THESIS

DEEP LEARNING FOR SHORT-TERM PREDICTION OF WILDFIRE USING GEOSTATIONARY SATELLITE OBSERVATIONS

Submitted by
Yousef Saqer
Department of Electrical and Computer Engineering

In partial fulfillment of the requirements
For the Degree of Master of Science
Colorado State University
Fort Collins, Colorado
Fall 2024

Master's Committee:

Advisor: Haonan Chen

Mahmood R Azimi-Sadjadi
Yu Wei

ABSTRACT

DEEP LEARNING FOR SHORT-TERM PREDICTION OF WILDFIRE USING GEOSTATIONARY SATELLITE OBSERVATIONS

The aim of this thesis is to utilize the Geostationary Operational Environmental Satellite (GOES) data for predictions regarding the intensity and potential path of wildfires. Using GOES to identify wildfires and extracting data from those events to help train a deep learning model. Three fires were selected for training the deep learning model: the Sequoia, Calwood, and Maui fires. The GOES data of the fires was obtained from band 7 which operates in the Shortwave Window or 3.9μm wavelength, band 7 is able to capture hotspots which is beneficial for wildfire prediction. The radiance data from band 7 is pulled from an Amazon Web Service (AWS) and becomes part of a dataset of 2513 samples. The data is then stacked to form a time series of approximately two hours and converted into a compressed h5 file. The pipeline distributes the dataset by taking in twenty five minutes of input data and feeding four different models to predict seventy five minutes, one hundred minutes, and one hundred and twenty five minutes of data. The data is then fed into a deep learning model utilizing a model known as Self Attention Gated Recurrent Unit (SaGRU). The SaGRU is tested four times, once for predicting seventy five minutes, once for predicting one hundred minutes, and twice for one hundred and twenty five minutes. The models were then compared against each other regarding Mean Squared Error (MSE) and Mean Absolute Error (MAE) along with the Normalized Mean Squared Error (NME) and the Normalized Mean Absolute Error (NMAE). Each metric was taken along multiple thresholds comparing the performance when hotspots are present and when hotspots are absent. The resultant showed that regardless of the sequence length, there was minimal negative impact on early predictions, but as the predicted sequence increased significant loss could be seen on the later predicted frames.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Haonan Chen, for taking me as his student and guiding me through this research process. Dr. Chen had taken time out of his week to help ensure my thesis was making progress, that any questions or concerns I had were answered, and proper next steps were taken. I would also like to thank Yifan Yang for his assistance; Yifan would sit down and address any confusion I had; he would help with designing and researching the pipeline for this thesis and listen in during each report I had made. I would like to thank Yao Shun for also supporting me in this project; he sat with me and explained his prior research and ensured that I understood how to implement his research for this project. Lastly I would like to thank my friends and family, for it was them who pushed me to pursue higher education and accompanied me during my journey.

DEDICATION

I would like to dedicate this thesis to my family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Problem Statement and Motivations

As a consequence of anthropogenic fires, ecological changes, and climate change the world has seen a sharp rise in wildfires. Each year brings difficult challenges as wildfires begin suddenly and quickly intensify over months, leaving acres of preventable destruction behind it [3]. This growing frequency and intensity poses a significant threat not only to the ecosystem but to humans [4] as well; as animals are forced from their habitats a loss of biodiversity can occur which potentially leads to species being unable to repopulate. The intensity of a wildfire can break down the structure and composition of the soil, crippling the ability of the forest to regenerate as it may struggle to hold any water [5]. The trees burning not only directly affects the human population, as the particle matter from the fire can result in respiratory damage, but the burning canopy can release carbon emissions into the atmosphere contributing to and potentially accelerating climate change [6]. With all these devastating impacts, a solution is utilizing deep learning for wildfire predictions.

Deep Learning has recently been used in plenty of environmental applications, from precipitation path prediction [7], where hurricanes from the coast southeastern coast were fed into a deep learning model to predict the path of storms with good accuracy was achieved, to pollutant detection [8], in which air pollutants such as NO2, SO2, SPM, RSPM, and PM from 1990 to 2020 from cities in India are fed to a model to predict future pollutant indices; the model was found to be able to predict future pollutant index from cities that it was not trained in with great accuracy. Deep Learning is a branch of machine learning that involves neural networks that contain multitudes; these multitudes are known as models. As data enters into the model, the cells that make up the architecture of the model do computations that also apply a weight, and typically sum them up until it passes through an activation function. These cells have a variety of functions such as the convolution cell which slides a small matrix known as kernel or filter over the data, which it then performs a multiplication over the portion it filtered and sums it into a single value known as a feature map. Combining cells that have each of their unique mathematical properties and functionality can assist in plenty of issues, from simple classification of images to sentence creation.

The flow of how all these cells combine is known as architecture. While there are plenty of well-known models such as a Convolutional Neural Network (CNN), which consists of multiple Convolutional layers being pooled together as in 3.1, there are more such as a Recurrent Neural Network (CNN) and Long Short Term Memory (LSTM). More advanced architectures have been created to solve significantly more complex problems such as the U-net [9], a U-shaped CNN in which the image goes through a convolution layer downsampled and then again through a convolution network until it cannot be downsized any smaller it is then upsampled back to its original size, but at the start of each upsample the prior convolutional layer gets added alongside the downsample size as seen in 1.1. Transformers are another example of a complex architecture, where an image is broken up into patches and embedded with specific tokens.

Together these complex architectures can assist in predicting a series of images that alter concerning their time, time series. This is another example in which machine learning has accelerated a field, specifically atmospheric science, as models are fed time-based radar images from multiple sources such as the Geostationary Operational Environmental Satellite(GOES) or CSU-CHILL. These images are taken at specific times logged into a database and fed over multiple sequences to assist in short-term (nowcasting) and long-term (forecasting) weather predictions, This thesis hopes to combine the current knowledge of nowcasting for precipitation and utilize it for wildfire intensity and pathing prediction.

1.2 Literature Review

Prior to the use of deep learning for wildfire prediction is the use of High-Resolution Rapid Refresh [10]. A model with hourly data gathering and assimilation capabilities that monitors over the continental US and 3 hour interval of data assimilation for Alaska at a 3-km resolution. The HRRR is used for its ability to predict wildfire smoke plumes. The idea of utilizing deep learning

Unet-SaGRU.png

Figure 1.1: Unet Architecture with CNN+RNN. The image gets downsampled after each convolution until it cannot be downsampled further for feature extraction, then it is upsampled and convolved with the prior layer

for wildfire prediction is not recent, in Shmuel and Heifetz [11] they had applied four machine learning models on a 0.25° monthly resolution global dataset of wildfires. Shmuel and Heifetz utilized Random Forest, eXtreme Gradient Boosting, Multilayer perceptron, and logistic regression to evaluate wildfire susceptibility on a global scale. Shmuel and Heifetz found that the most important predictor was the regional wildfire history, they had identified that these had not been utilized in most studies. Shmuel and Heifetz's ML models had shown a lot of promise in the field, with the eXtreme Gradient Boosting being the highest-achieving model and their MLP network being the lowest. In the article Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model [7] the group focused on a much higher resolution to assist in short-term weather predictions. Utilizing a high-resolution radar echo map from the Hong Kong Observatory to be the input of the model, the data was approximately 6 minutes apart with the total input sequence being five frames to predict twenty frames. The team developed a new model known as the TrajGRU [12] and compared it against the known ConvLSTM model; the benefit of the TrajGRU model is that it actively learns the location variant structures. While the ConvLSTM has fixed hyper-parameters, the TrajGRU allows for the recurrent connections to be dynamically determined enhancing the ability to predict complex weather patterns. The encoder-forecasting structure is a convolutional layer to the TrajGRU cell and then downsampled before it upsampled again to its original size. When examining the MSE and MAE results of the TrajGRU performed substantially better compared to ConvGRU, showcasing it as a new standard for precipitation nowcasting. Another research that has expanded upon the TrajGRU paper is [13], in which the TrajGRU cell is edited to have a self-attention gate, this cell was designed to help improve prediction capabilities. The data being fed into the model were multiple hurricane events from 2015 to 2020 concentrated near the southeastern United States. The results showed the SaGRU model outperforms traditional models alongside that adding non-hurricane events improved the prediction abilities.

1.3 Contributions of the Present Work

The main contributions of this work include (a) multiple models based on sequence length for predicting the intensity and pathing of wildfires and (b) the performances of each model based on multiple metrics, focusing on Mean Squared Error (MSE) and Mean Absolute Error (MAE).

To achieve the desired results a focus on the Geostationary Operational Environmental Satellite (GOES), in which a detailed review can be found [14]; and is utilized to obtain radiance data. Once the radiance data is stacked correctly it is fed into multiple models to measure the varying results. The model contains a convolutional neural network which passes it on to the SaGRU cell then it goes to a pooling layer. Five inputs are given to the SaGRU model, and each model is altered to accommodate each model's predicted sequence length. The MSE and MAE per frame are gathered from this alongside the increase in MSE and MAE per frame. For the final metric, the individual samples are predicted as the wildfire is out of the input frames, then in the last frame of the input, before being at the start of the input sequence. Observations and conclusions are based on the predicted result metrics.

1.4 Thesis Organization

This thesis is organized as follows: In Chapter 2, a full review of the methodology utilized to obtain the GOES data, process the GOES data, and pipeline it into their respective models. A detailed explanation of the wavelength chosen, alongside the calculations utilized to convert radiance into Brightness Temperature (BT). Chapter 3 provides an in-depth explanation of the SaGRU cell and the equations that define it. It will also go into detail about the larger architecture and the loss functions utilized to gather the metrics. Chapter 4 presents the implementation results and analysis of the four SaGRU models, alongside the correlations found. Finally, in Chapter 5, conclusions on the proposed method are made and some potential future works are discussed.

