3-2023

Inducing Sparsity Within High-dimensional Remote Sensing Modalities for Lightning Prediction

Grace E. Metzgar

INDUCING SPARSITY WITHIN HIGH-DIMENSIONAL REMOTE SENSING MODALITIES FOR LIGHTNING PREDICTION

THESIS

Grace E. Metzgar, Second Lieutenant, USAF

AFIT-ENS-MS-23-M-146

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-23-M-146

INDUCING SPARSITY WITHIN HIGH-DIMENSIONAL REMOTE SENSING MODALITIES FOR LIGHTNING PREDICTION

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Data Science

Grace E. Metzgar, B.S.O.R.

Second Lieutenant, USAF

March 23, 2023

AFIT-ENS-MS-23-M-146

INDUCING SPARSITY WITHIN HIGH-DIMENSIONAL REMOTE SENSING MODALITIES FOR LIGHTNING PREDICTION

THESIS

Grace E. Metzgar, B.S.O.R.
Second Lieutenant, USAF

Committee Membership:

Nathan Gaw, Ph.D
Chair

Bruce Cox, Ph.D
Member

Mostafa Reisi, Ph.D
Member

AFIT-ENS-MS-23-M-146

Abstract

The uncertainty of lightning constantly threatens many weather-sensitive fields where the slightest presence of lightning can endanger valuable personnel and assets. The consequences of delaying operations have incited the research of methods that can accurately predict the location of future lightning strikes from the current weather conditions. High-dimensional remote sensing modalities contain information capable of detecting significant patterns and intensities within storms that could indicate the presence of lightning. This thesis induces sparsity into convolutional neural networks (CNNs) and remote sensing modalities through a combination of regularization and tensor decomposition techniques to call attention to sparse features that are most indicative of lightning activity. The developed models produce accurate predictions of the general pattern of true lightning strikes at lower time lags. The results demonstrate the potential of using CNNs in combination with sparse methods that focus on important features for the prediction of close-range lightning activity.

Table of Contents

Page

List of Figures

List of Tables

INDUCING SPARSITY WITHIN HIGH-DIMENSIONAL REMOTE SENSING MODALITIES FOR LIGHTNING PREDICTION

I. Introduction

Lightning is a common hazard that poses significant threats to the general public and a variety of weather-sensitive mission sets. The slightest presence of lightning greatly endangers personnel, valuable assets, and mission success. Additionally, mission cancellations due to weather waste precious time and resources, which can impact subsequent operations. However, lightning strikes are among the extreme weather phenomena that are incredibly difficult to forecast due to their sporadic nature. As a result, methods that demonstrate the capability to accurately predict future lightning strikes are highly desired throughout many fields.

Space launches are among the operations most heavily affected by the threat of lightning. Nearby and direct lightning strikes can disrupt important communication and navigation systems that are essential to the mission and even destroy the launch vehicle altogether. As a result, aerospace engineers must either harden the launch system to withstand a lightning strike or avoid the hazards by launching only under safe flying conditions. Hardening the vehicle against lightning requires a large amount of time, money, and heavy materials that could potentially affect the performance of the system. Similarly, avoiding lightning storms requires extensive research into weather patterns of the surrounding area and the capability to accurately forecast potential storms. It also requires appropriate safety measures to be in place and considerable flexibility when a threat becomes present which creates obstacles to successful mission execution [1].

To mitigate the risk of lightning, the American space program established the Lightning Flight Commit Criteria (LFCC) to apply to all launch sites within the United States. The LFCC are a set of weather constraints that must be satisfied before the launch of a space vehicle is permitted. The LFCC has evolved as knowledge about lightning behavior and techniques for forecasting lightning has improved over time [1]. These improvements have increased the safety and availability of launches appealing to a variety of stakeholders.

Lightning greatly affects aviation operations as well. A severe storm near an airport can disrupt all ground operations and potentially delay or cancel all inbound and outbound flights. Delays can be extremely costly and troublesome for airlines and passengers. According to the Federal Aviation Administration, a delay costs an airline as much as $4,500 per hour, with the value of passenger time costing as much as $63 per hour [2]. Additionally, the sudden presence of lightning en route could require an aircraft to change their flight path or divert to an alternate airport. The United States Air Force requires weather personnel to routinely monitor weather along planned routes to alert pilots and other decision-makers of potential hazards, such as lightning [3]. Weather personnel need sophisticated technology and tools to promptly monitor and forecast important weather conditions to support these time-sensitive missions.

The ability to predict the patterns of lightning in an accurate and efficient manner can have immense impacts on aviation and many other operations that are sensitive to weather. Methods capable of determining if a lightning strike will occur at a specific time and location based on observed weather conditions could drastically improve mission safety and the criteria that must be satisfied to execute a specific mission, such as the LFCC. Additionally, methods that are able to make predictions efficiently can enable quicker adaptability to dynamic weather conditions.

