Spatiotemporal Wildfire Forecasting in California Using Recurrent Neural Networks

Kai Malloy

University of California, Irvine
Bren School of Information and Computer Sciences
ksmalloy@uci.edu

May 20, 2021

I. Abstract

Wildfires are a major concern for many regions that are prone to high winds, droughts, and heat waves. California is one of these regions and every year, land managers use forecasting tools to prevent and mitigate the spread of out of control wildfires. Existing solutions to forecasting wildfires have limitations as they often do not use a large quantity of high spatial and temporal resolution data. In this investigation, I explore the recurrent neural network extension to existing work on convolutional neural networks using high resolution, California wildfire data. Through experimenting with this model, I seek to understand how the recurrent neural network compares against baseline models and whether the model can produce valuable insight at intermediate time steps.

II. Introduction

For many regions with warmer and drier climates, wildfires are a recurring danger. Out of control fires cost billions of dollars in damages, increase homelessness, cause premature deaths (Johnston et al., 2012), release chemicals that pose health hazards, and contribute to global warming (Adetona et al., 2016). Wildfires are a significant area of interest to California as counties are ravaged by large blazes every year during the summer and fall seasons. Furthermore, current data suggest that major wildfires are becoming more frequent and intense; a trend that is projected to continue for decades (Sun et al., 2019). In combating wildfires, forecast models are necessary as they allow land managers to evaluate preventative measures, they support suppression efforts, and can lead to an increase in scientific understanding of fire dynamics.

There have been multiple approaches to forecasting wildfires but the existing solutions have their limitations and tradeoffs. Empirical, simulation-based models are often computationally expensive, are not trained or validated on large quantities of observational data, and rely on region-specific fuel models that limit their applicable scope (Finney, 1998; Finney et al., 2011). Machine learning approaches have not been extensively explored and those that have been introduced often sacrifice spatial resolution, temporal resolution, or quantity of training and evaluation data (Hodges and Lattimer, 2019; Radke, Hessler, and Ellsworth, 2019). One area in which both methods fall short is integrating and modeling high temporal resolution weather and fire spread data. This is a particular point of concern, as weather is one of the most important factors affecting fire intensity, spread, and duration; and the leading factor in the changing fire regime forecasted over the next century (Flannigan et al., 2009).

In this report, I describe my investigation into the extent to which the predictive accuracy of a high-resolution, learning-based forecast model can be improved by explicit temporal modeling of weather and fire spread. I build on existing work with convolutional neural networks and combine it with recurrent neural networks to investigate the model performance using California wildfire data. The primary goals of the project are to gauge the improvement in key evaluation metrics, identify changes in computational cost to train and evaluate the model, and to gain insight from the temporal evolution of the model's forecast.

III. Data

In this project, I use active fire detection and meteorology satellite data from May 9th 2012 to January 1st 2018. There is a total of 152,500 data points within the six year span. Both types of data are bounded within the California state lines as shown in figure 1. The fire detection data is generated at a 375 meter resolution every 12 hours while the meteorology data is generated at a 13 kilometer resolution every hour and to match the resolution, the meteorology data is down sampled to 375 meters. Each data point in the dataset will have a t-48, t-36, t-24, t-12, and t+0 time step for a total of 5 lags to predict the t+12 time step. The meteorology data can be generated at finer increments and will be used as intermediate lags between the 12 hour time steps depending on the time interval used.

i. Active Fire Detections

The active fire detection data is from the Visible Infrared Imaging Radiometer Suite (VIIRS) 375m active fire product which is collected by the VIIRS sensor aboard NASA and the National Oceanic and Atmospheric Administration's (NOAA) Suomi National Polar-orbiting Partnership and NOAA-20 satellites (Schroeder et al., 2014). The two satellites combine to produce 12 hour detections at a 375 meter resolution every day. The sizes of fires varies from only a few detections to clusters of

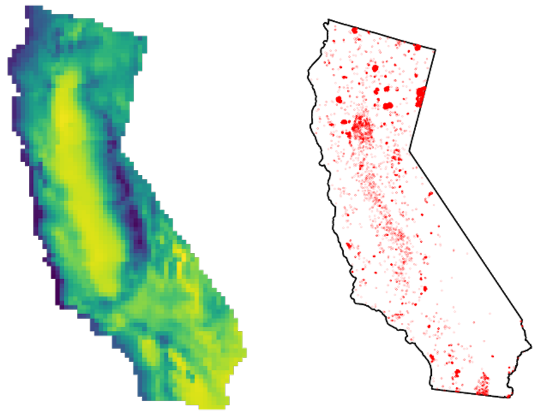

Figure 1: *Visualization of 2012 Rapid Refresh 13km Weather Forecast (Left) and VIIRS 375m Satellite Active Fire Product (Right)*

detections and can be caused by vegetation fires, volcanic eruptions, and flares from gas wells. For this paper, all fire detections are included for the models to learn the patterns of different fires in hopes of being able to apply it to different regions in the future.