Chapter 2

Review of Geostationary Operational Environmental Satellites Measurement

2.1 Introduction

Early fire detection has long relied on remote sensing technologies to help mitigate wildfire impacts in their early stages, this type of technology can include satellites such as Land Satellite (Landsat), Visible Infrared Imaging Radiometer Suite (VIIRS), and Geostationary Operation Environmental Satellites (GOES) [15]. Landsat has a spatial resolution of 15 to 60 meters [16] and, VIIRS has a spatial resolution of 370 to 740 meters [16]. Landsat-8 satellite images have been used to detect fires because of their spatial resolution, but the capabilities suffer as a result of the low temporal resolution. This would also be an issue with VIIRs which can detect more fire pixels due to their higher spatial resolution, but suffer from a very data gathering rate [17]. The GOES-R series focuses on improved rainfall detection, fire monitoring, intensity estimation, air quality warnings, and enhanced hurricane tracking, thanks to the Advanced Baseline Imager (ABI) [18]. Its geostationary position allows for a temporal resolution of 5 minutes, providing continuous coverage of active fires. The addition of the ABI enabled GOES to relay information to assist in weather tracking and monitor the vegetation health and fire temperature [18]. GOES is equipped to see 16 spectral bands that can be noted in 2.1, with band 7 and band 14 being ideal for spotting hotspots and smoke plumes.

This chapter focuses on Band 7 of GOES alongside the three specific fires that were chosen to be the dataset and which parts were designated for training or testing. Then, an in-depth examination of the pipeline was created specifically for data transfer to each model.

Table 2.1: Table of all ABI Bands

Category	Band	Description (wavelength)
Visible	1	Blue (0.47 μm)
	2	Red (0.64 μm)
	3	Near-IR (0.86 μm)
Near-Infrared	4	Veggie (1.37 μm)
	5	Cloud Particle Size (1.6 μm)
	6	Snow/Ice (2.2 μm)
Infrared	7	Shortwave Window (3.9 μm)
	8	Upper-Level Water Vapor (6.2 μm)
	9	Mid-Level Water Vapor (6.9 μm)
	10	Lower-Level Water Vapor (7.3 μm)
	11	Cloud-Top Phase (8.4 μm)
	12	Ozone (9.6 μm)
	13	Clean IR Longwave (10.3 μm)
	14	IR Longwave (11.2 μm)
	15	Dirty IR Longwave (12.3 μm)
	16	CO_2 Longwave (13.3 μm)

2.2 GOES Band 7

For wildfire detection, the ABI has a specific band that excels at hotspot detection known as Band 7. Band 7, also known as shortwave IR, typically has a wavelength of 3.9µm. Band 7 is the most efficient band to detect fire due to its thermal sensitivity, smaller thermal changes can easily show up on the band. In "Geostationary active fire products validation: GOES-17 ABI, GOES-16 ABI, and Himawari AHI" [19] the team compares LandSat active fire detection and GOES active fire detection against each other and found GOES band 7 ability to detect fires to be very accurate.

Band 7's multiple applications entail that it has the greatest bit depth [20]. While most bands contain a depth of 12 bits, Band 7 has a depth 14 bits. This is mostly because Band 7 must be able to detect cold features and hot features such as cloudtops and fires.

The sensitivity concerns the Planck equation which indicates that a band at 3.9µm will be much more sensitive to temperature than a band at a higher frequency.

2.2.1 Planck Equations

In 1900 Max Planck mathematically expressed spectral radiance as

$$B(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda k_B T}} - 1} \tag{2.1}$$

where B is the spectral radiance of a body to λ which is wavelength and T the absolute temperature. h is Plancks constant which could be defined as 6.62*10e-34 $Joule - Hertz^{-1}$. c is the speed of light in a vacuum which is equal to 299,792,458 m/s and k_B is known as the Boltzmann constant which is defined as.

$$k_B = \frac{PV}{TN} \tag{2.2}$$

where P is the pressure, V is the volume, N is the number of molecules in the gas, and T is the absolute temperature.

This derivation began when Planck was studying the classical theory of black body radiation which led to the Rayleigh-Jeans Law. The Rayleigh-Jeans law equation describes the spectral radiance of black body radiation for both frequency and wavelength.

$$B(\lambda, T) = \frac{2ck_B T}{\lambda^4} \tag{2.3}$$

However, this law had limitations as the law predicted that as the wavelength becomes shorter, or frequency becomes larger, the spectral radiance, $B(v, t)$ increases to infinity. This became known as the ”Ultraviolet Catastrophe", coined by Paul Ehrenfest [21]. Maxwell Planck resolved this by proposing that electromagnetic energy is emitted or absorbed in discrete quantities. This led to the Planck distribution law which explained spectral distribution across all frequencies without leading to infinity. This discovery was crucial in quantum mechanics as a way to accurately describe radiation behavior across all spectral regions.

2.2.2 GOES ABI and Resolution

The Advanced Baseline Imager (ABI) is a key component of the GOES-R series (GOES-16 through 19) and is an instrument critical for gathering data in weather forecasting, environmental monitoring, and climate research. As previously mentioned above it contains 16 spectral bands ranging from visible to infrared, which are used for varying purposes such as daytime observations, moisture and cloud observations, and thermal observations. GOES ABI also has enhanced spatial resolution, for the visible and near-infrared bands it offers a 0.5 km resolution, and for the infrared band it offers a 2km resolution [22]. While for infrared this resolution is good for capturing large objects, this does lend itself to missing smaller events. There is also improvement in the temporal resolution, as the ABI scans every 5 minutes, in specific regions, it can perform a rapid can every 30 seconds to 1 minute [18]. These bands become filtered into individual data products used by researchers which include Cloud Imagery, Sea Surface Temperature, Fire/Hot Spot Detection, Vegetation Health, and more. The data is stored in an online database such as an Amazon warehouse free for the public to obtain.

2.3 Data Selection for Wildfire Mapping

This section examines the data that was selected for the deep learning model. The criteria for data was that the fire must be large and alive through multiple days. The fire must be large enough to be picked up by the GOES ABI sensor, as it has a resolution of 2km; the fire must go for multiple days to have sufficient data for the model to study. The radiance data can also be converted to Brightness Temperature (BT); which is the radiation being emitted by a surface rather than the energy being emitted by a surface. Essentially BT measures temperature rather than energy. GOES also has a data product known as Fire Detection and Characteristics (FDC) which utilizes band 7 and the BT conversion to help detect heat signatures. Band 7 allows the FDC to estimate the size of the fire. As a result, the fires selected are more well-known fires that have caused great amounts of scarring in forests and property damage. The fires selected are the Sequoia Fires (2021), the Cal-Wood Fires (2020), and the Maui Fires (2023). These fires had gone on for several weeks before becoming extinguished and are easily picked up by the GOES satellite, and serve as the United States' most recent conflict in dealing with wildfires.

GOES-BAND_Comp.png

Figure 2.1: GOES ABI data starting with the top left being band 1 incrementing through each band until the bottom right where it is Band 16. Bands 1-6 are in the visible, and 7-16 are infrared

2.3.1 Sequoia Fires

The Sequoia Fires, also known as the KNP Complex Fires, began on September 9 2021 when a series of lightning strikes ignited the fires. These fires quickly spread and began united on September 17 2021 [23]. The forest had seen an abnormally dry year partially due to the ongoing drought, this caused a lot of dry vegetation which aided the fire in spreading more easily. Even live plants and trees had little moisture to prevent them from carrying the fire; as a result of the fire's spread, visibility suffered as aircraft could not assist in the ongoing fire suppression efforts. As a result of this fire thousands of mature sequoias were destroyed and approximately 88307 acres [24] were burned, the suppression efforts cost 170 million dollars [25]. The size, damage, and duration of the KNP fire make it the ideal case to use for wildfire prediction.

`2021_KNP_Complex_Fire_map_1.png`

Figure 2.2: Map of the burn scar from the Sequoia Fire (Red outline) [1]. This map is provided by the National Park Service https://www.nps.gov/seki/learn/nature/knp-complex-fire.htm

2.3.2 Calwood Fires

On August 13th, 2020 near Chambers Lake Colorado, a wildfire began from an unknown cause. The fire spread quickly and efforts to suppress via helicopters began. The fire was fueled by the hot and dry climate, steep terrain, and beetle-killed trees [26]. The suppression effort was struggling as the fire started to spread quickly due to the wind pushing it in every direction. The wind gusts would keep the fire alive, until late October when cooler temperatures and snowstorms helped extinguish the blaze. It was not until December 2nd, 2020 that the fire would be officially declared contained, torching 208,663 acres and costing 134 million dollars to fight [27].

`Cameronpeakfire.pdf`

Figure 2.3: Cameron Peak Burn Scar (Black Outline).https://ftp.wildfire.gov/public/incident_specific_data/rocky_mtn/2020/CameronPeak/IR/20201018/20201018_CameronPeak_IR_aerial_11x17.pdf. [2]

2.3.3 Maui Fires

On August 8th 2023, believed to be due to a downed power line, the Hawaii Fires began. It was relatively known that Hawaii had seen a decline in agriculture and nonnative species, and in 2023 Maui was experiencing a moderate drought [28]. Intense winds due to the nearby Hurricane Dora would begin to bring down utility poles which is believed to be the cause of the fire, meanwhile, in the small town of Lahaina, a brush fire would ignite and spread due to the intense winds [29]. As a result of the fire spreading quickly and closely to the town, the people were forced to self-evacuate resulting in a traffic jam. As a result of the fire, more than 100 people were dead and 5.5 billion dollars of damages are estimated to occur [30].

`Maui_Burnscar.jpeg`

Figure 2.4: Maui Burnscar (Red Outline) from the Pacific Disaster Center and can be found at https://www.mauicounty.gov/ImageRepository/Document?documentId=142365\cite{maui_report_2023}

2.3.4 Selections

Each wildfire event analyzed caused significant environmental damage and had profound impacts on surrounding communities. Due to their significance and size, they were selected and then randomly divided to be part of the training data and testing data for the model to assist in preventing a wildfire of this scale in the future. The data selected is split into a 70 percent training 15 percent testing and another 15 percent validation split. In band 7 of the GOES ABI, the data had been pulled from their respective servers and would go into the pipeline that feeds the model.