This research utilizes different facets of weather data to predict future lightning strikes and thus, provides advanced warning. Weather data is collected through a variety of high-dimensional (HD) sensing modalities to describe their various aspects. Some examples of these modalities include vertically integrated liquid (VIL), infrared (IR), and other satellite imagery. These images contain useful information that can identify certain features and patterns within weather events. Many studies identify VIL as the sensing modality most indicative of lightning behavior. VIL estimates the total amount of water contained within a vertical column of the atmosphere by measuring the reflectivity of the air and is often used to identify features within hail and thunderstorms. Shafer et al. observe a severe storm system in Oklahoma and find, as the concentration of VIL increases so does the density of lightning strikes [4]. Holleman develops a hail and thunderstorm detection product and concludes larger values of VIL are highly correlated with severe thunderstorms [5]. Lu et al. discover VIL and echo top radar data have the greatest impact on their lightning monitoring residual network model [6]. In addition to VIL, 10.7 μm infrared windows measuring brightness temperatures of cloud tops have also proved useful in detecting the behavior and intensity of thunderstorms. Molinie and Jacobson observe cloud top brightness and cloud-to-ground lightning strike densities over the continental United States and find, the likelihood of strike occurrence is greater when brightness temperatures are coldest [7]. As a result, this research utilizes information from VIL and 10.7 μm brightness temperature images as a predictor for lightning.

The spatio-temporal information within these sensing modalities are vital to detect significant patterns and intensities within storms that could indicate an imminent danger, such as lightning. However, since the features that contain information indicative of lightning within the images are sparse, there may be an abundance of data that is not relevant to the prediction of lightning strikes. This thesis aims to introduce sparsity into convolutional neural networks (CNN) through regularization to call attention to important features and subsequently reduce the complexity of the models. CNNs are well-suited for tasks such as predicting weather patterns, where the location of different weather features is important as they take into account the spatial relationships between different pixels within an image. This research also introduces sparsity into the remote sensing images by identifying sparse regions before model training through robust tensor decomposition (RTD). RTD is a tensor decomposition method that extracts anomalies within HD data in an efficient manner. The sparse images produced by RTD are then input into the CNN to focus the model on the most important features for prediction. Adding sparsity into the CNN and into the remote sensing modalities calls attention to important regions of the images to inform a more accurate prediction of the location of future lightning strikes.

The material within this thesis is organized as follows. Chapter II provides a literature review of similar methodologies and applications. Chapter III details the sparse CNN, the formulation of RTD, the different models tested, and the data set that is applied. Chapter IV presents the results from the applied methodology to the data set. Finally, Chapter V offers relevant insights from the results and provides recommendations to continue this research.

II. Literature Review

This chapter explores the wide range of literature related to this research. The first section provides a brief overview of convolutional neural networks (CNN) and their various applications in weather prediction problems. The second section discusses various sparse methods to increase attention and improve performance within deep learning models.

2.1 Convolutional Neural Networks

A CNN is a deep learning technique that is commonly used to find patterns within images for use in tasks such as image classification, object detection, and image segmentation. CNNs are comprised of a combination of convolutional layers, pooling layers, and fully connected layers which are designed to learn the spatial hierarchies of features through backpropagation. Convolutional layers perform feature extraction by applying a set of kernels to the input data. A dot product between the entries of the kernel and the input data at each location is calculated to create feature maps which highlight specific features or patterns in the input data. The pooling layers perform downsampling by dividing the feature maps into a set of non-overlapping windows and applying a summary function (e.g. maximum, minimum, average) to each window. This reduces the dimensionality of the data and makes the model more robust to small translations and distortions in the data. The fully connected layers interpret the extracted features and make a prediction for the final output, such as classification [8].

A CNN's ability to learn from spatial features is particularly useful in weather applications where the position and arrangement of different weather features, such as clouds and precipitation, are important for determining certain weather phenomena.

Remote sensing images contain an abundance of information such as cloud formation patterns, temperature, and the distribution of lightning strikes over a region. A CNN trained on images of weather events can learn to recognize complex patterns and relationships to predict future weather events based on the current conditions of the system.

2.1.1 Weather and Lightning Application

Deep learning methods are a popular choice for weather forecasting applications because of their ability to process and learn complex patterns within large amounts of data. CNNs and similar methods that consider the spatial information within weather imagery are especially useful in predicting future weather conditions where such information is critical to the prediction. There is an abundance of research available dedicated to using deep learning to detect and understand patterns in various weather phenomena. Racah et al. propose a convolutional autoencoder architecture for semi-supervised bounding box prediction to detect specific extreme weather events within simulated images of the Earth's atmosphere [9]. Han et al. employ various CNN models to extract spatial information from 3D doppler radar data to perform convective storm nowcasting [10].

More specific to the application in this thesis, many studies exist that explore how deep learning can be used to predict the occurrence of future lightning activity. Guo et al. develop the Convolutional LSTM Lightning Forecast Net (CLSTM-LFN), a nowcasting model which merges historical lightning occurrence frequency and physical variables to improve upon previous video prediction techniques [11]. Geng et al. presents a convolutional autoencoder called LightNet, which uses spatial features from simulated weather data to forecast the location of lightning strikes [12]. Lu et al. build a CNN that utilizes features from multiple weather radar modalities to predict the occurrence of a single lightning strike [6]. These studies find that the spatial information within weather sensing imagery has substantial predictive power over the patterns and occurrence of future weather phenomena, such as lightning.

2.2 Sparse Methods

Sparse methods help simplify the data and focus models on relevant information for a variety of machine learning techniques. Introducing sparsity into CNNs is an effective way to reduce model complexity and highlight the features of the input data that are most important to the prediction task. When the input data has sparse features, it is important to bring attention to the regions containing the most valuable information to form more accurate predictions.