ii. Meteorology

For the meteorology data, the NOAA's Rapid Refresh (RAP) 13km assimilation model (Benjamin et al., 2016) is used. This model assimilates information from a variety of observational sources every hour at a 13 kilometer resolution, and produces weather forecasts up to 18 hours in the future. There are four types of meteorology data used for this experiment: temperature, humidity, wind, and precipitation. Wind is split into the U and V components for a total of five layers of meteorology. 3 hour, 4 hour, 6 hour, and 12 hour increments are used for experiments and for each interval, meteorology data is averaged to produce an overall view of the change in weather.

IV. Model Architecture

The decision to use recurrent neural networks (RNNs) is natural as RNNs are well-studied for temporal problems. It enables efficient modeling of time series data by propagating and updating information from previous time steps using non-linear, differentiable transformations (Boden, 2002). Convolutional neural networks (CNNs) are able to capture spatial information from images and by combining CNN and RNN frameworks, spatial and temporal information can be learned. CNN+RNN architectures have proven to be successful in a number of related tasks including precipitation forecasting (Shi et al., 2015), traffic prediction in transportation networks (Yu et al., 2017), music classification (Choi et al., 2017), and video frame prediction (Hosseini et al., 2019). CNN+RNNs are especially useful in producing interpolated predictions at intermediate time steps which is difficult to produce in non-recurrent CNNs.

The CNN+RNN model architecture for this paper has been modeled off the UNet architecture proposed by Ronneberger, Fischer, and Brox (2015) and has been modified to include a gated recurrent unit (GRU) at the bottom of the model as shown in figure 2. At each depth, the model consists of a double convolution layer with an alternating ReLU layer. A double convolution is two convolutions with the same number of channels. With each additional depth, the number of channels is gradually increased while the size of the image is reduced using a pooling layer. The data is then reduced to

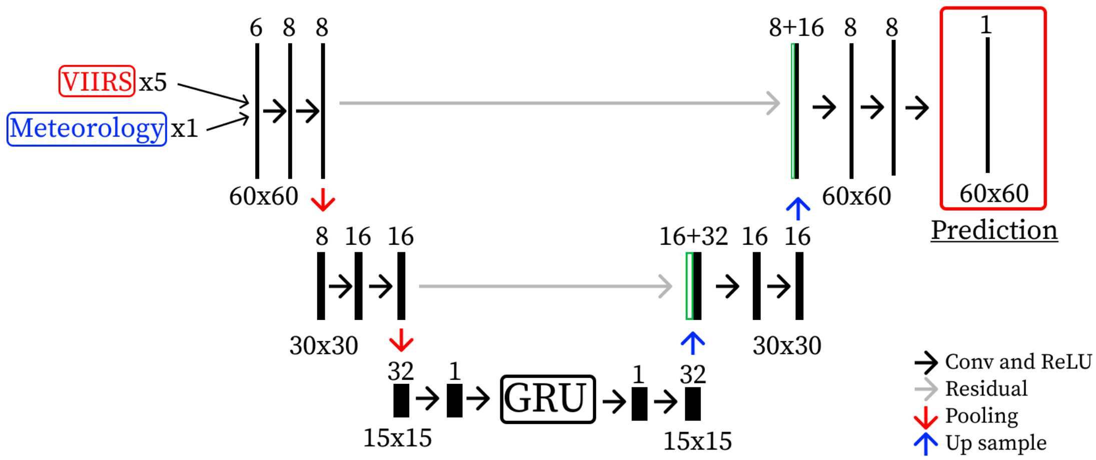

Figure 2: *UNet Style Architecture for CNN+RNN*

one channel at the bottom of the encoder and fed into the GRU. The GRU is the hidden state and passes an encoded representation of the current time step to the next time step. The decoder then reverses the process by up sampling and decreasing the number of channels. Unlike the UNet model, this model maintains the same image size at each depth to be able to produce same size predictions as the target data.