2.4 Conclusion

In this chapter, the GOES satellite and its capabilities were discussed, alongside the importance of the Planck equation and how it was derived. The detailed insight into the specifics of the GOES band 7 was utilized for wildfire prediction because of its ability to detect hotspots due to its sensitivity to thermal radiation, reduced atmospheric interference, and the ability to detect fires in the nighttime. The ABI allows the GOES series to obtain the data. The subject of data selection was covered in detail as well as the reason the respective fires were used. Sequoia due to its quick spread and damage, Calwood (Cameron peak) due to its proximity to the personal effects the authors have witnessed alongside the immense property damage that was accumulated and Maui for the human lives it had taken. The fires were also large and constantly moving which provided a lot of aid when assisting the model. In the next chapter, an in-depth explanation of SaGru will be examined alongside the architecture used, the data pipeline will be analyzed, from its ability to grab data from the AWS server to slicing the data to find the necessary hotspot, and the pre-processing required before being fed into one of the four models. Within the pipeline, normalization tactics will be explained alongside the conversion from radiance data to brightness temperature (BT).

Chapter 3

SaGRU and Deep Learning Model Architecture for Wildfire Prediction

3.1 Introduction

For a deep learning model to fully understand the context of wildfires, it must learn in the temporal and the spatial domain [31]. The temporal domain is responsible for the time dimension, and the spatial correlates to the occupying space. When the temporal and spatial dimensions conjoin it shows valuable data involving the motion of an object.

In the case of the wildfire data collected, the spatio-temporal axis will assist in showing the fire's intensity and movement. Multiple neural networks exist to obtain the temporal and spatial dimensions in the realm of deep learning and have been fused to allow for models to learn from both properties. The most commonly used is the Convolutional Neural Network (CNN) for the spatial dimension [32], and a Recurrent Neural Network (RNN) for the temporal [33]. When these neural networks are tied together to form an architecture, the model's ability drastically increases as it can study both dimensions in the correct context of each other [34].

In this chapter, a review of the CNN and RNN is covered; from the math that is utilized and their inherent structures. From there the concept of self-attention will be discussed to help introduce the cell that will be used in this study known as the Self-attention Gated Recurrent Unit (SaGRU) [13]. Alongside the SaGRU, a deep explanation of the data pipeline that will be supplying the model is given. Following the pipeline, a brief overview of the model structure will mentioned; then an overview of the main training approach(Adam) and the loss used (Mean Square Error).

3.2 Neural Networks

To understand the methods and concepts introduced in this chapter, a brief history of the neural network will be discussed. In 1943 McCulloch and Pitts published the paper "A logical calculus of the ideas immanent in nervous activity" [35], in which they proposed that artificial neurons could perform simple logical functions. This concept would be expanded upon in 1958 when Frank Rosenblatt developed the perceptron [36]; although, the neural network was very limited to only linear classification tasks. This did lead to some deterrence in the interest of neural networks until 1986 when Geoffry Hinton published "Learning representations by back-propagating errors" [37], which would calculate the gradient concerning the loss. During this time an interest in neural networks spiked and led to advancements in the CNN that were created by Kunihiko Fukushima, known as the neocognitron [38]. Yann LeCun would apply backpropagation to the CNN [39], fully learning the convolution kernel coefficients directly from the images: this would be the foundation of computer vision. During the same time as LeCun would be researching CNN, David Rumelhart along with Geoffrey Hinton and Ronald Williams would introduce the concept of RNNs [40], this introduced a hidden state in models to handle sequential data and capture information about prior inputs. Multiple advances have been made since these groundbreaking papers were written, a brief overview of the math that allows these units to function will be covered; the subsections will cover CNN, RNN, and Self-attention before moving forward to SaGRU.

3.2.1 Convolutional Neural Network

A convolutional neural network is composed of multiple convolutional layers, pooling layers, dense layers, and loss layers. This structure allows the model to excel when extracting features from an input image. As an image is passed into the model it will go through a process known as forward passing, a calculation and storing of variables from an output layer of the input data, it traverses through each neuron from the starting layer to the last layer.

As shown in 3.1 above the input image gets passed into the first convolutional layer, in which a defined kernel shape will convolve the input and pass it forward. The kernel shape is typically

CNN_Sagru.png

Figure 3.1: CNN sequence where the raw GOES data feeds through a convolution layer then to a pooling layer and once again through a convolution and pooling layer. The output of that is flattened and forwarded to a neural network of dense layers where an activation function known as Relu is used to assist in learning

defined based on the number of inputs, the height of the image, the width of the image, and the number of channels an image contains. The equation for the forward pass of the convolutional neural network can be defined as:

$$y_{ij} = (x * W)_{ij} + bF \tag{3.1}$$

for a simple single-input image. Where x is the feature map or image from the prior layer, W is the Convolutional filter that is scanned over the input image. The $*$ denotes the convolution between W and x, where W is convolved over x in the i and j position. The b represents the bias factor, it is added to the result of the convolution operation. y_{ij} is the output feature map. If multiple input channels appear it can be defined as

$$y^f_{ij} = \sum_{c=0}^{C-1} \sum_{m=0}^{k_H-1} \sum_{n=0}^{k_W-1} x_{c(i\cdot s+m)(j\cdot s+n)} \cdot W_{fcmn} + b_f \tag{3.2}$$

$x_{c(i\cdot s+m)(j\cdot s+n)}$ is the input image that the kernel is covering, where c is the channel $i \cdot s + m$ and $j \cdot s + n$ are the spatial coordinates of the input images: $i \cdot s$ and $j \cdot s$ indicate the position and m and n are the offsets. W_{fcmn} is the weight for the filter, channel, and positions. b_f is the bias term for the filter.$\sum_{c=0}^{C-1}$ indicates the summation over the input channels and $\sum_{m=0}^{k_H-1} \sum_{n=0}^{k_W-1}$ are the summations of the spatial dimension of the kernel, where H and W indicate height and width respectively. The output of the convolutional layer is known as a feature map, for the number of filters assigned to a convolutional layer will be the amount of feature maps found in the output of the layer. Usually, after the computation is complete the following layer is known as the pooling layer. The function of the pooling layer is to reduce the spatial of the feature map, this will reduce parameters in the network and make the representations not vary as much. When pooling the input is split into rectangles defined by a parameter, the type of pooling defines the output of the pooling layer. Max pooling outputs the maximum value of each region and is defined as

$$z^{(l)}_{b,(j_1,j_2)}(t) = \max_{i_1,i_2 \in \mathcal{S}} \{c^{(l)}_{b,(i_1,i_2)}(t)\}, \tag{3.3}$$

Average pooling is defined by the output being the average value in each region, it is mathematically described as:

$$z^{(l)}_{b,(j_1,j_2)}(t) = \frac{1}{|\mathcal{S}|} \sum_{i_1,i_2 \in \mathcal{S}} \sum c^{(l)}_{b,(i_1,i_2)}(t), \tag{3.4}$$

Once the pooling layer is completed it is not uncommon for another convolutional layer and pooling layer to be added for the model to learn deeper and understand the more subtle features that may be hidden. Often at the end, a dense layer may be added, this layer is connected to all the activations in the prior layer, which allows for a more global understanding of the image.

3.2.2 Recurrent Neural Networks

Another important neural network type is known as the Recurrent neural network (RNN), rather than being a uni-directional feedforward network; it is bi-directional. This allows the output layer to affect the input on the same node, this ability makes them very successful in speech recognition. The Self Attention GRU is an RNN itself utilizing bi-directionality to get a deeper understanding of the spatiotemporal effects of the trajectory of a wildfire. Plenty of layers utilize RNN, with one very commonly used known as Long Short-term Memory (LSTM). Containing "forget gates" LSTM typically prevents exploding or vanishing gradients, which are known for causing high instability in training models [41]. The errors tend to flow backward through a virtual layer which

Unfolded_RNN.png

Figure 3.2: An unfolding recurrent neural network that shows the input going into a hidden layer for undefined sequence length before going to the output

tends to be unlimited, this allows LSTM to learn events that occur over a long amount of discrete steps.

An application that benefits from LSTM specifically is natural language processing, where the network can learn the grammatical structure of a sentence. LSTM is not the only form of an RNN that is commonly used, a variant of one that was used for this subject matter is known as the Gated Recurrent Unit (GRU). A RNN with a gating mechanism that similarly to the LSTM contains a gate to input or forget features, with a key difference from LSTM being that the GRU does not contain the output gate. GRU showed similar performance to LSTM in tasks such as speech signal modeling and natural language processing [42], but because it lacks an output gate it contains fewer parameters. The formulas that define the GRU are important in understanding the following section; each gate contains its own formula in which The reset gate(r_t is responsible for the amount of information that will be forgotten and is defined as

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{3.5}$$

Where σ is defined as the sigmoid activation function, W_r is the weight matrix specifically for the reset gate. b_r is the bias term for the reset gate, and $h_{t-1}$is the hidden state from the previous step, LSTM.png

Figure 3.3: LSTM unit where the hidden layer from the prior cell and the input goes into 4 gates. Forget Gate multiplies with the Cells' prior layer and gets added with the input gate and the control gate before going through a Tanh activation and multiplying with the output gate.

and lastly, x_t is the input of the current step. Inside this equation, it should be noted that h_{t-1}, x_t is meant to denote a concatenation between h_{t-1} and x_t. The next gate to define is known as the Update Gate(z_t), which juxtaposed to the reset gate determines how much information is kept. It is also quite similarly defined to the reset gate as

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{3.6}$$

Where the W_z and b_z represent the weight matrix and the bias term for the update gate rather than the reset gate. Two more equations represent the hidden states in the GRU, one known as the candidate hidden state(h_t) which is the new hidden state that is computed from the reset gate defined as

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b) \tag{3.7}$$

GRU.png

Figure 3.4: A Gated Recurrent Cell where the Input goes through the forget gate(f_t) and gets multiplied to form the reset gate. It proceeds to the Update gate where it multiplies with Z_t, the update gate. It then gets added by the state candidate which is a product of the update gate and the input once it has gone through the Tanh function.

and the new hidden state(h_t) which is a combination of the previous hidden state and the candidate hidden state which is dictated by the update gate. The new hidden state can be mathematically defined by the formula

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{3.8}$$

These equations are necessary to understand how the future implementation of self-attention to the GRU will be utilized and how it benefits the performance of SaGRU in comparison to other RNNs. Before that, a brief discussion on the subject of Self-attention will be covered.

