Spatio-Temporal Wildfire Forecasting

Balaram Behera, William Boynton, Sarah Cheah, Jerry Cheung, Shrey Patel, Hui Min Tee

1 Introduction

Motivation. Over the past 5 years, California has faced 34 thousand fires over 10 million acres, killing dozens and destroying thousands of homes [6]. Increasing in number and size, these fires prompted research in wildfire prediction, patterns, and prevention, assisting state and federal institutions to reduce risk and improve mitigation strategy [9, 14]. Spatio-temporal wildfire data has never been used for deep learning analysis of wildfires due to the highly unstructured data repositories that hold this data. Hence, we focus on creating a usable dataset for future research and also focus on wildfire perimeter forecasting.

Problem Statement. In this project, we aim to solve the following problem: given a wildfire's current state, we find the forecasted geometry after 2, 4, and 6 days by applying recent deep learning innovations in spatio-temporal forecasting [20].

2 Literature Survey

Domain Knowledge. We looked to [14] for our project motivation to understand social impacts of wildfires and methods of combating wildfire effects, but found little for wildfire prediction. To contextualize vegetation data, [4] discusses the feedback loop between human-caused ignition on dry vegetation and the lack of wildfire policy implementation, but fails to suggest spatial considerations and actionable policy. In Williams's research where despite narrowing the mechanisms of fire spread to only fuels, we understand the importance of fuel features like vegetation [22]. Another paper by Pascolini-Campbell et al. used random forest regression to find that large burn areas depended on dry vegetation [15]. Their data can add to our vegetation dataset, despite their small fire data. Juan et al. leverages important features in wildfire behavior to forecast based on geological and economic data, but ignores the analytic implications of meteorological factors (wind, etc.) [12].

Functional Knowledge. We researched the spatio-temporal forecasting methods. Espinosa et al. focused on optimizing multiple linear regression models to build an ensemble learning model to predict air pollution levels [3]. To work with video data, Qiu et al. developed Local and Global Diffusion model, which divides the feature maps into local and global paths, focusing on the respective features [16]. Shi and Yeung discuss the pros and cons of forecasting algorithms and the practicality of scheduled sampling and boosting strategy [20]. To improve the model runtime in [3] and [20], we would need to change the scale at which we train our data. Cheng et al. considers crowd flow and encoding of temporal features to characterize their DHSTNet model (a spatio-temporal neural network model) for traffic flow prediction [2]. The paper introduces Graph Convolutional Networks which are useful in forecasting with external factors. While [2, 5, 16] may help make our spatial-temporal model more robust, we would need to translate their models and choose the compatible variables.

Previous Work. For spatial wildfire susceptibility, there are comparisons of deep learning models where SVM and Random Forest proved to be most accurate on test data [7]. Remote-sensing data, collected by monitoring physical changes in emitted radiation, is a key feature in prediction. The FireCast model predicts next-day fire likelihood using CNNs on remote-sensing data [17]. Similarly, Huot et al. used a multivariate dataset on a DL model to predict wildfire spread [10]. Using ANNs and SVMs, Sayad et al. recognized that vegetation, land surface temperature, and thermal anomalies to be key factors wildfire spread [19]. Feltman et al. suggest using GIS analysis to identify socioeconomic factors that contribute to wildfires in South Carolina [5]. Graff et al. discusses Poisson regression (non-ML) to predict wildfires 1-5 days in the future [8]. We can use a similar statistical method to perform inference, but like in [15], their small fire datasets reduce detail in providing accurate prediction of spread, thus additional weather conditions might add complexity to our model [8]. We will extend these methods to span multiple days and produce more accurate wildfire forecasts with only spatio-temporal perimeter data.

Our literature search showed the versatility and usage of spatio-temporal learning models for forecasting fires. Particularly, the diffusion-based convolution recurrent neural network, originally used to capture features in traffic flow, can be used to forecast fires by using a grid and diffusive resistances from our features [13]. Most of these applications require a base graph for the spatial element, so we use land grid as our base. A wildfire essentially diffuses from a point or area which can be captured in CNNs. Using CNNs instead of DCNNs have proven to accurately model traffic flow and other problems [2, 18, 21]. Another variant is the Graph GRU which is a generalization of the DCRNN which performed better for discrete traffic flow, but this might not extend as easily with continuous wildfire spread [23]. A further variant, the Graph Convolutional GRU has proven to work with grid graphs for landslide displacement with promising application with wildfire data [11].

