 indicates perfect discrimination, while 0.5 indicates performance no better than random guessing.

AUPRC focuses on the model’s ability to correctly predict the positive class (burned areas).

Precision and Recall are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

A higher AUPRC indicates that the model maintains both high precision and high recall, which is critical in wildfire forecasting where both false alarms and missed detections carry operational consequences.

Given the high class imbalance in wildfire data, both metrics offer more nuanced view of model performance than accuracy. AUROC evaluates the model’s overall ranking ability, while AUPRC emphasizes effectiveness on the positive class, making them complementary for this task.

4.5 Loss Function

We adopt the `CrossEntropyLoss` from PyTorch as the loss function. This loss is appropriate for multi-class classification problems and simplifies to binary cross-entropy in the case of two classes.

Given the predicted logits $\hat{\mathbf{y}} \in \mathbb{R}^2$ for each input and the corresponding ground-truth label $y \in \{0, 1\}$, the cross-entropy loss for a single sample is defined as:

$$\mathcal{L}_{\text{CE}}(\hat{\mathbf{y}}, y) = -\log \left(\frac{e^{\hat{y}_y}}{\sum_{j=0}^1 e^{\hat{y}_j}} \right)$$

Here, $\hat{y}_j$ is the predicted logit for class j , and the softmax function ensures the output to be a valid probability distribution over classes.

During training, the total loss is computed as the average cross-entropy across all samples in the batch.

Optimization

We optimize the loss using the Adam variant of gradient descent, which adaptively adjusts learning rates for each parameter based on running estimates of the first and second moments of past gradients. The parameter update rule at step t for parameter θ_t is:

$$\theta_{t+1} = \theta_t - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$$

where:

- $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected first and second moment estimates of the gradient,
- α is the learning rate,
- ϵ is a small constant for numerical stability.

We then apply a learning rate scheduler (`ReduceLROnPlateau`) to dynamically adjust the learning rate based on validation loss.

5 Results

5.1 Performance and Evaluation

Our re-implementation of the Causal Graph Neural Network (CGNN) demonstrates improved performance over the original baseline reported by Zhao et al. [12], with an AUROC of 93.3% (vs. 92.4%) and a significantly higher AUPRC of 73.1% (vs. 34.6%), as shown in Table 3. While these results are promising, they also warrant a careful examination of potential confounding factors and training dynamics.

Table 4: Figure 3: Model Performance

Model	AUPRC (%)	AUROC (%)
Random classifier	1.1	50.0
LSTM	32.9	91.9
GRU	34.6	91.6
GNN_CORR	30.5	92.1
GNN_FULL	29.1	90.8
GNN_CAUSAL (Baseline)	34.6	92.4
Baseline Reimplementation	71.3	92.5
No GraphNorm	73.1	93.3
Deeper & Wider	63.7	90.5

Note: Table shows comparison of model performance. Our reimplementation results are shown in the final row.

5.1.1 Volatile Validation Metrics Across Epochs

Although the training loss decreases steadily, the validation loss and AUROC fluctuate significantly across epochs, indicating potential problems in model generalization or stability. Below are some potential causes:

- **Small model:** The baseline architecture consists of one layer of LSTM and two layers of GCN with a hidden dimension of 256, making the model quite shallow and small. A potential increase in model layers and size could potentially lead to better performance through training. However, when increasing the model depth and hidden dimension, we observed more stable training outcomes but worse model performance (See Appendix). More ablations on exploring the most optimal model architecture and hyperparameters are necessary.
- **Overfitting:** The model may be overfitting to specific patterns in the training data as the validation loss starts to increase after 20+ epochs, especially if the causal graph overemphasizes short-term dependencies. Additional regularization mechanisms, such as dropouts, may help models to improve along with a deeper architecture.
- **Optimizer Dynamics:** Learning-rate schedules or batch-wise stochasticity could be causing noisy updates, particularly in the early or late stages of training. Additional ablations on adjusting the hyperparameters could help determine the optimal training strategy for our model.