3.2.3 Self Attention

Introduced in the paper "Attention is all you need" [43], attention was developed to address the weakness of hidden layers in RNNs. It functions by calculating weights for each piece of the input, this is known as embedding. These calculations are done by transforming the inputs into three vectors: query, key, and value. In the example shown in 3.5, shows that the input is multiplied

`Self_attention.png`

Figure 3.5: Attention Chart. Where the image is split into 3 vectors: query, Key, and Value. The Query and Key become multiplied together and divided by the square root of 100 to give the softmax value. The softmax and value then multiply to give the context

by each of the vectors, this is displayed in the equations

$$Q = X * W_q \tag{3.9}$$

$$K = X * W_k \tag{3.10}$$

$$V = X * W_v \tag{3.11}$$

Where X is the input the query vector (Q) and the key vector(k) become a vector of soft weights but utilizing the formula

$$a = \text{softmax}\left(\frac{(xW_q) \cdot (XW_k)^\top}{\sqrt{100}}\right) \tag{3.12}$$

before finally multiplying it by the value vector(V) and outputting the context as

$$context = \text{softmax}\left(\frac{(xW_q) \cdot (XW_k)^\top}{\sqrt{100}}\right) * xW_v \tag{3.13}$$

These formulas allow attention to identify the highest correlations among the input, and capture the relevant information. Attention has become a staple in NLP and is showing promise in classification and regression applications.

3.3 Self Attention GRU

"A Self-attention based Deep Learning Model for Hurricane Nowcasting" [13] introduces the concept of creating a self-attention-based gated recurrent unit (SaGRU), with the context of obtaining as much spatial context for landfalling hurricanes. Combining self-attention with convolutional gated recurrent unit (ConvGRU), a variant of the GRU discussed above, it was designed to study a range of hurricane events from 2015 to 2020. The structure the authors propose is found in 3.6

`Self Attention.png`

Figure 3.6: Self Attention Cell. The input has features extracted by a 1x1 Convolution and 3 feature maps known as f_i,g_i, and v_i receive the map from those convolutions. The f_i and g_i convolve to create an attention map and the attention map convolutes with the v_i. This output feeds to another 1x1 convolution and gets added back with the input to create an output

Where the I_t is the feature map from the prior layer, the input is then split and goes through a similar path as the self-attention model shown in 3.6. It adds a 1x1 convolution before becoming transformed into the three feature maps: f_i,g_i, and v_i, and a 1x1 convolution right before $\hat{I}_t$. The gates themselves are described

$$\hat{X}_t = SA(X_t), \quad \hat{H}_{t-1} = SA(H_{t-1}) \tag{3.14}$$

$$Z_t = \sigma(W_{xz} * \hat{X}_t + W_{hz} * \hat{H}_{t-1} + b_z) \tag{3.15}$$

$$R_t = \sigma(W_{xr} * \hat{X}_t + W_{hr} * \hat{H}_{t-1} + b_r) \tag{3.16}$$

$$\hat{H}'_t = \tanh(W_{xh} * \hat{X}_t + R_t \circ (W_{hh} * \hat{H}_{t-1} + b_h)) \tag{3.17}$$

$$H_t = (1 - Z_t) \circ H'_t + Z_t \circ \hat{H}_{t-1} \tag{3.18}$$

where the SA is meant to be the self-attention module, the rest of the variables hold the same definition as the GRU gates defined in the above sections.

SAGRU_CELL.png

Figure 3.7: RNN cell (SaGRU). Where the current input, X_t and the prior hidden layer $h_t - 1$ are inputs and the update gate, Z_t and reset gate, R_t are used to control the hidden layer

The results of the design showed remarkable scoring when examining the utilized metrics known as Heidke skill score (HSS), critical success index (CSI), probability of detection (POD), and false alarm rate(FAR). Compared to another contemporary model, TrajGRU, SaGRU scored higher in the four metrics. The SaGRUs in the study were also trained for atmospheric science purposes, specifically using RADAR. While not the same as the GOES data used in this study, it is still similar enough that it became the primary RNN for this study.

3.4 Data pipeline for Wildfire collection

Once the GOES data was obtained from the Amazon Web Services (AWS) server, all the imagery was either of the West Hemisphere or the East Hemisphere of the Continental United States. The GOES data that had been obtained is simply an array, that can be visually examined using a colormap to see the difference in radiant flux emitted back into the satellite. The data would have been zoomed in from its 2500 by 1500 pixel count, to a 40 by 40 pixel size or an 80km by 80km resolution. This would allow for a proper examination of the fire; utilizing the known coordinates of the fire at their respective times allowed for easy identification of the fire and allowed for a quick slicing of the data.

Once the images were obtained it was important to keep the time series in proper sequence, otherwise the model could not easily understand the correlation between the events. Once the time series were in the correct order; the arrays were stacked together, joining the arrays together and adding to the depth dimension of the array. These arrays were prepared for three different models, each of the models would take the same input length but predict a different sequence length. Each of the models would take 5 inputs (25 minutes), and predict 15 figures (75 minutes), 20 figures (100 minutes), and 25 figures (125 minutes). Each of these models had identical structures and filters to achieve the best comparison. The 25-figure model was run twice to get a better understanding of how replicable the data was.

Input_Data.png

Figure 3.8: Raw input radiance data from GOES Band 7 over an area (80 km by 80 km) in California at 07:16 UTC, 20211509. The x- and y-axis indicate pixel index with a resolution of 2 km. The color bar shows the radiance intensity.

3.4.1 Normalization

The simplest way to describe Normalization is to define it as adjusting values to a different common scale. This is to ensure that across all the data they become on the same scale, there are numerous approaches to normalization depending on what the goal is. The formulas used for normalization can be utilized for mean and standard deviation, in this case, it may be best to use what is known as the "Standard Score"

$$z = \frac{x - \mu}{\sigma} \tag{3.19}$$

In this case, μ is the population's mean, and σ is the standard deviation. The standard score has multiple applications, the most known is the "Z-test", a statistical test in which the data is made into a normal distribution. It is often used to compare multiple means independent of the samples, an example would be a distribution of exam scores done by a class.

Another example is the "Min-Max feature scaling" also known as "Unity-based" normalization [44] technique. This utilizes the min and max values of the entire data set to restrict the data set to be in a range of 0 to 1.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3.20}$$

The minimum value and the maximum value must be found across the entire dataset to ensure all the values are between 0 and 1. Unity-based normalization is often utilized as a pre-processing step in data processing. It is most commonly used when the ranges in the raw data are widely varied and can create instability without normalization. In machine learning, this is an almost necessary step, as without unity-based normalization, the stability of the model can jeopardized.

$$T = \frac{\left(\frac{fk2}{\log\left(\frac{fk1}{\lambda L}+1\right)}\right) - bc1}{bc2} \tag{3.21}$$

$$fk1 = 2 * h * c^2 * cwn^3 \tag{3.22}$$

Norm_Fire.png

Figure 3.9: Normalized Data of Figure 3.8

$$fk2 = h * \frac{c}{b} * cwn \tag{3.23}$$

3.5 Model and Training of Wildfire Data

The subject of the topics covered in this chapter will all be utilized in the architecture that was used for this study, the U-net model. In combination with an encoder and decoder for downsampling and upsampling the inputs, and a convolution right before the first RNN and before the output of the cell.

This structure was first introduced for biomedical image segmentation, by Ronneberger, Fisher, and Brox in the paper "U-Net: Convolutional Networks for Biomedical Image Segmentation" [9]. The goal of this is to replace the pooling layer with the upsampling layer, this will increase the resolution back in the output rather than shrinking the input image. This study modifies the convolutional U-net in favor of using RNN where the convolutional network will be. The input will be fed into the first convolutional layer, before being passed onto the RNN, unfolded in 3.2, it will be downsampled and then once again passed into an RNN until it can no longer be downsampled. Then the RNN will then begin to upsample the image and pass the output of the upsample to the next RNN cell until the output shape is the same size as the original input shape. This allows the RNN, the SaGRU cell mentioned above, to gather more features and context per downsample before reconstructing the image at the output, due to the data being in a time series allows the RNN also gain an understanding of the temporal dimension as well. The RNN unrolling is based on the amount of sequences desired at the output, for this study three models with unique sequences were created with this architecture: 15 frames, 20 frames, and 25 frames.

The pipeline described in Chapter 2 would feed into the U-net SaGRU architecture and begin the training of the model. Each model would run 100 epochs with 2513 training samples and 536 validation samples to help correct the model. The model utilized the Adam [45] optimizer alongside the Mean Squared Error for the loss. The Adam optimizer was selected as it combines RMSProp with AdaGrad, it utilizes an adaptive learning rate for each parameter utilizing the formulas:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3.24}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{3.25}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{3.26}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{3.27}$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{3.28}$$

Where g_t is the gradient of the loss function, β_1 and β_2 is the decay rate, α is the learning rate, and ϵ is a small constant to prevent a division by zero. The adaptive learning rate allows the training of the model to be responsive and achieve an MSE value when examining the loss per epoch.

3.5.1 Evaluation Metrics

The metric used for loss was Mean Squared Error due to the nature of the problem, when examining the input the goal is one of regression not of classification. MSE allows the predicted values generated by the model to be compared to the ground truth or actual input of the model.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.29}$$

The result is that MSE is differentiable which pairs well with an optimizer such as Adam, allowing for the lowest score possible. Since MSE averages the squared errors it allows for large errors to be penalized more heavily, while this does make it more susceptible to outliers, the data was vetted to remove any extraordinarily high values.

Mean Absolute Error was another metric used for evaluation, it is another form of regression testing. MAE takes the absolute difference between the predictions and the ground truth.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.30}$$

MAE is quite similar to MSE but differs because rather than squaring the difference of the predicted and ground truth; it takes the absolute value of it. This means that the MAE is less sensitive to outliers since the error is not squared, which in turn prevents large outliers from affecting the overall score disproportionately. Variations of MAE and MSE are also used to evaluate each model, known as Normalized Mean Absolute Error (NMAE) and Normalized Mean Squared Error (NME).

$$\text{NMAE} = \frac{MAE}{NormalizationFactor} \tag{3.31}$$

$$\text{NME} = \frac{MSE}{NormalizationFactor} \tag{3.32}$$

The normalization factor is the mean of the observed values or the range of the values. This allows different models to be comapred regardless of dataset or scale. This is useful since

3.6 Conclusion

In this chapter, the history of neural networks was overviewed and Convolutional Neural Networks and Recurrent Neural Networks were discussed in depth, while also going into detail about self-attention and the three vectors that assist in the context computation. An explanation of how the Self Attention Gated Recurrent Unit came to be based on the neural network overview, and why it was utilized for this study. A detailed explanation of the data pipeline was also discussed, with Normalization and Radiance to Brightness being key talking points. An introduction to U-net and the alteration from convolution to RNN was discussed, alongside an explanation of the metric utilized, MSE.