Chosen Model. The model we chose is CNN+LSTM, consisting of a convolutional neural network paired with convolutional long-short term memory. This model serves as an appropriate starting point for us because of its spatio-temporal capabilities. It outperforms other baseline spatio-temporal models like U-net and SMA [21].

3 Proposed Method

Intuition. Predicting the spread of wildfires is a difficult task. There are numerous factors causing a wildfire and it is infeasible for a statistical or probabilistic model to cover every single factor. Thus, it is common for existing work to focus on only a subset of these factors and perform inference based on the selected factors. In addition, papers often focus on a small fixed area for prediction.

As for our innovation, we decided to include a greater geological area to train a downstream inference model. If we can curate the dataset, we can use the dataset to train a model that is more comprehensive than the state of the art as the model can make inferences on a greater area.

Data Collection. We chose to focus our prediction in the state of California. We obtained historical fire information from the NIFC [1]. We obtained 43GB of historical geological fire data. Each top level folder represents an individual fire. The sub directories within each of the fire folder contains geological fire information separated by days. For example, if the fire burned for 5 days, there will be 5 sub directories within the fire folder named by the date of the data it contains. These sub directories contains .shp files that capture the precise geolocation of each fire.

To clean the obtained data, we first filtered fires that occurred outside of the state of California. After that, we unarchived archived files and converted GIS formats to SHP files. We then removed spurious data with the use of python scripts. Namely, we removed data that either has an empty folder or contains corrupted .shp files that can't be opened. We only consider fires with incident data of at least 10 days. When days are missing, we used the previous fire state as a proxy for the missing day. Next, we standardized the projection of .shp files to EPSG 3310 which is ideal for distance and area calculations of polygons in California. To finally structure the files as the above mentioned directories, we had to laboriously restructure the folders manually as there was no structure with the public repository.

Utilizing the .shp files, we picked the centroid of the biggest fire on the day that it started burning as our reference point. We then expanded around the reference point towards north, south, west, and east by 20 km. each to generate a 40 km. by 40 km. bounding box. We then fix the bounding box and track the spread of the largest fire over time. For each squared kilometer in the bounding box, we check if the center of that squared kilometer is within the range of the fire. This allows us to transform a geometric .shp data into a binary matrix, significantly reducing the size of our dataset and speeding up the downstream computation.

Model Design. In terms of model design, we first implemented a vanilla dense neural network. The network contains two fully connected layers with the ReLU activation function and an output fully connected layer with the Sigmoid activation function. We used the Adam optimizer and Mean Squared Error as our loss function. The input to the model is a 40 by 40 matrix that contains binary values for each squared kilometer indicating whether the fire is burning within that region. The output of the model is a 40 by 40 matrix containing the logits of the last layer. We treat the logits in a greedy way: if the logit is greater than 0.5, we then consider that the fire is burning at that specific squared kilometer.

We will have two additional model designs that are used to compare with the dense baseline.

The first model is a CNN + LSTM, which contains a convolution neural network and a feed forward network. The model contains 4 convolution layers with a kernel size of 3 by 3, and output channels of 8, 8, 8, and 32 respectively. We perform 2 by 2 max pooling after the convolution layers. Connected to the max pooling layers are 4 long-short term memory networks with reLU activation. The input and output of the model is treated similarly as the dense baseline. We expect this model to be useful because of two reasons. First, we believe that CNN can attend to local information on the 40km by 40km bounding box with the use of kernels. This could potentially help the model recognize fire contours and make downstream predictions. Second, the use of LSTM keeps track of temporal information. By training the model with multiple days, the use of LSTM can attend to those information and thus make the final prediction more accurate.

The second model is an ensemble dense model. This model is slightly modified from the dense baseline. Instead of the baseline where we fed a 40 by 40 matrix to the model, we feed a 120 by 40 matrix to the model where the 40 by 40 matrix is repeated 3 times. This allows the model to make separate predictions for the 40 by 40 squared kilometer area and aggregate the results together. We expect this variation to perform better than the dense baseline. This is because the model is making multiple predictions on the spread of the fire and decide the final range of fire similar to a voting mechanism.

We will evaluate the performance of the models against each other as well as with the dense baseline. If we are successful, we will obtain a model that will accurately predict future fires. This could have the similar benefits as the curated dataset, allowing a client to take precautions to prevent and minimize the damage caused by the wildfires.