5.1.2 Performance Comparison to Baseline

Here are the potential factors that contribute to performance gains:

Data Splitting Strategy As the original author did not disclose the details regarding their sampling methods, it is difficult to conclude whether our model will achieve the same performance in samples from the original authors. However, our sampling strategy targets a specific task, that is, predicting the danger of wildfires in a region with little to no fire occurrence in the past 10 months. We believe that such a scenario should be emphasized because it may have a higher likelihood of human negligence

than regions with recent wildfire events and active mitigations. We carefully balanced the ratio of positive and negative samples to avoid our model being overly rewarded for negative predictions. In addition, our definition of positive and negative samples may differ from that of the original author, causing the models to perform different tasks.

Adjacency Matrix Variations We generate the causal adjacency matrix independently using the PCMCI algorithm on a subset of our preprocessed matrix dataset. Differences in variable normalization, masking strategies, or significance thresholds may result in a structurally distinct causal graph, which could provide a better inductive bias for the GNN’s message passing.

5.2 Future Extensions

Advancing wildfire danger prediction requires addressing both methodological and data-driven challenges that constrain current machine learning paradigms. A promising future direction for our baseline reimplementation involves the integration of attention mechanisms, which could enhance a model’s capacity to dynamically prioritize critical variables and timesteps, particularly beneficial in highly non-stationary environments like wildfire-prone regions. Furthermore, leveraging multimodal data input, such as satellite imagery, topographic maps, and socio-economic indicators, can enrich the feature space beyond climate and vegetation indices, allowing for more comprehensive modeling of ignition risk and fire spread. However, one of the most persistent limitations in the field is data scarcity and heterogeneity across regions. Most existing studies are geographically constrained and rely on region-specific datasets, making it difficult to generalize DL models to broader contexts. The lack of high-resolution, standardized datasets across temporal and spatial scales continues to hinder robust benchmarking and global model applicability.

Although DL approaches offer superior representation learning and predictive capacity, they often inherit the same limitations that have long challenged traditional ML methods, such as sensitivity to feature noise and reliance on balanced training data, and even more limited interpretability. Our baseline framework, which blends DL modeling with a causally informed matrix construction phase, helps to increase interpretability by offering a more interpretable structure. The causal graph generated from PCMCI allows researchers to examine temporal and inter-variable dependencies a priori, offering transparency into the data’s underlying structure before it is passed to the neural network. This hybrid approach exemplifies how deep learning and classical statistical insights can be integrated, producing models that are both performant and scientifically interpretable. As wildfire risks grow in frequency and intensity, future work must aim for scalable, explainable, and cross-regionally transferable models, supported by investments in open, multimodal environmental data platforms.

6 Conclusion

In this study, we re-implemented and evaluated the Causal Graph Neural Network (CGNN) model for wildfire danger prediction, demonstrating improved performance over previously reported baselines. By integrating causal discovery with graph-based deep learning, the model leverages both statistical rigor and neural representation learning to capture complex dependencies among environmental and climatic variables. Although our model shows quantitatively superior performance, particularly in AUPRC, a metric sensitive to class imbalance, the volatility in validation metrics and possible differences in experimental setup highlight the need for rigorous sampling and evaluation standards. Continued experimentation and sensitivity analysis are necessary to fully comprehend the gains over baseline performance. Our findings also suggest that data partitioning strategies and sampling assumptions can substantially influence evaluation results, emphasizing the need for standardized benchmarks and validation protocols in wildfire modeling.

7 Team Contributions

Collectively contributed to all aspects of researching, dataset collection, model construction, and report drafting.