Encoder_Decoder.png

Figure 3.10: Encoder and Decoder structure. The input is fed through a convolution layer to the RNN, SaGRU, where it is downsampled and fed into the next RNN. The RNN feeds to the decoder where it is Upsampled to reconstruct the output

Chapter 4

Results and Analysis

4.1 Introduction

In this chapter, the results of each model will be discussed alongside comparisons between each model as well. Before that, a quick overview of the prior chapters will be discussed to ensure that any possible confusion of the dataset, pipeline, model, and training is clarified. The results will focus on each model's performance, and then a comparison of models. There will be a greater focus on the 25-sequence model specifically both variants (known as 1.00 and 1.01), with metrics to display the findings. Afterward, a discussion will be posted on the data, the observations noted, and an explanation of what the analysis may mean.

4.2 Review of information

4.2.1 Data and pipeline

The fires that were the focus of the study are the Sequoia, Calwood, and Maui fires; the main reason for their selection was the scale of the fires, it would be easily seen on GOES band 7 and the fires were quite mobile. All the fires had been obtained using the Amazon Web Service NOAA server, from GOES 16, GOES 17, and GOES 18. Band 7 was strictly chosen because of its ability to identify hotspots and its near continuous data gathering, the 5-minute intervals that it logs the data were sufficient in creating a time series. When all the data is gathered, it is sliced into a 40x40 array that highlights the fire but retains the background as well. Once the data is sliced, it is organized by time and then stacked, this stack contains a depth of 30 creating a sample shape of 30x40x40 where 30 is the depth, 40 is the height, and 40 is the width.

The first 5 of the stacked array are the inputs, the rest is meant to be the ground truth for the model to train on. Once all the data was stacked it was split into three categories randomly: train, validation, and test. The train data always contained 2513 samples and the test and validation contained 536 samples, this was a 70, 15, 15 split of the total data. The data generator would identify the max and min values across all the data and normalize the data to be between 0 and 1 and then the generator would convert the data from radiance to brightness temperatures utilizing the Planck function.

`PIPELINE.PNG`

Figure 4.1: Data Pipeline where multiple labels are created and added with the 5 frame input. They are then trained and then show the prediction frames.

4.2.2 Model and Training for Wildfire Prediction

The model architecture is defined as a modified U-net structure, where the CNN is replaced by the SaGRU. The SaGRU being a combination of the Gated Recurrent Unit and Self-attention, allows the model to understand the spatio-temporal of the problem quite well. Three distinct models were created to see the effects of the sequence length on the model, their respective lengths were 15 frames, 20 frames, and 25 frames. The 25-frame model was run twice to study the effects of weight variation in the model and is denoted as 1.00 and 1.01. It is important that models 1.00 and 1.01 are the same model just with different weights, and outside of the sequence length of the

RNN there are no changes in the model. Because the size of each frame, 40x40, is quite small this allowed the model to have a small amount of filters assisting in having a very small model size.

All the models were trained at exactly 100 epochs and utilized the Adam optimizer. Once the model had completed training, the output was de-normalized to obtain visual data, and the metrics were gathered and compared.

4.3 Results

In this section, the results of the study will be covered and the proceeding will be the observations. Starting from the 15-frame model a quick overview of the metrics will spoken about, then the 20-frame, to covering both 25-frame models (1.00 and 1.01).

4.3.1 15 frame model

The first model created and tested was the 15-frame model. This model would take in 5 input frames and output 15 frames, 75 minutes in advance. The data fed into this model was the same stacked data, but the generator would only stack 20 images rather than the 30 used for the full 25 models. The output of the data was put through a metric function that gave Mean Squared Error (MSE), Mean Absolute Error (MAE), Normalized Mean Error (NME), and Normalized Mean Absolute Error (NMAE) for various values known as thresholds. The threshold values used here are 0.01, 0.05, and 0.1.

15_MSE.png

(a) 15 Frame MSE

15_NME.png

(b) 15 Frame NME

15_MAE.png

(c) 15 Frame MAE

15_NMAE.png

(d) 15 Frame NMAE

Figure 4.2: Metrics for 15 frames focusing on each of the 4 evaluation methods. MSE is on the top left where an increase is constantly shown as the sequence continues. NME on the top right to show a greater increase in loss as the sequence continues. MAE at the bottom left to show a slow increase as the sequence continues. NMAE at the bottom right displays a slow increase as the sequence continues. This is all done for three thresholds 0.01,0.05, and 0.1.

The trend showed a gradual increase in MSE and MAE as the sequence increased, that steady increase is shown to be .008 for all three thresholds in the MSE and MAE

15_metrics_per_frame.png

Figure 4.3: 15 Frame model MSE between frames. It shows that the frame 1 to 2 contributes to the highest amount of MSE increase while it decreases as the sequence length continues showing less variance as the model continues in the sequence.

Table 4.1: Evaluation of 15 Frame of MSE

	0.01	0.05	0.1
1 to 2	0.03415066	0.03419311	0.03432882
2 to 3	0.02109414	0.0211224	0.02118482
3 to 4	0.01561971	0.01563166	0.01565099
4 to 5	0.01278207	0.01279735	0.01280753
5 to 6	0.01265573	0.01266436	0.0126871
6 to 7	0.01249818	0.01250691	0.01252517
7 to 8	0.01216902	0.01217951	0.01221722
8 to 9	0.01266751	0.01268125	0.01271032
9 to 10	0.0118374	0.01184766	0.0118776
10 to 11	0.01132933	0.01133711	0.01136192
11 to 12	0.01165471	0.01166427	0.01167607
12 to 13	0.01190205	0.01190946	0.01193025
13 to 14	0.01082566	0.01083007	0.01084555
14 to 15	0.01023297	0.01023752	0.01025479
Average:	0.01409460933	0.01677350933	0.02013721

4.3 and 4.4 shows a sharp increase in MSE and MAE for frames 1 to 2, but steady out almost immediately afterward.

`15_per_frame_MAE.png`

Figure 4.4: 15 Frame model MAE between frames. Frames 1-2 show a sharp increase in MAE but as the sequence continues the increase in MAE plateaus

Table 4.2: Evaluation of 15 Frame Model MAE

	0.01	0.05	0.1
1 to 2	0.02416492	0.02415892	0.02430421
2 to 3	0.01338934	0.01340105	0.01350584
3 to 4	0.00849483	0.00848928	0.00850492
4 to 5	0.00639045	0.00640897	0.00642572
5 to 6	0.00601302	0.00602709	0.00611685
6 to 7	0.00613504	0.00615181	0.00623007
7 to 8	0.00617793	0.00620857	0.00634702
8 to 9	0.00613405	0.00617293	0.00629259
9 to 10	0.00528944	0.00601084	0.00612108
10 to 11	0.00646228	0.00579323	0.00588631
11 to 12	0.00556134	0.00559029	0.0056424
12 to 13	0.00530005	0.0053128	0.00539624
13 to 14	0.00501637	0.005025	0.00508843
14 to 15	0.00477269	0.00478219	0.00484507
Average:	0.007807267857	0.007823783571	0.007907625

4.3.2 20 frame model

The second model was the 20-frame model, a gradual increase from the 15-frame model in the case of any instability. In this model 5 input frames were given and the model output 20 frames, one hundred seconds in advance. The data fed was 25 stacked images for one sample, to account for the additional 5 frames added from the 15 frame model. The same set that had been split and shuffled was used in the 20-frame model to limit any potential discrepancies in performance. The expectation for the 20-frame model is that it would perform worse than the 15-frame model as it is expected to predict an additional five frames, but the results showed a different result.

`20_,MSE.png`

(a) 20 Frame MSE

`20_NME.png`

(b) 20 Frame NME

`20_MAE.png`

(c) 20 Frame MAE

`20_NMAE.png`

(d) 20 Frame NMAE

Figure 4.5: Metrics for 20 frames. It shows thresholds from The 20 Frame model contains MSE which increases per frame as the sequence increases.The NME shows a sharp decrease as the sequence continues. The MAE at the bottom left indicates the a steady increase in loss. The NMAE at the bottom right also indicates a constant increase in loss

The 20-frame model starts at a lower MSE and MAE than the 15-frame model. but similarly to the 15 frame, it shows a steady growth in MSE and MAE. Approximately .0086 for MSE and MAE the loss decreased compared to the 15 frame as it increased per frame approximately .0035. Both the 15 frame and 20 frame share similar curves with both having a sharp spike at the first frame, similarly they both see a roughly .8 increase in MSE.

20_frame_increase_per_frameMSE.png

Figure 4.6: MSE increase per frame of 20 Frame Model. The most MSE lost begins between frames 1-2 and then seem to plateau to a steady increase of loss.

Table 4.3: Evaluation of 20 Frame Model MSE

	0.01	0.05	0.1
1 to 2	0.02613756	0.02617592	0.02633985
2 to 3	0.01149981	0.0115175	0.01152579
3 to 4	0.00857185	0.00856438	0.00859378
4 to 5	0.00804402	0.0080503	0.00806173
5 to 6	0.0083901	0.00839773	0.00839241
6 to 7	0.00806678	0.00806515	0.00805274
7 to 8	0.00770172	0.00769622	0.00774634
8 to 9	0.00822694	0.0082317	0.00826001
9 to 10	0.00741398	0.00743184	0.00743098
10 to 11	0.00713144	0.00714013	0.0071392
11 to 12	0.0077434	0.00773589	0.00775272
12 to 13	0.0082078	0.00821847	0.00822971
13 to 14	0.00738197	0.00738702	0.00740842
14 to 15	0.00698696	0.00697914	0.00700805
15 to 16	0.00696981	0.00696888	0.0070015
16 to 17	0.00672807	0.00674606	0.00674364
17 to 18	0.00661871	0.00663462	0.00664139
18 to 19	0.00576597	0.00576691	0.00577127
19 to 20	0.00608247	0.00607973	0.00608013
Average:	0.008614176842	0.008620399474	0.008641034737

20_frame_increase_per_frameMAE.png

Figure 4.7: MAE increase per frame of 20 Frame Model. MAE increase per frame shows a high increase at 1-2 before immediately stagnating between frames 3-4 and onwards.