2.2.1 Regularization

One way to add sparsity in a CNN is to add a regularization term to the input convolutional layer. Weight regularization introduces sparsity by adding a penalty term to the loss function that encourages the model weights to be as small as possible. This penalty term is typically a function of the L1 or L2 norm and serves to constrain the model and prevent the weights from becoming too large. Adding the penalty term encourages the optimization process to find a solution with smaller weights, which can result in a sparser model, where more of the weights will be exactly or near zero. Regularizers generalize the model and make it less prone to overfitting, helping to improve prediction performance. Regularization methods to induce sparsity within deep learning models is a heavily researched topic. Yoon et al. propose a combined group and exclusive LASSO for deep neural networks to enforce sparsity within the network's weights, finding that the sparse model improves performance and reduces computational time [13]. Additionally, Wu et al. apply L1/2 regularization for the specification of hidden layers in feed forward neural networks and conclude that L1/2 regularization helps improve model generalization [14].

L1 regularization (LASSO) adds a penalty term to the loss function that is the sum of the absolute values of the weights in the network multiplied by a lambda parameter which controls the strength of the regularization. This allows some of the unimportant weights to shrink to zero, effectively selecting a subset of the most important features. This can reduce variance in the predicted values and improve the interoperability of the model. However, LASSO can also produce models that are too sparse, where only a small number of weights are non-zero, reducing the predictive power of the model [15].

L2 regularization (Ridge Regression) adds a penalty term to the loss function that is the sum of the squared weights in the network multiplied by a lambda parameter. This allows the weights to shrink towards zero, but not as aggressively as LASSO. This means that the model is encouraged to set the weights of features to small, non-zero values, rather than zero. Like LASSO, this helps to minimize the impact of some features included in the model and diminish multicollinearity, but is less likely to reduce the predictive power of the model compared to LASSO. However, since Ridge Regression does not shrink weights to exactly zero, the models are not sparse and not as effective in feature selection as models trained with LASSO, which might be a disadvantage in certain applications [16].

Lastly, Elastic Net extends LASSO by adding the Ridge Regression penalty term. Elastic Net produces models much more robust to multicollinearity, as it groups and shrinks highly correlated variables together and includes or excludes all of them from the model. In contrast, LASSO randomly chooses one of the variables to include in the model and removes the others, potentially making faulty assumptions about which variables will produce the best model. However, Elastic Net can be more computationally expensive to implement because it includes calculations for both the LASSO and Ridge Regression penalty terms [17].

2.2.2 Tensor Decomposition

Another way to add sparsity in a model is to extract sparse anomalies within the input data itself. There is a substantial amount of literature dedicated to the diagnosis of anomalies within high-dimensional (HD) data. Tensor decomposition is an overarching method widely used to accomplish such a task by decomposing tensors into their background and sparse components. Tensor decomposition builds upon traditional matrix decomposition by extending it to process higher-order tensors. This section describes the various matrix and tensor decomposition methods used to identify sparse anomalies within images and data streams.

Some of the most popular methods for detecting anomalies within HD data are based upon principal component analysis (PCA) due to its scalability and effective dimensionality reduction abilities. However, the performance of PCA is limited due to its sensitivity to outlying or corrupted observations. Robust PCA (RPCA) is a dimension-reduction method that finds the best low-rank representation in large, high-dimensional data, while also being robust to large errors and outliers. Candès et al. propose an RPCA method known as principal component pursuit (PCP) that decomposes a noisy matrix $\mathbf{B}$ into its low-rank component $\mathbf{X}$ and sparse components $\mathbf{E}$ through a convex optimization problem [18]:

$$\begin{aligned} &\min_{\mathbf{X},\mathbf{E}} ||\mathbf{X}||_* + \lambda||\mathbf{E}||_1 \\ &\text{s.t.} \quad \mathbf{B} = \mathbf{X} + \mathbf{E} \end{aligned}$$

Here, $\mathbf{X}$ can be viewed as the image background and $\mathbf{E}$ its sparse anomalies. λ

controls the strength of the sparsity enforced by the L1 norm. RPCA can be solved efficiently through a variety of algorithms, such as accelerated proximal gradient (APG) and augmented Langrange multiplier (ALM) algorithms for use in various applications. Wright et al. apply RPCA to successfully remove shadows from face images and to separate the background from the activity in airport surveillance images [19].

RPCA can be scaled to process tensors in a similar method known as robust tensor decomposition (RTD) which decomposes tensors into their low-rank and sparse components using an extension of the convex optimization problem used in RPCA. RTD considers the low-rank structure of every matrix within the tensor which influences what outliers the model will detect [20]. RTD can also be solved efficiently using an algorithm known as alternating direction method of multipliers (ADMM) that solves complex problems by breaking them into smaller subproblems and updates the relevant variables in an alternating process. The formulation of RTD is futher explained in Chapter III of this thesis. Hu and Work apply RTD to speed maps of traffic in Nashville and find the algorithm can accurately detect outliers to indicate a car crash [21].

Similar methods have been developed to detect anomalies within HD data through tensor decomposition. Smooth Sparse Decomposition (SSD) uses a similar framework to RPCA but operates on images that have a smooth background. As a result, SSD decomposes an image into three components: its smooth background, sparse anomalies, and random noise. This is accomplished through a penalized regression model that enforces background smoothness and anomaly sparsity through penalty terms added to the loss function. Yan et al. find SSD reduces computation time and improves detection accuracy within stress maps due to its ability to decompose an image in a single step [22].