V. Model Training and Evaluation Methods

The dataset is first shuffled by data point (groups of lags) before performing an 80/20 split for training and testing data. This ensures that both parts have data from all years. This is important because each year may have a drastic difference in the amount and type of fires. Stochastic gradient descent is used as an optimization algorithm which randomly splits the data into batches and runs for a number of epochs. A batch is the number of samples to process while the epoch is how many times the model will view the whole training dataset. The models are all trained using cross entropy (CE) and mean squared error (MSE) is used as an additional evaluation metric. For both metrics, a lower value results in a more accurate prediction. The CNN+RNN model is compared against four baseline models: persistence, logistic regression, CNN, and UNet. The persistence model is the current observation used directly as a prediction for the next time step. This model performs fairly well given the temporal nature of wildfires. The CNN and the UNet are both CNNs but the CNN in this paper refers to a simple CNN with 8 by 16 by 32 by 1 channels.

VI. Results

The experiments were conducted in three steps. First, various extensions to the RNN model were tested to find the best architecture and parameters. Then, this configuration was used to view the models output at different time steps: 3 hour, 4 hour, and 6 hour, and 12 hour. To conclude, the RNN model was compared to baseline models to understand the predictive accuracy.

i. RNN Exploration

Various experiments for the CNN+RNN model are summarized in table 1 with the cross entropy score and the mean squared error. One aspect of the model was changed while the remaining factors remained constant for each experiment.

	CE Train	CE Test	MSE Test
Double Conv, **No Residuals**, VIIRS, Kernel: 7x7, Depth 1	0.0756	0.0796	0.0179
Double Conv, Residuals,VIIRS, Kernel: **3x3**, Depth 1	**0.0527**	0.0731	0.0172
Double Conv, Residuals,**VIIRS + Weather**, Kernel: 7x7, Depth 1	0.0536	**0.0691**	**0.0171**
Single Conv, Residuals, VIIRS, Kernel: 7x7, Depth 1	0.0636	0.0834	0.0191
Double Conv, Residuals, VIIRS + Weather, Kernel: 7x7, **Depth 2**	0.0596	0.0854	0.0180

Table 1: *Various Experiments for the CNN+RNN Model*

The first experiment is with removing the residual connections from the UNet architecture to make it a simple encoder-decoder model. The second experiment is to change the kernel size for each convolution operation to 3 by 3 instead of 7 by 7. Padding of 1 is used around the image instead of 3 for a 3 by 3 kernel size to maintain the same image dimensions with each convolution operation. The third experiment is to include the five weather variables with VIIRS as input into the model. The fourth experiment is to replace all double convolution operations with a single convolution. The last experiment is to change the depth from 1 to 2.

The third model performed the best for CE and MSE Test scores while the second model was second. The 7 by 7 kernel size allows for the model to get a greater grasp of the change in neighboring pixels while smoothing out the results which leads to a higher score. The first and fourth experiment show the effectiveness of the UNet architecture as removing residual connections and changing the double convolution to a single convolution result in a lower predictive accuracy. The last model showed an increasing CE Test score after the first epoch of training which signified that the model is most likely overfitting so I decided not to make the model more complex by keeping a depth of 1.

ii. Weather Exploration

Since the VIIRS fire detections are only produced every 12 hours, the RNN model is valuable as it can provide predictions at more frequent time intervals. For this paper, I have decided to look at 3 hour, 4 hour, and 6 hour weather data aside from the 12 hour dataset. For each time step where there is no VIIRS lag, I pass in the last VIIRS lag with the weather data to the model. Given that the 12 hour data requires 5 lags, the 6 hour data requires 10 lags, the 4 hour data requires 15 lags, and the 3 hour data requires 20 lags. This adds to the computation load as the RNN model will iterate for each lag and this occurs for each batch and each epoch. Since the dataset is fairly large, the 12 hour data takes 20 minutes for 5 epochs using a GPU and the 3 hour data takes approximately an hour to run.

The model outputs of the intermediate time steps for each dataset are shown in figure 3. Each image is a visualization of the conditional probability from 0 to 1 of the likelihood of a fire occurring. The image uses a color scheme from blue to yellow with blue representing the minimum probability and yellow representing the maximum probability in a single image. Only the last VIIRS lags from t+0 to t+12 are shown for each dataset.