Visualization. Visualizing forecasting simulations is an important aspect of machine learning. Our interface utilized a map of California as a GeoJSON file, sectioned into 40×40 1 sqkm. squares. The map of California came from TigerLine as a JSON file, and was run through QGIS to create a grid-map intersection over the boundary of California to get the gridded output as a GeoJSON file. The interface combined this processed map and the D3.js library to display an interactive choropleth map. Each tile in the map of California corresponds to a 40x40 km area that serves as a placeholder for a more refined gridded area of 1 sqkm. squares. Once the data is selected on the map, it will be pushed through the model, producing a visualization of all tiles to which the fire has spread.

In our interface, the user will have the ability to select a 40x40 km tile located somewhere in California. (Figure 1) Selecting a tile will bring up a grid of 40×40 1 sqkm. tiles, each representing a small portion of land to which the fire may spread. Each tile in the sub-grid has a coordinate and value that signifies whether it is on fire and over which day(s) it is predicted to be on fire for. Here, the user can drag their mouse across the grid to draw an initial map of the fire at day 0. Due to there being a large number of squares with such small areas relative to the screen size, the selection was given a radius of 2 tiles so the user can select a large area in a short time. The user can then pass the initial data through the model by clicking the “Submit” button, which is one of the two buttons located below the sub-grid.

The "Submit" button passes in a 40x40 array of true/false values corresponding to whether each tile was deemed to be on fire or not. Once passed through the model, the output form is a dictionary of the same shape and size as the passed-in grid of fires. However, each cell in the dictionary contained a true/false value for days $0, 2, 4$, and 6. Each cell within the visualization 40x40 tile grid is then updated with the values in the dictionary and colored a shade of red. The shade of each cell depends on which day it is expected to be on fire, with the initial data represented by a dark-red, and days $2, 4$, and 6 being represented respectively by progressively lighter shades of red to make the visualization intuitive for the users. (Figure 2)

If the user would like to see what the fire might look like another 2, 4, and 6 days later, they can simply click the "Submit" button again. This will pass in the current day 6 data as the initial data, and output a new dictionary of values. If instead, they would like to start over on the same grid or highlight a region by accident, they can click the “Restart” button, which is the other button located below the grid. This will clear the 40 × 40 tile grid they are working on, and they will be able to create and submit a new set of initial fire conditions. However, if the user would like to select a different 40x40 km square entirely, they can return to the map of California by selecting the dark-grey square they previously selected. Then, they can select a different 40 × 40 grid input to visualize.

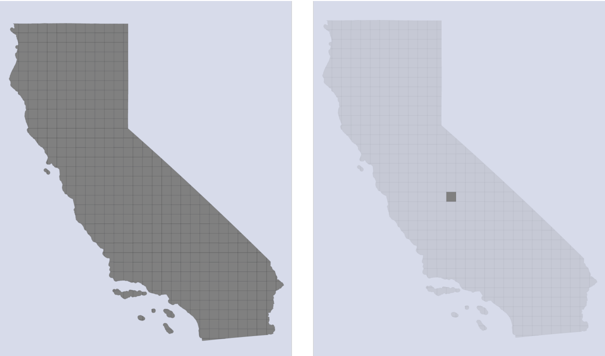

Figure 1: Upon mouse over, user may select a square representing a 40x40km area within California

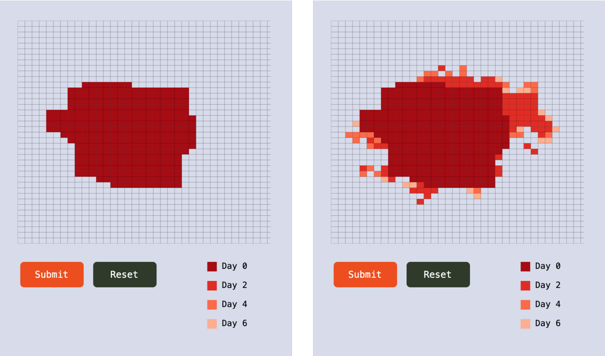

Figure 2: After user inputs a portion of the 40x40km area to simulate an initial fire, they can visualize the fire’s spread after 2, 4, and 6 days

4 Evaluation

Data Preparation We ran our experiments on the two models discussed above: the Convolutional LSTM model and the simple Feedforward (Dense layers) model. We employed Tensorflow’s Keras library and used the Sequential model framework along with their built-in layers. Each fire perimeter is represented by a 40 × 40 array where each cell contains a 0 or 1 value that indicates the presence of the fire in that specific cell (each cell refers to a 1 sqkm. area). This window size was chosen as many fires, although not close to 1,600 sqkm. in size, will spread wide and long. So to contain as many fires in the window without exceeding amounts of negative space, we chose these dimensions. We considered coordinate features in regards to the location of the fire, but their addition did not enhance the model in any metric.