Spatio-temporal smooth sparse decomposition (ST-SSD) extends SSD by introducing a temporal component to process HD data streams with a time-varying mean. ST-SSD can process entire data streams and output how the smooth background, sparse anomalies, and random noise change over time. It extends the SSD penalized regression model by adding temporal parameters to model the temporal trends of HD data streams in addition to the spatial structures. ST-SSD identifies anomalies when an abrupt change is detected in the functional mean of the smooth background, as opposed to RTD which extracts anomalies within every image based its low-rank structure. It is typically used in applications where there are few sparse anomalies in only a small proportion of the frames within a large data stream. Yan et al. applies ST-SSD to streams of solar images to detect solar flares [23]. ST-SSD improved the detection of solar flares due to its ability to model the spatial features of the images, as well as detect temporal changes that reveal sparse anomalies.

For the purposes of this problem, RTD is chosen as the most appropriate method to extract anomalies from the HD remote sensing modalities. RTD identifies sparse features within each image of a data stream instead of only when an abrupt change occurs as in ST-SSD. This is especially important when predicting lightning activity far in advance, when more information about the storm is required to model the future patterns of the event rather than just a singular anomalous occurrence in a particular time frame. It is important to display sparse features within every frame of the data stream so each image contains relevant data the CNN can learn from to make an accurate prediction of the location of lightning.

III. Methodology

3.1 Problem Statement

Remote sensing modalities of weather events contain important spatio-temporal information crucial in identifying behaviors within storms that contribute to the occurrence of extreme weather phenomena. These high-dimensional (HD) images include complex patterns and sparse features that can be difficult for deep learning methods to learn from and use for prediction. This thesis aims to induce sparsity into convolutional neural networks (CNN) through regularization techniques to focus on sparse features that are most indicative of lightning activity. Sparsity is also introduced into the remote sensing images themselves through a tensor decomposition method known as Robust Tensor Decomposition (RTD). RTD extracts sparse anomalies within the HD data streams which are then input into the CNN to reduce the impact of potentially unimportant features. In addition, adding sparsity into the models will help determine the predictive power of multiple remote sensing modalities and their sparse features in relation to lightning activity.

3.2 Data Set

This research applies its methodology on the Storm EVent ImagRy (SEVIR) data set. Collaborators at MIT Lincoln Laboratories created the SEVIR data set to provide researchers with the appropriate tools to develop innovative and insightful models on weather patterns [24]. The SEVIR data set contains spatially and temporally aligned image sequences for thousands of weather events associated with three years of thunderstorm events in the continental United States. Each event includes a four-hour length sequence of images in five-minute frames that cover 384×384 km patches for each of the five recorded sensing modalities: 0.6 μm visible satellite imagery

(VIS), 6.9 μm channel infrared satellite imagery (IR 6.9), 10.7 μm channel infrared satellite imagery (IR 10.7), vertically integrated liquid (VIL), and total lightning flashes collected by the GOES-16 geostationary lightning mapper. Intercloud and cloud to ground lightning strikes are recorded continuously throughout the four-hour duration of an event with a time and location. The strikes are converted into images of five-minute increments where a single pixel is an integer value indicating the amount of strikes experienced within that pixel during that period. Additionally, events are separated into storm events and random events. Storm events were deliberately selected to target severe storm events, while random events were randomly chosen across the U.S.

Since this research focuses on the VIL, IR 10.7, and lightning modalities, only the 12,872 events that contain all three modalities are considered. The five modalities for these events are stored in sequences of 49 images spanning the four-hour time period where each image represents a snapshot in time every 5 minutes . The VIL, IR 10.7, and lightning images are 384 $\times$ 384 pixels, 192 $\times$ 192 pixels, 48 $\times$ 48 pixels, respectively. Figure 1 displays an example of a single time step image for IR 10.7, VIL, and lightning modalities. In the lightning image, the brightness of a pixel indicates the amount of strikes, where black indicates zero strikes.