Overall, there is not a significant change in shape but the overall probabilities do decrease and increase in some areas. The 4 hour predictions have close to no change but the 3 hour predictions have a gradual change. When comparing the first lag and the last lag as shown in figure 4, the middle area and the right edge decreases in probability over time. A change can also be seen in the right side of the 6 hour predictions as it increases in probability. This shows how the model struggles to predict drastic changes in movement but can grasp a change in the intensity of a fire in a given spot.

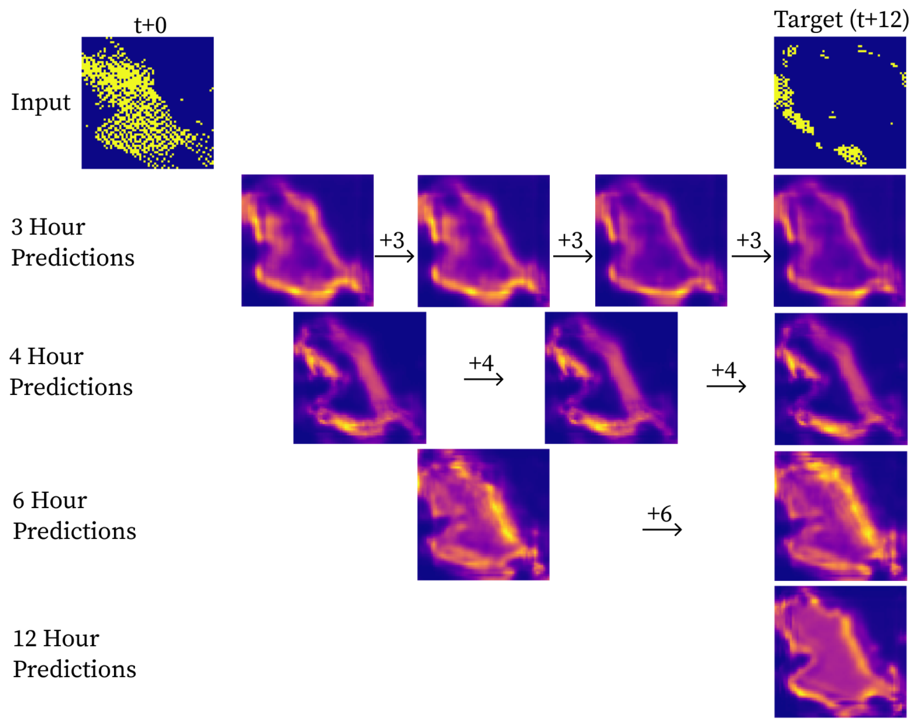

Figure 3: *Output of Intermediate Steps for 3 Hour, 6 Hour, and 12 Hour Predictions*

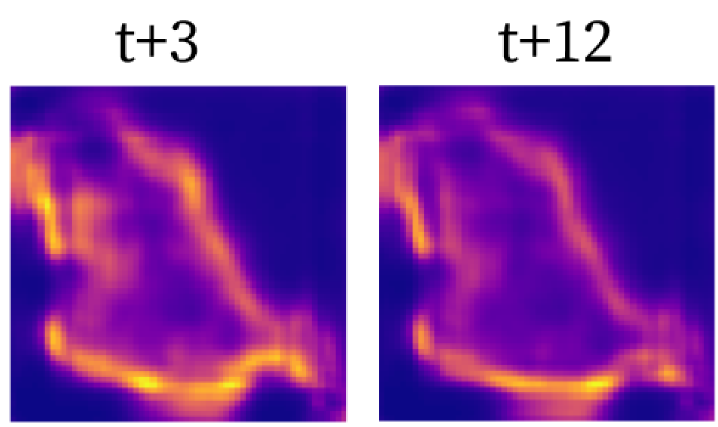

Figure 4: *3 Hour Prediction at t+0 and t+12*

iii. Model Comparisons

When compared to the baseline models, the CNN+RNN models performed better than the persistence and logistic regression models but slightly worse than the pure CNN model as shown in table 2.