We constructed a training dataset with the input data as one fire perimeter and the output as an array of the same fire’s perimeter 2, 4, and 6 days into the future (if the full sample array exists). We also repeat the input array 3 times as we are solving a One-To-Many problem, where there is a widely-accepted technique to repeat the input vector with Keras to essentially create a parallel ensemble of 3 vanilla models. We then, split the dataset into 50% training data, 25% validation data (during training), and 25% testing data, and this was done using Scikit-Learn’s functionality. After this preparation, we remain with approximately 1,000 valid training samples.

Accuracy Metrics We optimize both our models using Adam with the default learning rate of 0.001, and it tries to optimize the Mean Squared Error. The output of the model is in the unit interval by using Sigmoid activation, and since our data is a binary array, Mean Squared Error captures the correctness of a prediction as our goal is to have the least conflicts between the predicted output and the actual label. To measure the accuracy of our model, we used Keras’ built-in *binary accuracy* metric which will round the predicted output and then compare it with the ground truth.

However, as fire expansion is usually focused on the circumference of the fire, and with such a large window, accuracy can be high but not directly ensure a good model. Hence, tracking the percentage

Model	Data	MSE Loss	Accuracy	Precision	Recall	Prec. At Rec.	Rec. At Prec.
Dense	Training	0.36%	99.56%	99.34%	97.84%	99.85%	99.43%
Dense	Validation	0.0137	98.40%	95.49%	94.46%	99.01%	98.45%
Dense	**Testing**	**0.0151**	**98.27%**	**96.23%**	**93.05%**	**98.96%**	**97.85%**
Conv+LSTM	Training	0.0055	99.38%	98.76%	97.25%	99.60%	98.87%
Conv+LSTM	Validation	0.0156	98.20%	94.41%	94.29%	98.77%	97.87%
Conv+LSTM	**Testing**	**0.0158**	**98.19%**	**96.05%**	**92.73%**	**98.75%**	**96.93%**

Table 1: Final Results

of false positives and false negatives through the metrics of precision and recall are vital in evaluating our models. Precision measures the fraction of times the model is correct when it predicts a cell is on fire. Recall measures the fraction of times the model predicts correctly when the cell is actually on fire. Maintaining both high precision and recall is crucial, as without high precision the model is likely predicting larger fire spread and without high recall the model is likely predicting smaller fire spread. The correct simulation of the temporal expansion of wildfire perimeters depends on both metrics being similar in value and close to 100%. In our experiments, we found that precision was always high, but recall would take more epochs to increase and usually separated a good model from an ineffective one in our experiments.

Conv+LSTM Results For our Conv+LSTM model, we experimented with a various number of ConvLSTM2D layers (built-in Keras layer). We noticed with too many layers, the accuracy of our model with greatly suffer which we suspect to be caused by the aggressive down-sampling. Too much down-sampling would eradicate important features in predicting fire expansion as most of the important features lie on the circumference. Hence, we found best results with 2 layers with only one iteration of maximum pooling before flattening. For the proceeding Dense layers, we used 3 as with less, the model was not able to predict well and with more, the recall was usually low. More units in each layer contributed to over-fitting and low recall.

Slower and faster learning rates would either contribute to divergent training or unnecessarily slow improvement, so we remained with Adam's default learning rate of 0.001. Smaller batches in training resulted in both faster convergence and higher recall and precision, so we chose a batch size of 8. We trained this model for a total of 60 epochs where more epochs did not improve the model in any metric.

Dense Results We experimented with a basic feed-forward model since we were dealing with a small temporal frame of only 3 days, so such a model could still be useful without becoming overbearing with too many neurons. We trained models with different numbers of layers, and noticed that beyond 3-4 layers, the algorithm would not produce significantly better results in any of the metrics. Furthermore only a few hundred units for every layer would better results than with thousands, especially with respect to recall. Hence, we finalized on a structure with 3 layers with ReLU activation with 100, 400, and a final layer with 4800 units (the last layer's activation is the Sigmoid function). We train this model with a low batch size of 16 which produced the best results compared to both higher and lower sizes. Beyond 60 epochs, the model would not improve, so we only train for 75 epochs.