8 Github Repository

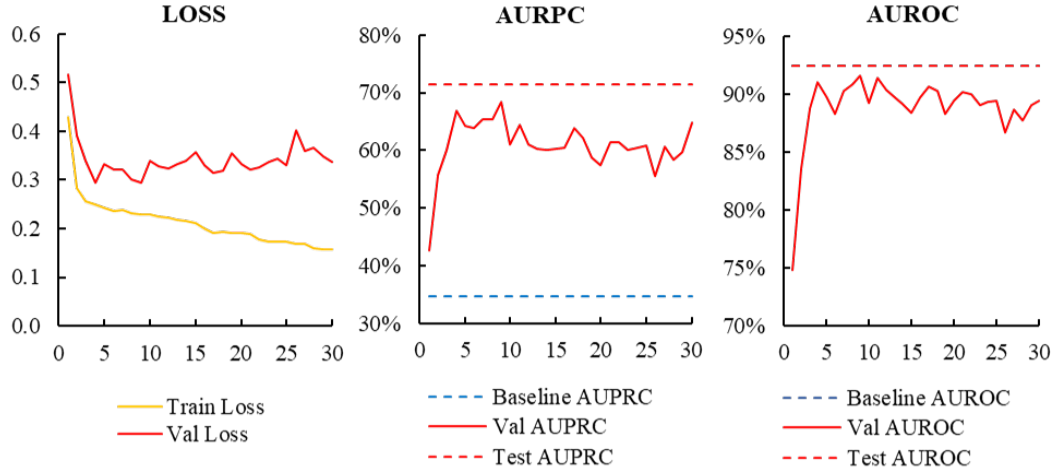
https://github.com/youyouh511/11785_IDL_S25_Final-Project/tree/main

References

- [1] Henintsoa S. Andrianarivony and Moulay A. Akhloufi. Machine learning and deep learning for wildfire spread prediction: A review. *Fire*, 7(12):482, 2024.
- [2] Yuzhou Chen, Joel Chacon Castillo, Huikyo Lee, and Yulia R. Gel. Firecast zigzag convolutional network for wildfire prediction. In *IEEE International Conference on Big Data (BigData)*, pages 1943–1949, 2024.
- [3] Calum Cunningham, Grant J. Williamson, and David M. J. S. Bowman. Increasing frequency and intensity of the most extreme wildfires on earth. *Nature Ecology & Evolution*, 8(8):1420–1425, 2024.
- [4] Rafik Ghali and Moulay A. Akhloufi. Deep learning approaches for wildland fires using satellite remote sensing data: Detection, mapping, and prediction. *Fire*, 6(5):192, 2023.
- [5] Ilektra Karasante, Lazaro Alonso, Ioannis Prapas, Akanksha Ahuja, Nuno Carvalhais, and Ioannis Papoutsis. Seasfire as a multivariate earth system datacube for wildfire dynamics. *arXiv preprint arXiv:2312.07199*, 2023.
- [6] M. Marjani and M. S. Mesgari. The large-scale wildfire spread prediction using a multi-kernel convolutional neural network. *Ecological Informatics*, 78:102282, 2023.
- [7] Mohammad Marjani, Seyed Ali Ahmadi, and Masoud Mahdianpari. Firepred: A hybrid multi-temporal convolutional neural network model for wildfire spread prediction. *Ecological Informatics*, 78:102282, 2023.
- [8] Mohammad Marjani, Masoud Mahdianpari, and Fariba Mohammadimanesh. Application of explainable artificial intelligence in predicting wildfire spread: An aspp-enabled cnn approach. *IEEE Geoscience and Remote Sensing Letters*, 21:1–5, 2024.
- [9] Arif Masrur, Manzhu Yu, and Alan Taylor. Capturing and interpreting wildfire spread dynamics: Attention-based spatiotemporal models using convlstm networks. *Ecological Informatics*, 82:102760, 2024.
- [10] Moritz Rösch, Michael Nolde, Tobias Ullmann, and Torsten Riedlinger. Data-driven wildfire spread modeling of european wildfires using a spatiotemporal graph neural network. *Fire*, 7:207, 2024.
- [11] Zeinab Shirvani, Omid Abdi, and Rosa C. Goodman. High-resolution semantic segmentation of woodland fires using residual attention unet and time series of sentinel-2. *Remote Sensing*, 15(5):1342, 2023.
- [12] Shan Zhao, Ioannis Prapas, Ilektra Karasante, Zhitong Xiong, Ioannis Papoutsis, Gustau Camps-Valls, and Xiao Xiang Zhu. Causal graph neural networks for wildfire danger prediction. *arXiv preprint arXiv:2403.08414*, 2024.