Table 4.4: Evaluation of MAE for 20 frame model

	0.01	0.05	0.1
1 to 2	0.01901572	0.01901216	0.01912
2 to 3	0.00604414	0.00606168	0.00606781
3 to 4	0.00275465	0.00275092	0.00281344
4 to 5	0.00256144	0.00257675	0.00261608
5 to 6	0.00257571	0.00258992	0.0025872
6 to 7	0.00263033	0.00262916	0.00260959
7 to 8	0.00261588	0.00260547	0.00271102
8 to 9	0.00258802	0.00259536	0.00266367
9 to 10	0.00249341	0.00251253	0.00250816
10 to 11	0.00243147	0.00244796	0.00243173
11 to 12	0.00247254	0.00246417	0.00248671
12 to 13	0.00249108	0.00250857	0.00251825
13 to 14	0.00242023	0.00243185	0.00247494
14 to 15	0.00239171	0.00237797	0.00245092
15 to 16	0.00237731	0.00237209	0.00242957
16 to 17	0.00231529	0.00233607	0.00235761
17 to 18	0.00230804	0.00233331	0.00234789
18 to 19	0.00228812	0.00228203	0.00229464
19 to 20	0.00230822	0.00229471	0.00230056
Average:	0.003530700526	0.003535930526	0.003567883684

4.3.3 25 frame model 1.00

In the following subsections two 25 frame models will be explained. The reason for the two is to see how weight variation plays a part in the stability and accuracy of the model. 1.00 refers to the first model tested.

25_MSE.png

(a) 25 Frame MSE

25_NME.png

(b) 25 Frame NME

25_MAE.png

(c) 25 Frame MAE

25_NMAE.png

(d) 25 Frame NMAE

Figure 4.8: Metrics for 25 1.00 frames with three thresholds 0.01, 0.05, 0.1. The MSE on the top left indicates a steady growth of loss as the sequence continues. The NME on the top right shows a steady increase. The MAE has a steady increase as the sequence continues. The NMAE is a very gradual increase as well.

The starting MSE and MAE are noticeably higher than the prior two models, 15 and 20 frame models, but still show similar steady growth.

25_100_MSEperframe.png

Figure 4.9: MSE increase per frame 25 Frame Model (1.00)

. Frames 1-2 show a sharp increase in MSE and as the sequence continues it begins to decline though very gradually. Each threshold remains quite similar in increase.

Table 4.5: Evaluation of MSE for Model 1.00

	0.01	0.05	0.1
1 to 2	0.01750645	0.01751359	0.01754084
2 to 3	0.01213902	0.01214022	0.01215242
3 to 4	0.0118493	0.01185165	0.01186127
4 to 5	0.01042143	0.01042294	0.0104279
5 to 6	0.01030878	0.01031019	0.01031377
6 to 7	0.00958728	0.00958828	0.00958969
7 to 8	0.00901022	0.00901062	0.00901038
8 to 9	0.00952327	0.0095238	0.00952435
9 to 10	0.00880811	0.0088089	0.00881084
10 to 11	0.00852441	0.00852518	0.0085269
11 to 12	0.00909616	0.0090971	0.00909846
12 to 13	0.00954486	0.00954622	0.00954803
13 to 14	0.00869338	0.00869446	0.0086958
14 to 15	0.00832083	0.00832187	0.00832325
15 to 16	0.00832755	0.00832883	0.00833006
16 to 17	0.00811276	0.00811393	0.00811489
17 to 18	0.0079903	0.0079917	0.00799235
18 to 19	0.00712574	0.00712767	0.00712858
19 to 20	0.00746135	0.00746324	0.00746425
20 to 21	0.00692554	0.0069278	0.00692931
21 to 22	0.00677951	0.0067815	0.00678355
22 to 23	0.00683488	0.00683658	0.00683851
23 to 24	0.00676893	0.00677039	0.00677124
24 to 25	0.00605152	0.00605297	0.00605351
Average:	0.0089879825	0.008989567917	0.008992922917

25_100_MAEperframe.png

Figure 4.10: MAE increase per frame 25 Frame Model (1.00)

. Threshold 0.01 shows a very high increase compared to the 0.05 and 0.1. The patterns are the same where frames 1-2 shows the sharpest increase before stabilizing.

Table 4.6: Evaluation of MAE for Model 1.00

	0.01	0.05	0.1
1 to 2	0.01750645	0.00848839	0.00866866
2 to 3	0.01213902	0.00536637	0.00541232
3 to 4	0.0118493	0.00479009	0.00484604
4 to 5	0.01042143	0.00394212	0.0039634
5 to 6	0.01030878	0.00360309	0.00362577
6 to 7	0.00958728	0.00335319	0.00337244
7 to 8	0.00901022	0.00320521	0.00320333
8 to 9	0.00952327	0.0032197	0.00322622
9 to 10	0.00880811	0.00322915	0.00325196
10 to 11	0.00852441	0.00324557	0.00326781
11 to 12	0.00909616	0.00324221	0.003257
12 to 13	0.00954486	0.00320665	0.00322664
13 to 14	0.00869332	0.00314942	0.00315737
14 to 15	0.00832089	0.00311645	0.00312703
15 to 16	0.00832755	0.00309973	0.00310834
16 to 17	0.00811276	0.00309825	0.00310458
17 to 18	0.0079903	0.0030901	0.00309165
18 to 19	0.00712574	0.00304457	0.00304862
19 to 20	0.00746135	0.00305134	0.00305784
20 to 21	0.00692554	0.00303814	0.00304907
21 to 22	0.00677951	0.00305439	0.00306507
22 to 23	0.00683488	0.00308351	0.00309091
23 to 24	0.00676893	0.00310585	0.00310287
24 to 25	0.00605152	0.00307905	0.00307638
Average:	0.0089879825	0.0035792725	0.003600055

The MSE model sees an average .009 increase per threshold which is quite higher than the other two models, but in the MAE it shows the highest increase at the 0.001 threshold and seems to steady out at higher thresholds. Unlike the other models, in 4.9 the MSE tends to keep declining rather than simply plateau after the first frame transition to the second frame.

4.3.4 25 frame model 1.01

The second 25-frame model tested will be noted as 1.01 to indicate a difference from the first 25-frame trained model. The same model with the same data was tested, the only difference was the weights initialized which were randomized by the optimizer.

25_1.01_MSE.png

25_1.01_NME.png

(a) 25 Frame MSE for model 1.01

(b) 25 Frame NME for model 1.01

25_1.01_MAE.png

25_1.01_NMAE.png

(c) 25 Frame MAE

(d) 25 Frame NMAE

Figure 4.11: Metrics for 25 1.01 frames. The three thresholds 0.01, 0.05, and 0.1 all increase at roughly the same place.

25_101_MSEperframe.png

Figure 4.12: MSE increase for 25 Frame (1.01). The thresholds 0.01, 0.05, and 0.1 increase at roughly the same rate as each other, but have an erratic increase as the sequence goes on.

Table 4.7: Evaluation of MSE for Model 1.01

	0.01	0.05	0.1
1 to 2	0.01008268	0.0101232	0.01037525
2 to 3	0.00761522	0.00761752	0.00771379
3 to 4	0.00951607	0.00955062	0.00968999
4 to 5	0.01028908	0.01031758	0.01043085
5 to 6	0.01116639	0.01118365	0.01128449
6 to 7	0.01101841	0.0110418	0.01112452
7 to 8	0.01070071	0.01071875	0.01079433
8 to 9	0.01148521	0.01151509	0.01157491
9 to 10	0.01114465	0.01117041	0.0112493
10 to 11	0.01144073	0.01146798	0.01155575
11 to 12	0.01240204	0.01244158	0.01251274
12 to 13	0.01333	0.01336093	0.01343718
13 to 14	0.01264163	0.01267227	0.01273533
14 to 15	0.0125431	0.01257386	0.01261103
15 to 16	0.01257965	0.0126032	0.01263931
16 to 17	0.01245498	0.01248129	0.01252614
17 to 18	0.01234906	0.01237353	0.01241143
18 to 19	0.01153498	0.0115559	0.01159763
19 to 20	0.01179198	0.01180875	0.01185314
20 to 21	0.01120158	0.01121462	0.0112521
21 to 22	0.01093987	0.01095994	0.01099034
22 to 23	0.01095179	0.01096544	0.01100638
23 to 24	0.01084152	0.01085333	0.01088892
24 to 25	0.01000699	0.01001903	0.0100459
Average:	0.01125118	0.01127459458	0.01134586458

`25_101_MAEperframe.png`

Figure 4.13: MAE increase for 25 Frame (1.01). The MAE does not show a sharp increase in frames 1-2, but increases sharply at frames 4-5 and continues to increase until frame 17-18 when it begins to decrease.

Table 4.8: Evaluation of MAE for 1.01 frame

	0.01	0.05	0.1
1 to 2	0.00202902	0.00205318	0.00220088
2 to 3	0.00188978	0.00189131	0.00193065
3 to 4	0.00382427	0.003842	0.00393658
4 to 5	0.00508167	0.0050902	0.00515855
5 to 6	0.00576386	0.0057608	0.00582669
6 to 7	0.00596154	0.00596192	0.00602048
7 to 8	0.00597092	0.00595774	0.00600876
8 to 9	0.00618169	0.00619019	0.00621118
9 to 10	0.00654284	0.00654517	0.00658933
10 to 11	0.00709967	0.00710278	0.00717064
11 to 12	0.00758773	0.00761407	0.00767153
12 to 13	0.0078003	0.00781316	0.00789446
13 to 14	0.00785531	0.00787575	0.00792053
14 to 15	0.00792132	0.00794115	0.00794584
15 to 16	0.00795763	0.00796314	0.00796861
16 to 17	0.00786516	0.00788214	0.00790569
17 to 18	0.00776323	0.00777665	0.00778559
18 to 19	0.00755784	0.0075713	0.00760446
19 to 20	0.00745264	0.00746126	0.00750932
20 to 21	0.00728236	0.00728039	0.00732427
21 to 22	0.0071597	0.0071685	0.00721777
22 to 23	0.00705782	0.00705159	0.00714044
23 to 24	0.00698095	0.00696972	0.00703642
24 to 25	0.00682794	0.00682287	0.00686627
Average:	0.006475632917	0.006482790833	0.006535205833

Unlike the prior models, the increase in MSE and MAE does not sharply decline but makes a similar shape to a bell curve, this is the cause for the sharp rise in MSE and MAE.