	CE Train	CE Test	MSE Test
Persistence	1.3309	2.0660	0.0598
Logistic Regression	0.0646	0.0828	0.0187
CNN	**0.0512**	**0.0665**	**0.0169**
UNet w/weather	0.0565	0.0708	0.0176
CNN + RNN 12hr w/weather	0.0536	0.0691	0.0171
CNN + RNN 6hr w/weather	0.0525	0.0704	0.0172
CNN + RNN 4hr w/weather	0.0542	0.0707	0.0174
CNN + RNN 3hr w/weather	0.0531	0.0706	0.0172

Table 2: *Model Results Comparison*

I expected the CNN+RNN model to outperform all other models but it may have under performed due to the complexity of the model. Since the CE train score of the CNN model is lower than the CNN+RNN model's score, the CNN+RNN model may not have learned the best set of parameters because it was harder to optimize due to the complexity. The CNN+RNN model at least showed a slight improvement in performance over the regular UNet which shows that the hidden state is passing information to the next time step. Among the four CNN+RNN models, there were very small differences in performance but the 12 hour model performed the best, presumably because each time step had an up-to-date VIIRS input to correct the prediction.

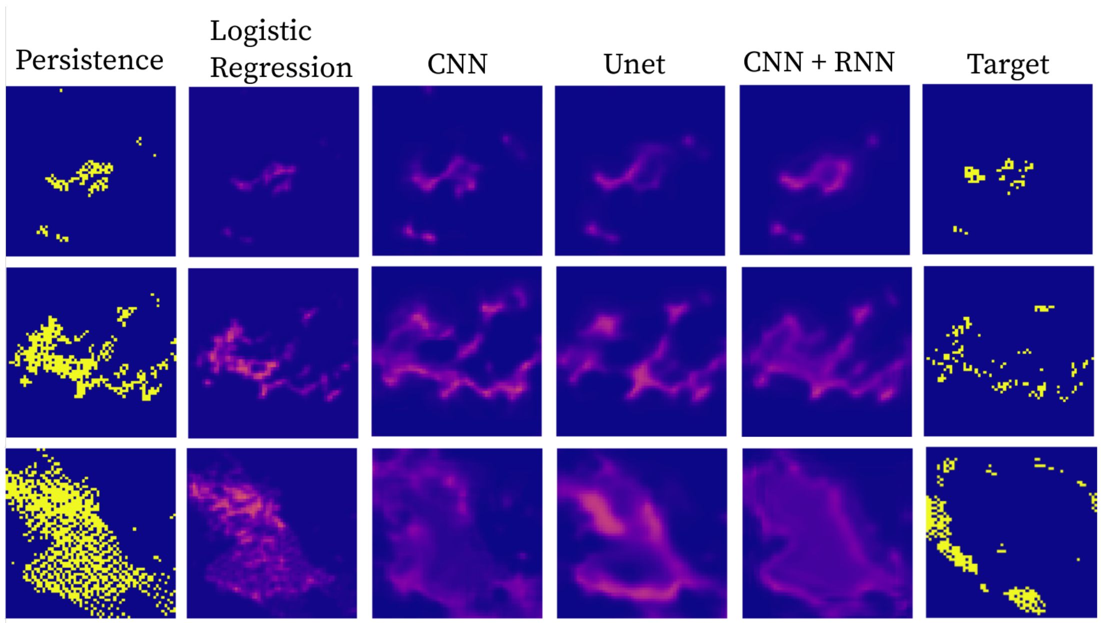

Figure 5: *Prediction Output of Each Model*

The three fires in figure 5 were selected by size from small to large and the color map for the images have been set to a minimum of 0 and a maximum of 1 for consistency. The largest fire on the bottom row is the same dataset and 12 hour CNN+RNN prediction from figure 3. The persistence model is simply the previous time step (t+0) which is the input used by all models to predict the target (t+12). Looking at the fires, there is a general correlation between model complexity and a smoothing out of the predictions. The logistic regression model predicts extremely similar to t+0 with little to no smoothing. As the model increases in complexity from CNN to UNet to CNN+RNN, the prediction gets more smoothed out. For example, the CNN+RNN model in the second row produces a very smooth prediction that covers a wider area than the target. For the last row, the CNN, UNet, and CNN+RNN models learned to predict a circular pattern given a large, dense fire in t+0. The CNN predicts a soft boundary while the UNet over predicts, and the CNN+RNN model predicts a sharp circular boundary but the inside shape is smoothed out because of the model complexity condition previously mentioned.

VII. Discussion

While the RNN models are overall successful, there are several future directions that can be explored. Hourly forecast of fires is an interesting topic to many climate scientists, although I did not explore this due to memory limitations. Spatial information can be further passed into the GRU using a 2d representation and convolutions instead of a 1d representation similar to Shi et al. (2015). Another variation would be to have both 1d and 2d internal representations passed through. Instead of passing the same fire detection between 12 hour intervals, the previous time step output could be used as input for the next time step, although this may be unnecessary given the flow of information in the hidden layer of the RNN. Transformers have also gained popularity in sequential problems, most notably in Natural Language Processing (Vaswani et al., 2017) and could outperform other models if applied to wildfires given the temporal nature of wildfire spread.