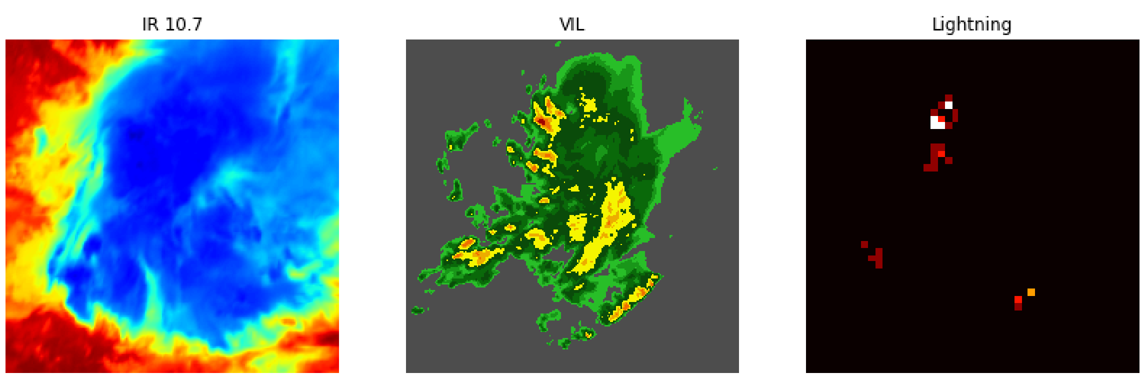


Figure 1: Weather Sensing Modalities

3.2.1 Data Pre-Processing

Since the SEVIR data set contains tens of thousands of events with multiple time steps each, the number of events input into the model must be limited to reduce the computational complexity. Additionally, the data set is extremely imbalanced, with most events containing little to no lightning activity. Therefore, in order to capture the patterns indicative of lightning, events exhibiting high lightning activity must be injected into the training data to give the model enough information to make an informed prediction. As a result, 100 events are randomly chosen from the entire set of events and another 56 are randomly chosen from the events that record the most lightning strikes within their four-hour time period. Each event contains sequences of 49 images for a total of 7,644 images in each modality used in this analysis. Furthermore, 80% of these events are utilized for training, 10% for validation, and 10% for testing.

In addition, the lightning images are transformed to fit the purpose of this project. Each image is transformed into a binary representation, where each pixel is converted to a 1 if lightning occurred within the pixel and 0 otherwise. The images are then flattened into vectors of size 1×2304 to be used as the labels in a multi-label classification problem. Finally, both the VIL and IR 10.7 images are normalized to reduce the complexity of the computations during model training.

3.3 Building the Convolutional Neural Network

This research transforms lightning prediction into a multi-label classification problem. In this case, a class is representative of a single pixel within an image that displays the dispersion of lightning strikes in a single storm event. Each pixel is binary, where a 1 indicates that a lightning strike occurred in that pixel and a 0 indicates otherwise. The CNN takes in remote sensing images as the input and classifies them into any number of the many classes (pixels) within the lightning image. The model outputs the probabilities that an input image belongs to each of the classes, representing the probabilities that a lightning strike will occur in each pixel. Therefore, the pixels with high probabilities are predicted to experience a lightning strike.

The CNN used in this research contains 9 layers, including a combination of 4 convolutional and max pooling layers, and a single fully connected layer. The convolutional layers obtain the important features of the image while the pooling layers decrease the spatial size of the convoluted features. The convolutional layers have a kernel size of 3×3 with a stride of 1. The number of filters range in order from 16 to 128, doubling in each convolutional layer. The pooling layers have a pooling window size of 2×2 with a stride of 2. The convolutional and pooling layers both utilize same padding to ensure the outer edges of the images are retained due to the potentially important patterns related to the lightning activity within those regions. Furthermore, the convoluted features are flattened and output through a fully connected layer with a sigmoid activation function that will predict the probability of class membership for each of the pixels in the output map. In other words, each pixel will have a probability from 0 to 1 on whether it contains lightning. Lastly, the model is fit using the Adam optimizer and the binary cross-entropy loss function which compares the predicted probabilities to the true label. Table 1 details the architectures of the CNNs used for the VIL and IR 10.7 modalities. A multimodal CNN is also developed to use both the VIL and IR 10.7 as input. This CNN uses the same architecture in Table 1b except the input size is $192 \times 192 \times 2$ as the VIL images were resized to reflect the smaller dimensions of the IR 10.7 images.

Table 1: CNN Architecture

(a) VIL CNN

Layer	Output Size
Input	384 × 384 × 1
Convolution	384 × 384 × 16
Max Pooling	192 × 192 × 16
Convolution	192 × 192 × 32
Max Pooling	96 × 96 × 32
Convolution	96 × 96 × 64
Max Pooling	48 × 48 × 64
Convolution	48 × 48 × 128
Max Pooling	24 × 24 × 128
Dense	2304

(b) IR 10.7 CNN

Layer	Output Size
Input	192 × 192 × 1
Convolution	192 × 192 × 16
Max Pooling	96 × 96 × 16
Convolution	96 × 96 × 32
Max Pooling	48 × 48 × 32
Convolution	48 × 48 × 64
Max Pooling	24 × 24 × 64
Convolution	24 × 24 × 128
Max Pooling	12 × 12 × 128
Dense	2304

3.3.1 Sparse CNN

Sparsity is introduced into the CNN through a regularizer added to the first convolutional layer of the model. The regularizer adds a penalty term to the loss function which prevents the weights of the network from becoming too large. The penalty term encourages many weight values from the first layer towards zero, effectively selecting a subset of features to be processed throughout the network. This results in a sparse network where only the most important features are considered for prediction. Regularization can help to prevent overfitting and improve the generalization of the model by focusing the model on the relevant features of the input data.

Three different types of regularization techniques are explored in this research: LASSO, Ridge Regression, and Elastic Net. LASSO is particularly suitable for creating sparse models, as it shrinks some of the weights to exactly zero, essentially removing certain features from the model. The LASSO penalty term utilizes the L1 norm which is the sum of the absolute values of the weights:

$$\lambda||\mathbf{w}||_1 = \lambda \sum_{i=1}^{N} |w_i|$$

Here, $\mathbf{w}$ is the vector of weight values in the model and λ is the regularization parameter that controls strength of the regularization.