4.3.5 Review

In the section, all the graphs and tables were shown for the performance of the individual models. The models were shown per frame and the graphs show the increase in loss from frame to frame. The analysis and observation section will primarily focus on the 25-frame model but will still mention the prior models.

4.4 Analysis and Observations

From the data of the prior section, it is quite clear that as the sequence increases so does the loss, in every model it can be seen that the MSE and MAE both increase gradually at a constant rate. However, it is quite interesting when comparing the MSE of the model to how they perform. It

15v20_MSE.png

Figure 4.14: MSE difference between 15 frame Model and 20 Frame Model. The difference between model 15 and model 20 when examining each frame increases as the sequence increases. The thresholds remain constant

15_20_MAE.png

Figure 4.15: MAE difference between 15 frame Model and 20 Frame Model. Simiarly to its MSE counterpart it had a small difference when examining each frame, but the difference increased as the sequence length increased

is important to examine the first frames as it shows that the MSE and MAE in the 20-frame model were not higher than in the 15-frame model as was to be expected. From comparing these two models it can be seen that the loss in the 20-frame model does not drastically increase to match the loss in the 15-frame, as can be examined by looking at the trend of the difference increases between the two models. This suggests that the starting MSE may be a very important performance factor. Examining how the 20-frame model does against the 25-frame model (1.00) and then comparing the 20-frame model to the other 25-frame model (1.01) may give a strong indication.

MSE_diff_20v25_1.00.png

(a) 20 vs 25 (1.00) MSE

MSE_diff_20v25_1.01.png

(b) 20 vs 25 (1.01) MSE

MAE_diff_20v25_1.00.png

(c) 20 vs 25 (1.00) MAE

MAE_diff_20v25_1.01.png

(d) 20 vs 25 (1.01) MAE

Figure 4.16: MSE and MAE difference per frame comparing the 20 frame model and both 25 frame models (1.00 and 1.01

.

Comparing the 20 models to the first 25-frame model, 1.00, it can be shown a drastic difference in compared to the 15-frame model. The 1.00 model shows a large increase at the start and a slow gradual increase in the difference between MSE meanwhile MAE shows at the .01 threshold a sharp increase and stable difference in the .05 and .1 thresholds. This is most likely due to how the MAE treats each threshold as equal, there is likely more variance in the .01 threshold resulting in a compounding MAE error. When examining the 20-frame model against the other 25-frame model, 1.01, a very different result is seen. The MSE difference starts to decrease from the 20-frame model, but as it hits frame 15 the losses match and the 25-frame model begins to increase. This is also similar to the MAE, which shows an increase in frame 17. This does indicate that the models are affected by the sequence length, though this doesn't inherently start at the first frame. The differences between the two 25-frame models were quite interesting, and to understand these behaviors it will be necessary to compare the two models.

MSE_DIFF_1.00v1.01.png

MAE_DIFF_1.00v1.01.png

(a) 1.00 vs 1.01 MSE

(b) 1.00 vs 1.01 MAE

Figure 4.17: Both 25 frame models compared in MSE and MAE. MSE and MAE begin with 1.01 providing significantly sharp difference but as the sequence increases the difference in MSE and MAE becomes smaller.

While an inference could have been made that model 1.01 would have a lower MSE than 1.00, what was found was that around frame 8 begins the decline. The difference between these models shrinks and almost seemingly with enough of a sequence length they will share the same loss. This is a strong indicator of how the initial weights of the model can play a core part when it comes to the early prediction of the sequence but as towards the end of the sequence, the loss will constantly compound at an increased rate. To fully realize these results an actual test data was used from the Maui fire in 2023, in which each of the models were given the same inputs, and their metrics were recorded.

4.4.1 Nowcasting Metrics for Wildfire Prediction

The data used to gather the metrics was August 8th, 2023 at 10:46 (UTC) through August 8th, 2023 at 12:01 (UTC), it is important to note that each date and time is one sample and contains 5 inputs and 25 output frames. These times were selected because at 10:46 am no fire could be seen through the GOES data and the fire would become visible at roughly 11:21 am where a hotspot could be noticed. Starting at 10:46 and shifting the input by 5 minutes allowed metrics to be obtained based on when the fire became part of the input. It can be seen that when the fire truly becomes visible it is around 11:51 am and by 12:01 pm the wildfire had been considered out of control. It was expected that model 1.01 would predict better than the 1.00 model, but it was interesting to note that both models had a slight bell curve shape and both saw a slight increase in

Maui_GT.png

Figure 4.18: Maui Fire starting at 10:46 UTC on August 8th, 2023 and continuing until 13:11 UTC on August 8th, 2023

MSEofnowcast.png

Figure 4.19: Model 1.00 and Model 1.01 prediction power when comparing MSE. Model 1.01 has lower MSE than Model 1.00 but both create the same sort of curve. Time at the bottom is the dates in UTC

the final sample. With model 1.01 being a higher-performing model, it was important to see how it performed visually as well. With the MSE being very slightly different per sample, the visuals give a better indication of the model's ability to predict the wildfire. In this case, only a few times were selected: 10:46 am, 11:21 am, and 12:01 pm, this was because 10:46 has no fire involved in the input frame so it would be interesting to see if it could predict the fire, 11:21 am is the highest MSE value seen in 4.19, and 12:01 pm is where the fire is constantly ablaze. When the fire input is shown at the start then the model can correctly predict that the fire will occur and rise, even if it is 2 frames prior.

For 12:01 UTC the fire was constantly in the input therefore it did correctly predict the intensity, but lost some of the background context. This may be the reason for the increased MSE on the last frame as the predicted struggled to successfully correct the background following the fire.

1121_GT.png

(a) 11:21 am Ground Truth

1121_Pred.png

(b) 11:21 am (UTC) Prediction

Figure 4.20: 11:21 (UTC) Ground Truth against Prediction. The intensity of the fire predicted is not the same size as the ground truth at 11:21 on August 11th, 2023

1201_GT.png

Figure 4.21: Ground Truth at August 8th 12:01 UTC

1201_Pred.png

Figure 4.22: Prediction at August 8th 12:01 UTC

4.5 Conclusion

In this chapter, a review of the pipeline, model, and training was overviewed to assist in clarifying any questions the reader may have had, and then began to go over the model's results. Each model's MSE and MAE were shown with any observations that may have impacted that score. It was noted that the main observation was that the sequence length did not have a profound effect on the overall ability of the model's prediction but rather towards the end of the sequence. The analysis focused on the results and expanded on the reasons why they had occurred, focusing on the differences between the 20-frame model and the 25-frame models. This led to a deeper comparison between the two 25-frame models, where model 1.01 was found to predict more accurately the 1.00 which was attributed to how the weights were distributed from the very start. However, it was also noted that the model would soon catch up as the MSE and MAE would rise towards the end of the sequence. One more comparison was done between 1.00 and 1.01 which would be the MSE and MAE before the Maui Fires and, it was noted that they both had similar shapes when examining the MSE and MAE but 1.00 was quite higher. A visual inspection was focused on 1.01, examining specific times to see how the model predicted.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, the groundwork to utilize deep learning for wildfires is proposed with the hope and intention of building upon this foundation. It was covering over the GOES data, which was utilized for hotspot detection by slicing the data and stacking the data to be in the correct time series. Then using min-max normalization techniques to help the stability of the model, before transforming the base radiance to brightness temperature. It was fed to multiple models, each using the Self-attention GRU and metrics were taken to see the model's performance and visual data to understand how the model is learning the inputs quickly. It was optimized using Adam [45] and ran for 100 epochs to ensure that each model would converge. The focus was at first seeing the impact of sequence length on models, with the expectation that the overall model would be deterred. This was not the case though, it was easily seen that the loss would spike towards the last predicted frames compared to the start. Some models started with a lower MSE/MAE than its predecessor, on a deeper analysis one model was reran with varied results.

To summarize, it was found that the sequence length did not have any effect on the early predictions of a model but did have a large effect on the later predictions. A comparison of the 25-frame model, 1.00 and 1.01, showed that weights had a significant part in the results. When comparing the data visually though, the model could only predict a hotspot accurately if it was already present in the input. This meant that it has no true ability to identify small hotspots, and can only be of use when it becomes identifiable with GOES band 7.

This is a foundation for wildfire path prediction, many issues must be addressed to get the full potential of this research.

5.2 Future Work

While this thesis provided a solid framework for wildfire predictions, there are issues to focus on to help improve the capabilities of deep learning for wildfire predictions. These are

- Resolution increase
- Wind, Vegetation, and Historical Data Implementation
- Pathing Prediction
- Pipeline Implementation

Resolution Increase: GOES is currently limited by a 2km resolution, this resolution results in quite undefined pathing of the fire and incorrect location of the fire. The Generative Adversarial Network(GAN) [46] utilizes two models, a generator and a discriminator, the generator creates its images and feeds it to the discriminator. The discriminator takes both the generator image and the real image and tries to tell the two apart. This could use the GOES input data to create a higher resolution of the fire being depicted. This could also correct the location at which the fire is occurring by the GAN seeing the difference in latitude and longitude from the higher resolution fire and the GOES data. This method to some extent has been tested by White and Ebert-Uphoff [47] where the authors take a 2km resolution, similar to that used in Band 7, and super-resolves the image to become .5km.

Wind, Vegetation, and Historical Data Implementation: While GOES band-7 is quite good at detecting hotspots, it does lack some crucial information. It does not account for wind, the vegetation the fire is on, aspect, elevation, etc. While it is possible for a model to identify these variables based on the GOES input data, it could have assisted in the prediction of the models. Mainly due to how wildfires spread quickly with the help of gusts, dry vegetation can carry the fire because it catches easily, and the aspect since fire tends to go upward. These known variables could help the model predict related pixels to fire more accurately.