Discussion We display the final metrics for training, validation, and testing after fully training the models in Table 1. We have added two additional metrics: precision when recall is greater than 20% (Prec. At Rec.) and recall when precision is greater than 20% (Rec. At Prec.) which offer a better perception of these metrics by removing prediction outliers in the samples. From the metrics, it is easy to see that both the Dense and Conv+LSTM models performed almost the same, but the Dense model was slightly better in all metrics, especially in loss. However, the training of the Dense model was much faster and more consistent over different training and testing data splits. Hence, we choose the Dense model as our final spatio-temporal predictor.

We observed that the precision was usually higher than the recall, but since our visualization layers the outputs, the recall is more important in a good prediction. The Dense model would converge to a high recall much quicker than the Conv+LSTM model which would take more training iterations to consistently increase its recall metric. Moreover, in testing the model, we noticed that for larger fires there tend to be fixed points where the model does not predict the fire to increase and cover the entire window. We suspect that this is a result of most wildfires becoming fully contained once they reach such a large size. This containment bias is also observed when we see that the expansion of the fires slows down as the number of days increases. The lack of diverse data has more implications in the model's capabilities as tends to show more vertical growth than linear growth which is the case for most fires in California due to its climate and geography. By integrating more data, we can analyze these subtle biases in our data with more rigor.

Visualization Usability. Ideally, we would have sought out volunteers from the wildfire management field to help us evaluate our interactive visualization's effectiveness since that is our target audience. This would allow us to measure the level of utility of our visualization, i.e., whether it improves wildfire prediction and risk mitigation strategies. However, with time and resource constraints, we mainly focused on testing our visualization's usability through our peers.

For our simple user study, we did not give subjects much guidance other than to follow the instructions on the interactive visualization. Following that, we surveyed them about their user experience, focusing on any difficulties and frustrations they faced in interacting with the visualization. Through this subjective feedback from the subjects as well as observational notes from us, we were able to qualitatively measure our visualization's ease of use, i.e., usability, and improve our visual representation in an iterative loop.

For instance, our chosen colors were because some users voiced their preferences for a warm color palette over a cool one since it intuitively made more sense as a fire representation. Based on feedback, we also enhanced the speed and thus ease of fire "drawing" by enabling the user to drag their mouse over the 40×40 grid, instead of manually clicking on one cell at a time to paint over the grid. In addition, some users also found discontinuity of information presented in our initial design where we used separate grids representing each predicted day. Thus, with their feedback, we changed our design to the current layering of our wildfire forecasting results for 0, 2, 4, and 6 days in the same grid which is now more intuitive to our users and represents the functionality of our model better.

5 Conclusion

Our focus was to develop a CNN+LSTM spatio-temporal wildfire forecasting model for California, USA. After comparing the CNN+LSTM model to a dense neural network model, the latter performed better and became our chosen model for deployment and incorporation into our visualization. We integrated our model with a D3 visualization in JavaScript to allow the user to select 1 sqkm. areas from a 1,600 sqkm. region that would represent the presence or absence of a fire. Our model yielded high accuracy for predicting the fire spread 2, 4, and 6 days after the initial fire selection.

We could further improve our model by taking into account other factors that influence fire spread like vegetation, elevation, and road presence. Although we would need more powerful computing capabilities for handling multiples datasets that compose of hundreds of gigabytes of data, considering such factors that exist in the real world would make our prediction more reflective of reality. We would not only have a better prediction of fires in California, but we could easily extend the model to the rest of the country, provided the corresponding fire data is accessible. If other fire data is similar to the NIFC California fire data we used, our model can be easily adapted to locations beyond California. With our model's superb accuracy and implementation of the mentioned improvements, our fire prediction model could be applied by local authorities in preparing for and combating wildfire instances.

In completing this project, all team members contributed a similar amount of effort.

References

[1] National Interagency Fire Center. 2022. Incident Specific Data. https://ftp.wildfire.gov/

[2] Sibo Cheng, I. Colin Prentice, Yuhan Huang, Yufang Jin, Yi-Ke Guo, and Rossella Arcucci. 2022. Data-driven surrogate model with latent data assimilation: Application to wildfire forecasting. *J. Comput. Phys.* 464 (2022), 111302. https://doi.org/10.1016/j.jcp.2022.111302

[3] Raquel Espinosa, Fernando Jiménez, and José Palma. 2022. Multi-objective evolutionary spatio-temporal forecasting of air pollution. *Future Generation Computer Systems* 136 (2022), 15–33. https://doi.org/10.1016/j.future.2022.05.020

[4] Farshad Farkhondehmaal and Navid Ghaffarzadegan. 2022. A cyclical wildfire pattern as the outcome of a coupled human natural system. *Scientific reports* 12, 1 (2022), 1–14.