Different datasets can also bring new insight when used with the models. VIIRS and RAP have data available for the years 2019 and 2020 which could be aggregated with the current data to increase the size of the dataset. The new data is of interest given the record breaking wildfires in the past few years. NOAA also has a High-Resolution Rapid Refresh 3km model that can forecast weather at a 3 kilometer resolution every 15 minutes. Using this dataset would improve spatial and temporal resolution but will sacrifice the size of the dataset as it is only available from 2014. Inclusion of land cover is another potential area of improvement as the models can learn how vegetation and urban areas dictate fire spread.

With success in forecasting California wildfires, it will be interesting to see how the models perform in other regions in the United States and the world. In particular, applying and adapting the models to the Amazon basin and Central Africa are of interest due to the differences in fire dynamics. Specifically, many small, human-caused fires that occur relatively frequently; which the model must account for it to be accurate.

VIII. Conclusion

In this paper, I experimented with various extensions of the CNN+RNN model and showed that the best model can out perform all baseline models besides the pure CNN model while capturing change in growth of the wildfire at more frequent time steps. The 6 hour, 4 hour, and 3 hour predictions showed that the CNN+RNN model has a harder time learning drastic changes in fire movement but is able to learn changes in fire intensity.

IX. Acknowledgements

Special thanks to Casey Graff, Padhraic Smyth, and the rest of the fire research team at the University of California, Irvine. They have helped me immensely over the past year and a half in learning about the wildfire problem and guiding me in exploring various machine learning models. This paper has been produced with their help for the Bren School of Information and Computer Sciences Honors Program.

References

Adetona, Olorunfemi et al. (2016). "Review of the health effects of wildland fire smoke on wildland firefighters and the public". In: *Inhalation Toxicology* 28.3, pp. 95–139.

Benjamin, Stanley G et al. (2016). "A North American hourly assimilation and model forecast cycle: The Rapid Refresh". In: *Monthly Weather Review* 144.4, pp. 1669–1694.

Boden, Mikael (2002). "A guide to recurrent neural networks and backpropagation". In: *The Dallas Project*.

Choi, Keunwoo et al. (2017). "Convolutional recurrent neural networks for music classification". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 2392–2396.

Finney, Mark A (1998). *FARSITE, Fire Area Simulator–Model Development and Evaluation*. 4. US Department of Agriculture, Forest Service, Rocky Mountain Research Station.

Finney, Mark A et al. (2011). "A method for ensemble wildland fire simulation". In: *Environmental Modeling & Assessment* 16.2, pp. 153–167.

Flannigan, Mike D et al. (2009). "Implications of changing climate for global wildland fire". In: *International Journal of Wildland Fire* 18.5, pp. 483–507.

Hodges, Jonathan L and Brian Y Lattimer (2019). "Wildland fire spread modeling using convolutional neural networks". In: *Fire Technology* 55.6, pp. 2115–2142.

Hosseini, Matin et al. (2019). "Inception-inspired lstm for next-frame video prediction". In: *arXiv preprint arXiv:1909.05622*.

Johnston, Fay H et al. (2012). "Estimated global mortality attributable to smoke from landscape fires". In: *Environmental Health Perspectives* 120.5, pp. 695–701.

Radke, David, Anna Hessler, and Dan Ellsworth (2019). "FireCast: Leveraging Deep Learning to Predict Wildfire Spread." In: *IJCAI*, pp. 4575–4581.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, pp. 234–241.

Schroeder, Wilfrid et al. (2014). "The New VIIRS 375 m active fire detection data product: Algorithm description and initial assessment". In: *Remote Sensing of Environment* 143, pp. 85–96.

Shi, Xingjian et al. (2015). "Convolutional LSTM network: A machine learning approach for precipitation nowcasting". In: *arXiv preprint arXiv:1506.04214*.

Sun, Qiaohong et al. (2019). "Global heat stress on health, wildfires, and agricultural crops under different levels of climate warming". In: *Environment International* 128, pp. 125–136.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *arXiv preprint arXiv:1706.03762*.

Yu, Haiyang et al. (2017). "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks". In: *Sensors* 17.7, p. 1501.