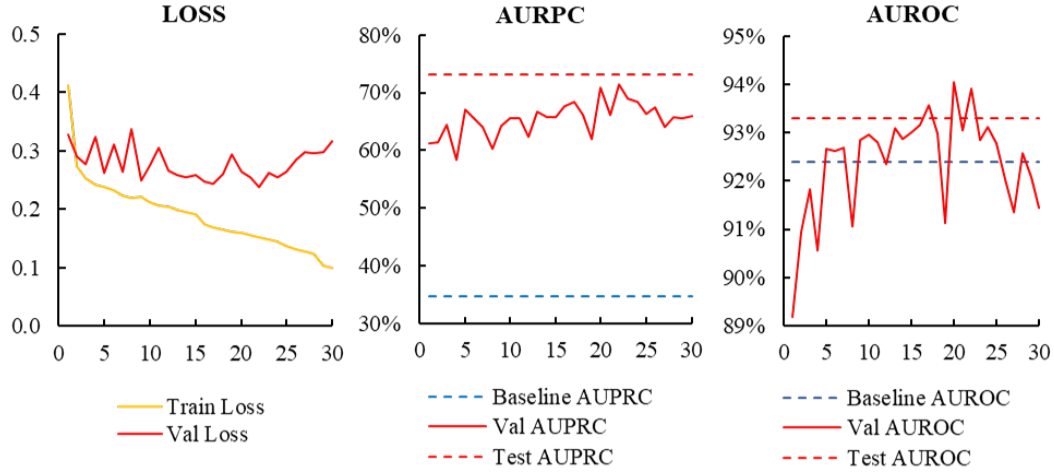
A Appendix

Figure 3: Reimplementation Performance



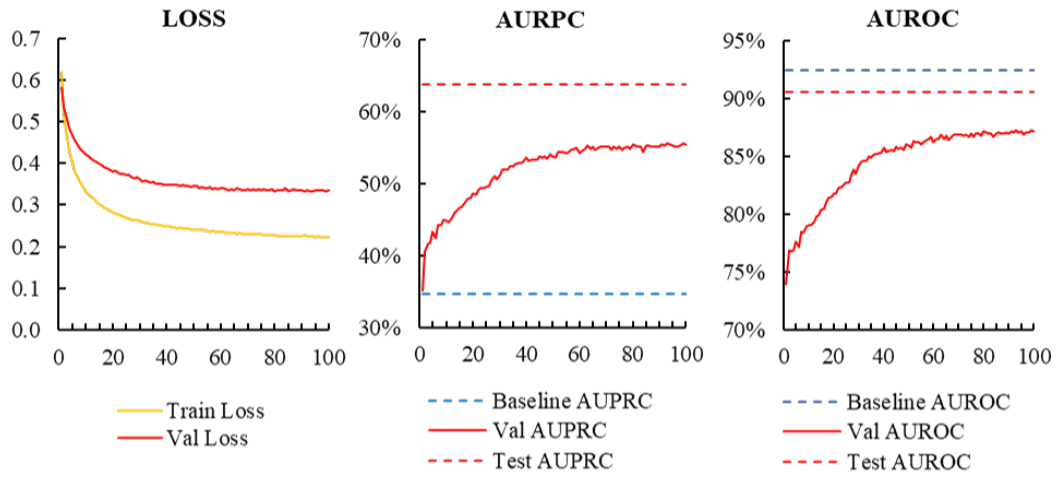
Note: Figure illustrates the training, validation, and test performance, and relevant comparison against the reported baseline in Zhao et al. [12].

Figure 4: Reimplementation Performance, No GraphNorm



Note: Figure illustrates the training, validation, and test performance, and relevant comparison against the reported baseline in Zhao et al. [12]. Due to the small model size, we experimented with removing GraphNorm layers after the graph convolution layers. The training becomes less stable, but the prediction metrics improve.

Figure 5: Reimplementation Performance, Double Layer Counts and Hidden Dimension



Note: Figure illustrates the training, validation, and test performance, and relevant comparison against the reported baseline in Zhao et al. [12]. We experimented with doubling layer counts for LSTM and graph convolution, as well as increasing the hidden dimension from 256 to 512. Model training is more stable but performance worsens.