[5] Joel A. Feltman, Thomas J. Straka, Christopher J. Post, and Stephen L. Sperry. 2012. Geospatial Analysis Application to Forecast Wildfire Occurrences in South Carolina. *Forests* 3, 2 (2012), 265–282. https://doi.org/10.3390/f3020265

[6] CAL FIRE. 2022. *Stats and Events*. https://www.fire.ca.gov/stats-events/

[7] Omid Ghorbanzadeh, Khalil Valizadeh Kamran, Thomas Blaschke, Jagannath Aryal, Amin Naboureh, Jamshid Einali, and Jinhu Bian. 2019. Spatial Prediction of Wildfire Susceptibility Using Field Survey GPS Data and Machine Learning Approaches. *Fire* 2, 3 (2019). https://doi.org/10.3390/fire2030043

[8] Casey A. Graff, Shane R. Coffield, Yang Chen, Efi Foufoula-Georgiou, James T. Randerson, and Padhraic Smyth. 2020. Forecasting Daily Wildfire Activity Using Poisson Regression. *IEEE Transactions on Geoscience and Remote Sensing* 58, 7 (2020), 4837–4851. https://doi.org/10.1109/TGRS.2020.2968029

[9] Katie Hoover and Laura A Hanson. 2021. *Wildfire statistics*. Technical Report. Congressional Research Service.

[10] Fantine Huot, R. Lily Hu, Nita Goyal, Tharun Sankar, Matthias Ihme, and Yi-Fan Chen. 2022. Next Day Wildfire Spread: A Machine Learning Dataset to Predict Wildfire Spreading From Remote-Sensing Data. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), 1–13. https://doi.org/10.1109/TGRS.2022.3192974

[11] Yanan Jiang, Huiyuan Luo, Qiang Xu, Zhong Lu, Lu Liao, Huajin Li, and Lina Hao. 2022. A Graph Convolutional Incorporating GRU Network for Landslide Displacement Forecasting Based on Spatiotemporal Analysis of GNSS Observations. *Remote Sensing* 14, 4 (2022), 1016.

[12] Pablo Juan, Jorge Mateu, and M Saez. 2012. Pinpointing spatio-temporal interactions in wildfire patterns. *Stochastic Environmental Research and Risk Assessment* 26, 8 (2012), 1131–1150.

[13] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJiHXGWAZ

[14] National Academies of Sciences, Engineering, and Medicine et al. 2020. *Implications of the California wildfires for health, communities, and preparedness: Proceedings of a workshop*. National Academies Press.

[15] Madeleine Pascolini-Campbell, Christine Lee, Natasha Stavros, and Joshua B Fisher. 2022. ECOSTRESS reveals pre-fire vegetation controls on burn severity for Southern California wildfires of 2020. *Global Ecology and Biogeography* (2022).

[16] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. 2019. Learning spatio-temporal representation with local and global diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12056–12065.

[17] David Radke, Anna Hessler, and Daniel Ellsworth. 2019. FireCast: Leveraging Deep Learning to Predict Wildfire Spread. In *IJCAI*.

[18] Michele Ronco, Ioannis Prapas, Spyros Kondylatos, Ioannis Papoutsis, Gustau Camps-Valls, Miguel-Ángel Fernández-Torres, Maria Piles Guillem, and Nuno Carvalhais. 2022. Explainable deep learning for wildfire danger estimation. In *EGU General Assembly Conference Abstracts*. EGU22–11787.

[19] Younes Oulad Sayad, Hajar Mousannif, and Hassan Al Moatassime. 2019. Predictive modeling of wildfires: A new dataset and machine learning approach. *Fire Safety Journal* 104 (2019), 130–146. https://doi.org/10.1016/j.firesaf.2019.01.006

[20] Xingjian Shi and Dit-Yan Yeung. 2018. Machine learning for spatiotemporal sequence forecasting: A survey. *arXiv preprint arXiv:1808.06865* (2018).

[21] Selim Furkan Tekin, Oguzhan Karaahmetoglu, Fatih Ilhan, Ismail Balaban, and Suleyman Serdar Kozat. 2021. Spatio-temporal weather forecasting and attention mechanism on convolutional lstms. *arXiv preprint arXiv:2102.00696* (2021).

[22] F.A. Williams. 1977. Mechanisms of fire spread. *Symposium (International) on Combustion* 16, 1 (1977), 1281–1294. https://doi.org/10.1016/S0082-0784(77)80415-3

[23] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit Yan Yeung. 2018. GaAN: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*.