In contrast to LASSO, Ridge Regression shrinks weights to small, non-zero values. As a result, Ridge Regression is not as effective at focusing models on sparse features as LASSO. However, it can still reduce the impact of certain features by decreasing the magnitude of their weights. The Ridge Regression penalty term utilizes the L2 norm which is the sum of the weights squared:

$$\lambda||\mathbf{w}||_2 = \lambda \sum_{i=1}^{N} w_i^2$$

Finally, Elastic Net combines LASSO and Ridge Regression to zero out certain weights while still ensuring all weights do not become to large in magnitude. The Elastic Net penalty term is a function of both the L1 and L2 norm:

$$\lambda_1||\mathbf{w}||_1 + \lambda_2||\mathbf{w}||_2 = \lambda_1 \sum_{i=1}^{N} |w_i| + \lambda_2 \sum_{i=1}^{N} w_i^2$$

Elastic Net requires more hyperparameter tuning as both the L1 and L2 norms have a regularization parameter, λ_1 and λ_2. Each type of regularization technique aims to reduce the complexity of the model by controlling the model weights and focusing on the most important features. LASSO, Ridge Regression, and Elastic Net are tested to determine which technique produces the best results in the context of this research problem.

3.3.2 Hyperparameter Tuning

Regularization techniques require hyperparameter tuning to choose the optimal regularization parameter, λ. This parameter controls the amount of regularization that is applied to the model. Increasing the parameter value, increases the strength of the regularization which can significantly affect the performance of a model. Next, the regularization techniques themselves must be evaluated to select the method that produces the best results.

This research utilizes Bayesian Optimization to efficiently select the optimal regularization parameters and regularization technique. Bayesian optimization uses a Gaussian process to map the relationship between the hyperparameters and the performance of the model. The response surface is built using prior evaluations of the model which is then used to select the set of hyperparameters most likely to result in the best performance of the model. This process is repeated until the optimal hyperparameters are found [25].

The parameters evaluated range on a logarithmic scale from 0 to 1: 0.0001, 0.001, 0.01, 0.1. The Bayesian optimization tuner is applied to each of the regularization techniques (LASSO, Ridge Regression, and Elastic Net) to select the parameter values from the given list that produce optimal results for the respective technique. Lastly, the Bayesian optimization tuner is applied again to determine the regularization technique that maximizes model performance using the specific optimal regularization parameters for each technique. The best performing regularization technique and its optimal parameter values are then used in model testing.

3.4 Robust Tensor Decomposition

The second way this research introduces sparsity is within the remote sensing images themselves. Sparse features in the images are extracted and then input into the CNN. The intent is to focus the CNN on the spatial features that are most indicative of lightning, while also removing unnecessary information that could potentially distract the model, to inform a better prediction. This is accomplished using RTD, a tensor decomposition method that separates images into their background and sparse features.

RTD decomposes an HD tensor into its low-rank and sparse components by solving the following convex optimization problem:

$$\min_{\mathcal{X},\mathcal{E}} \sum_{i=1}^{N} ||\mathbf{X}_i||_* + \lambda||\mathbf{E}||_1$$
$$\text{s.t.} \quad \mathcal{B} = \mathcal{X} + \mathcal{E}$$

Here, $\mathcal{B}$ is an HD tensor of observed data that is comprised of a low-rank tensor $\mathcal{X}$ and a sparse tensor $\mathcal{E}$. $\mathbf{X}_i$ is a two-dimensional matrix within $\mathcal{X}$, $\mathbf{E}$ is the sparse matrix, and λ is a parameter that controls the sparsity of $\mathbf{E}$. The model finds the lowest rank $\mathcal{X}$ that can generate the data $\mathcal{B}$ while ensuring that the entries of $\mathcal{E}$ are sparse. The nuclear norm is used in place of rank($\mathcal{X}$) because it is the convex relaxation of tensor rank and the L1 norm is used to enforce sparsity within the tensor $\mathcal{E}$.

RTD is solved via an alternating direction method of multipliers (ADMM) algorithm which is an efficient method to solve distributed convex optimization problems by breaking them down into smaller, more manageable subproblems. ADMM forms an augmented Lagrangian function where the primal variables (solutions to the original problem) and dual variables (Lagrange multipliers) are updated in alternating fashion until an optimal solution or consensus is found [26].

In the context of this problem, $\mathcal{B}$ represents a stream of remote sensing images. The low-rank tensor $\mathcal{X}$ can be viewed as a tensor comprised of the backgrounds of the images and $\mathcal{E}$ is the tensor comprised of their sparse features. The λ parameter is chosen based on how well the matrices of sparse features match the most intense regions of the original images. The sparse images are then used as inputs into the CNN. Figure 2 displays of an example of this process on a single image from the vertically integrated liquid (VIL) modality.