Pathing Prediction Utilizing GANs for super-resolution can also be used to assist in predicting the path of the fire. By using a Geographic Information System it is possible to have a GAN create a time series of the progression of the fire, this could utilize land cover, elevation, orthoimages, structures, and transportation to create a detailed event of a wildfire progression since its start and the variables that lead to the containment of the fire. This would allow a model to predict what possibilities a fire could take and could be used in combination with GOES data to create a 2-hour long prediction of the pathing fire.

Pipeline Implementation With the model trained, it is now deployable but needs to be implemented in a product, at work at Colorado State University (CSU) can be done for the algorithm to be implemented. A collaboration with a government agency such as the National Aeronautics and Space Administration (NASA) or National Oceanic and Atmospheric Association (NOAA) can lead to an implementation of this product to help predict wildfire intensity and pathing via a supercomputer or alternative means.

Bibliography

[1] Penitentes, “2021 knp complex fire map,” 2021. [Online]. Available: https://commons.wikimedia.org/wiki/File:2021_KNP_Complex_Fire_map_1.png

[2] Unknown, “Burn area map of cpf,” 2020. [Online]. Available: https://ftp.wildfire.gov/public/incident_specific_data/rocky_mtn/2020/CameronPeak/IR/20201018/20201018_CameronPeak_IR_aerial_11x17.pdf

[3] Colorado Division of Fire Prevention and Control, “Community fire preparation,” 2024. [Online]. Available: https://dfpc.colorado.gov/communityfireprep

[4] K. Gajendiran, S. Kandasamy, and M. Narayanan, “Influences of wildfire on the forest ecosystem and climate change: A comprehensive study,” *Environmental Research*, vol. 240, p. 117537, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0013935123023411

[5] J. R. E. Campbell, M. B. Baker, P. F. Ffolliott, F. R. Larson, and C. C. Avery, “Research paper,” USDA Forest Service, Rocky Mountain Forest and Range Experimental Station, Fort Collins, CO, Research Paper RM-191, 1977, u.S. Department of Agriculture, Forest Service, Rocky Mountain Forest and Range Experimental Station.

[6] D. Lüthi, M. Le Floch, B. Bereiter, T. Blunier, J.-M. Barnola, U. Siegenthaler, D. Raynaud, J. Jouzel, H. Fischer, K. Kawamura, and T. F. Stocker, “CO2 record from the EPICA Dome C 1999 (EDC99) ice core (Antarctica) covering 650 to 800 kyr BP measured at the University of Bern, Switzerland,” 2008, in supplement to: Lüthi, D et al. (2008): High-resolution carbon dioxide concentration record 650,000 - 800,000 years before present. Nature, 453, 379-382, https://doi.org/10.1038/nature06949. [Online]. Available: https://doi.org/10.1594/PANGAEA.710901

[7] S. Yao, H. Chen, E. J. Thompson, and R. Cifelli, “An improved deep learning model for high-impact weather nowcasting,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 7400–7413, 2022.

[8] M. N and K. R, “Detection of hazardous gases using deep learning on benchmark dataset,” pp. 1–5, 12 2023.

[9] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[10] D. C. Dowell, C. R. Alexander, E. P. James, S. S. Weygandt, S. G. Benjamin, G. S. Manikin, B. T. Blake, J. M. Brown, J. B. Olson, M. Hu, T. G. Smirnova, T. Ladwig, J. S. Kenyon, R. Ahmadov, D. D. Turner, J. D. Duda, and T. I. Alcott, “The high-resolution rapid refresh (hrrr): An hourly updating convection-allowing forecast model. part i: Motivation and system description,” *Weather and Forecasting*, vol. 37, no. 8, pp. 1371 – 1395, 2022. [Online]. Available: https://journals.ametsoc.org/view/journals/wefo/37/8/WAF-D-21-0151.1.xml

[11] A. Shmuel and E. Heifetz, “Global wildfire susceptibility mapping based on machine learning models,” *Forests*, vol. 13, no. 7, 2022. [Online]. Available: https://www.mdpi.com/1999-4907/13/7/1050

[12] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, “Deep learning for precipitation nowcasting: A benchmark and a new model,” 2017. [Online]. Available: https://arxiv.org/abs/1706.03458

[13] S. Yao, H. Chen, and V. Chandrasekar, “A self-attention based deep learning model for hurricane nowcasting,” in *2023 United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)*, 2023, pp. 292–293.

[14] GOES-R Program Office, “Beginner’s guide to goes-r series data,” 2020. [Online]. Available: https://www.goes-r.gov/downloads/resources/documents/Beginners_Guide_to_GOES-R_Series_Data.pdf

[15] B. Pradhan, M. D. H. Suliman, and M. Awang, “Forest fire susceptibility and risk mapping using remote sensing and geographical information systems (gis),” *Disaster Prevention and Management*, vol. 16, pp. 344–352, 06 2007.

[16] P. Oliva and W. Schroeder, “Assessment of viirs 375m active fire detection product for direct burned area mapping,” *Remote Sensing of Environment*, vol. 160, pp. 144–155, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003442571500019X

[17] F. Li, X. Zhang, and S. Kondragunta, “Biomass burning in africa: An investigation of fire radiative power missed by modis using the 375 m viirs active fire product,” *Remote Sensing*, vol. 12, no. 10, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/10/1561

[18] GOES-R Program Office, “Advanced baseline imager (abi) fact sheet,” 2017. [Online]. Available: https://www.goes-r.gov/education/docs/Factsheet_ABI.pdf

[19] J. Hall, W. Schroeder, K. Rishmawi, M. Wooster, C. Schmidt, C. Huang, I. Csiszar, and L. Giglio, “Geostationary active fire products validation: Goes-17 abi, goes-16 abi, and himawari ahi,” *International Journal of Remote Sensing*, vol. 44, no. 10, pp. 3174–3193, 2023.

[20] Cooperative Institute for Meteorological Satellite Studies (CIMSS), “Abi band 7 – shortwave window: Factsheet,” 2018. [Online]. Available: https://cimss.ssec.wisc.edu/goes/abi/factsheets/ABIBand7ShortwaveWindowFINAL.pdf

[21] P. Ehrenfest, “In which features of the light quantum hypothesis does thermal radiation play an essential role?” *Annalen der Physik*, vol. 36, pp. 91–118, 1911, translation of the original title: "Welche Züge der Lichtquantenhypothese spielen in der Theorie der Wärmestrahlung eine wesentliche Rolle?".

[22] J. Hall, R. Zhang, W. Schroeder, C. Huang, and L. Giglio, “Validation of goes-16 abi and msg seviri active fire products,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 83, p. 101928, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0303243419306336

[23] National Park Service, “Knp complex fire,” 2021. [Online]. Available: https://www.nps.gov/seki/learn/nature/knp-complex-fire.htm

[24] ——, “Two sequoia groves and a wildfire,” 2023. [Online]. Available: https://www.nps.gov/seki/two-sequoia-groves-and-a-wildfire.htm

[25] National Interagency Fire Center (NIFC), “National interagency coordination center (nicc) annual report 2021,” 2021. [Online]. Available: https://www.nifc.gov/sites/default/files/NICC/2-Predictive%20Services/Intelligence/Annual%20Reports/2021/annual_report_0.pdf

[26] N. Swayze, C. Tsz Hin Choi, G. Knowlton, and J. Kilsauskaite, “Investigation of optical diagnostics for supercooled large droplet effects on airfoil performance,” NASA Langley Research Center, Tech. Rep. 20210014944, 2021, accessed: 2024-10-18. [Online]. Available: https://ntrs.nasa.gov/citations/20210014944

[27] “National interagency fire center 2020 annual report,” 2020, accessed: 2024-10-13. [Online]. Available: https://www.nifc.gov/sites/default/files/NICC/2-Predictive%20Services/Intelligence/Annual%20Reports/2020/annual_report_0.pdf

[28] “Report on wildfire prevention and cost recovery on maui - part 1: Report and exhibits a-b,” Maui County, Tech. Rep., 2020, accessed: 2024-10-18. [Online]. Available: https://www.mauicounty.gov/DocumentCenter/View/129493/Report-on-Wildfire-Prevention--Cost-Recovery-on-Maui---Part-1-Report--Exhibits-A-B-33-MB

[29] National Weather Service, Honolulu Office, “Wind summary for august 9, 2023,” 2023, accessed: 2024-10-18. [Online]. Available: https://www.weather.gov/hfo/windSummary20230809

[30] National Centers for Environmental Information (NCEI), “Billion-dollar weather and climate disasters: Events,” 2023, accessed: 2024-10-18. [Online]. Available: https://www.ncei.noaa.gov/access/billions/events.pdf

[31] K. C. Short, "A spatial database of wildfires in the united states, 1992-2011," *Earth System Science Data*, vol. 6, no. 1, pp. 1–27, 2014. [Online]. Available: https://essd.copernicus.org/articles/6/1/2014/

[32] A. J. Alzubaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, 2021. [Online]. Available: https://doi.org/10.1186/s40537-021-00444-8

[33] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014. [Online]. Available: https://arxiv.org/abs/1409.3215

[34] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf

[35] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[36] F. Rosenblatt, "The perceptron - a perceiving and recognizing automaton," Cornell Aeronautical Laboratory, Ithaca, New York, Tech. Rep. 85-460-1, January 1957.

[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, D. E. Rumelhart and J. L. Mcclelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 318–362.

[38] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[39] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[41] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[42] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, “Light gated recurrent units for speech recognition,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, p. 92–102, Apr. 2018. [Online]. Available: http://dx.doi.org/10.1109/TETCI.2017.2762739

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[44] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015. [Online]. Available: https://arxiv.org/abs/1502.03167

[45] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[46] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-realistic single image super-resolution using a generative adversarial network,” 2017. [Online]. Available: https://arxiv.org/abs/1609.04802

[47] C. H. White, I. Ebert-Uphoff, J. M. Haynes, and Y.-J. Noh, “Superresolution of goes-16 abi bands to a common high resolution with a convolutional neural network,” *Artificial Intelligence for the Earth Systems*, vol. 3, no. 2, p. e230065, 2024. [Online]. Available: https://journals.ametsoc.org/view/journals/aies/3/2/AIES-D-23-0065.1.xml