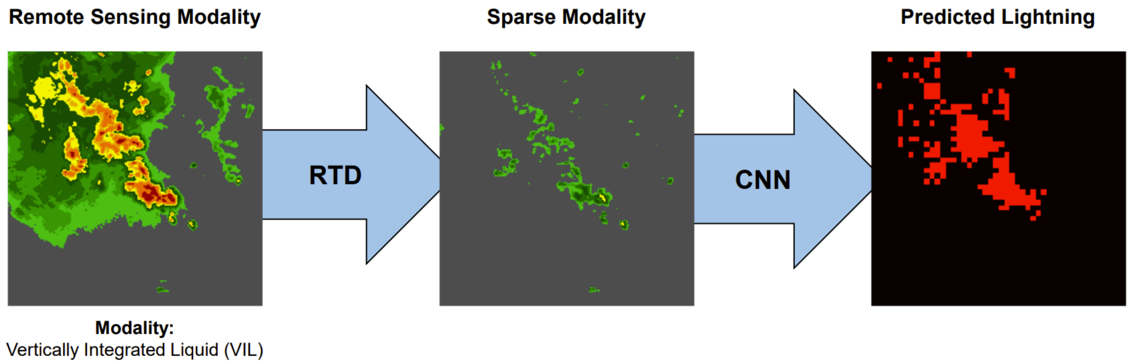


Figure 2: RTD Model Framework

3.5 Model Testing

To evaluate the performance of the sparse CNN and the sparse images created with RTD in the prediction of future lightning strikes, several models are generated using a combination of both methods. The first model uses the sparse CNN with the original remote sensing images as input. The second model uses a non-sparse CNN (i.e. without regularization) with the original images as input. The third model uses the sparse CNN with the RTD images as input (RTD + Sparse CNN). Finally, the fourth model uses a non-sparse CNN with the RTD images as input (RTD + Non-Sparse CNN).

In addition, these four models are tested at different time lags to analyze how their performance changes when predicting lightning activity further in the future. Introducing a time lag means a remote sensing image is paired with a lightning image from a later time step for model training. For example, in a 5-minute time lag, a remote sensing image is paired with the lightning image from the succeeding time step. The different time lags investigated are: 5 minutes, 10 minutes, 15 minutes, 20 minutes, 25 minutes, 30 minutes, 45 minutes, and 1 hour.

The performance of these models are evaluated using four performance metrics: area under the ROC curve (AUC), area under precision-recall curve (AUPRC), precision, and recall. These metrics are chosen because of their ability to better capture the performance of the multi-label classification model in the context of this problem, where typical metrics, such as accuracy, are not appropriate. This specific application is more concerned with capturing the overall pattern of the lightning strikes and accurately predicting specific classes, rather than perfectly matching the entire set of predicted labels to their true labels. AUC measures class separability and informs how well the model can distinguish between positive and negative classes. Precision is the proportion of true positive predictions to all positive predictions made by the model, while recall is the proportion of true positive predictions to all actual positive instances in the data. AUPRC is a single value that summarizes the trade-off between precision and recall. Each of these metrics range between 0 and 1, where 1 indicates perfect classification of the model.

The models are trained to optimize AUC to maximize their ability to recognize what distinguishes pixels that contain lightning from those that do not. This is done by recording the weights that produce the best AUC value on the validation set to ensure the highest performing model is applied to the test set. In addition, early stopping is implemented which ceases training once the validation AUC stops improving with a patience of five epochs and restores the best weights to reduce the effects of overfitting. The four models are trained 10 times each for each of the different time lags, to account for model variation. The average and standard deviation of the performance metrics are recorded for each model and time lag combination.

IV. Results

This chapter presents the results of this thesis using sparse convolutional neural networks (CNN) and sparse inputs to predict the location of lightning strikes. The models were trained and tested on images from the SEVIR data set which includes multiple remote sensing modalities and their corresponding lightning strike data. The different models are compared using several metrics, including area under the ROC curve (AUC) and area under the precision-recall curve (AUPRC). The results demonstrate the potential of utilizing sparse methods for accurate prediction of close-range lightning activity.

4.1 Modality Trials

One of the main focuses of this thesis is to find the remote sensing modality or combination of modalities that best predict future lightning activity. Previous research suggests that vertically integrated liquid (VIL) and 10.7 μm brightness temperature (IR 10.7) modalities contain information that are most indicative of lightning, prompting their use in this application. VIL, IR 10.7, and a combination of the two modalities were used as inputs into the sparse and non-sparse CNNs. For the multi-modal model, the VIL and IR 10.7 images were aligned in a two channel input to be input into the CNNs. The performance of these models were used as initial results to determine the potential of the different modalities in predicting future lightning strikes.

While training and testing these models, the performance of the VIL modality far surpassed the performance of IR 10.7 and the combination of VIL and IR 10.7. The models utilizing the VIL images consistently produced higher values of AUC, AUPRC, precision, and recall at all time lags. Furthermore, the predictions made by the VIL models closely matched the general pattern of the true lightning activity in the majority of the images, whereas the predictions made by the IR 10.7 and multimodal models appeared much more sporadic and random. As a result, IR 10.7 was determined to not contain enough relevant information to indicate the presence of lightning. Additionally, the mutimodal model, may have suffered from information loss by reducing the size of the VIL images and unnecessary complication by adding the IR 10.7 data which provided little value. Therefore, the VIL modality was chosen for further testing and analysis as outlined in chapter III. The remaining results in this thesis focus on the utilization of the VIL modality in the various methods and models tested.

4.2 RTD Results

The VIL sequences of all 156 events were processed through RTD to extract the sparse features from each image. A λ value of 0.08 was subjectively chosen because it generated an appropriate amount of sparsity within the VIL images that align with the corresponding lightning activity. RTD was effective at extracting the regions of the images that are most intense where lightning activity is most likely to occur. Figure 3 displays examples of images where RTD accurately identified the sparse features that match the corresponding regions where lightning strikes occurred in the following frame. The figure also shows the predicted lightning produced by the non-sparse CNN which used the RTD images as input. The model demonstrates its ability to accurately predict the pattern of lightning activity from sparse VIL images in which RTD correctly identifies the sparse features that correspond with lightning.

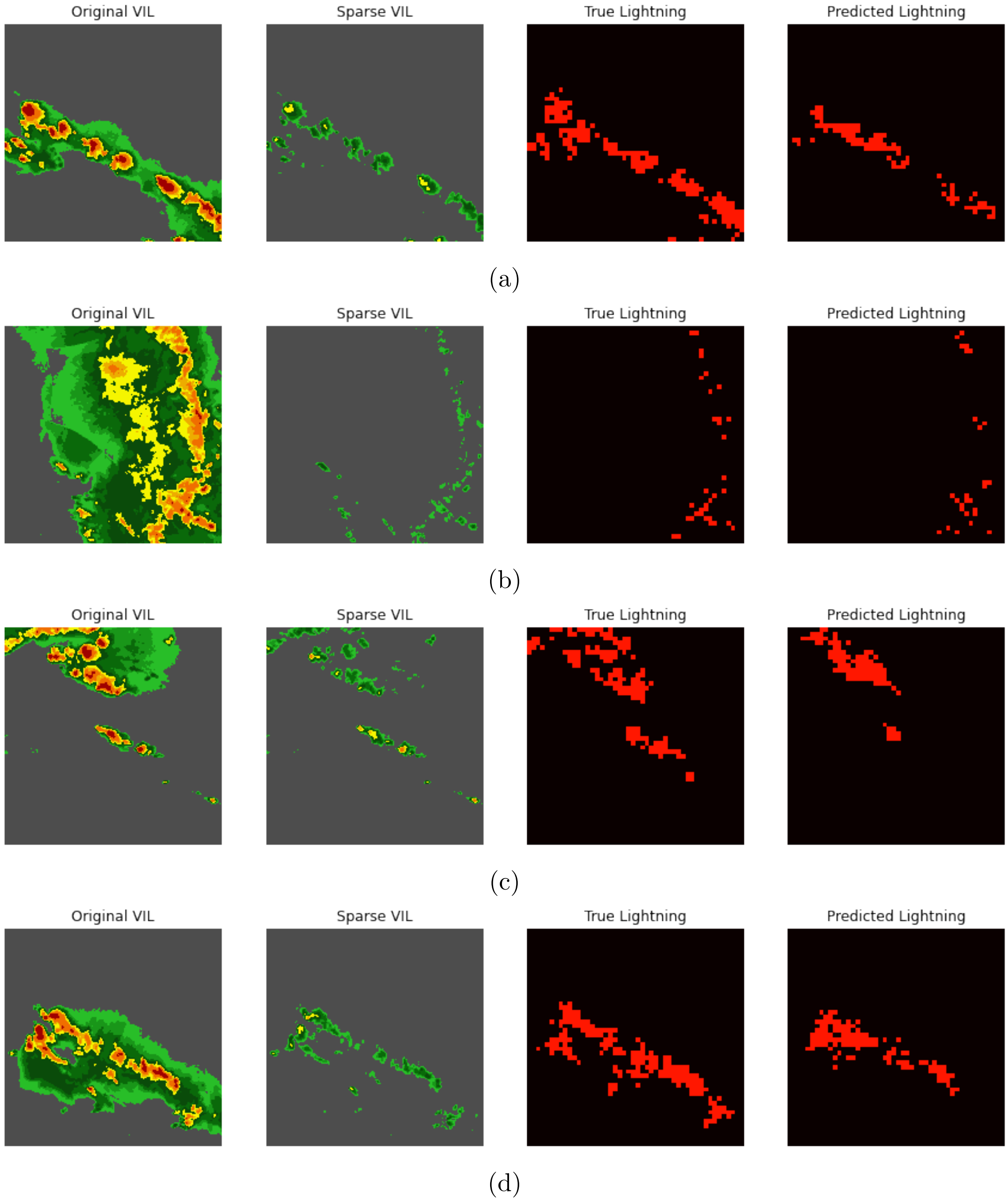
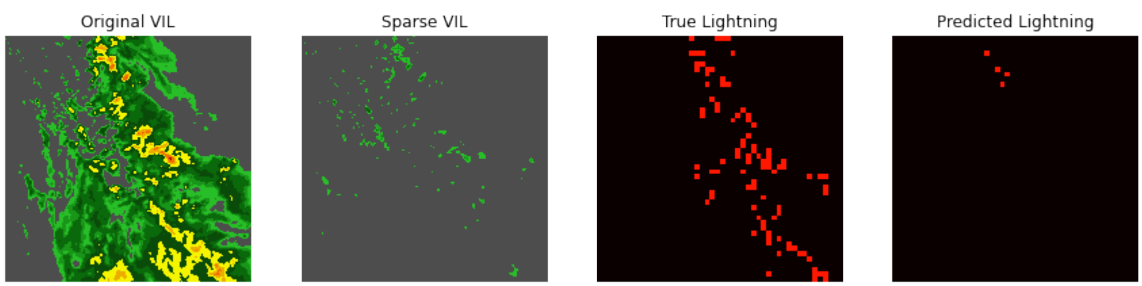
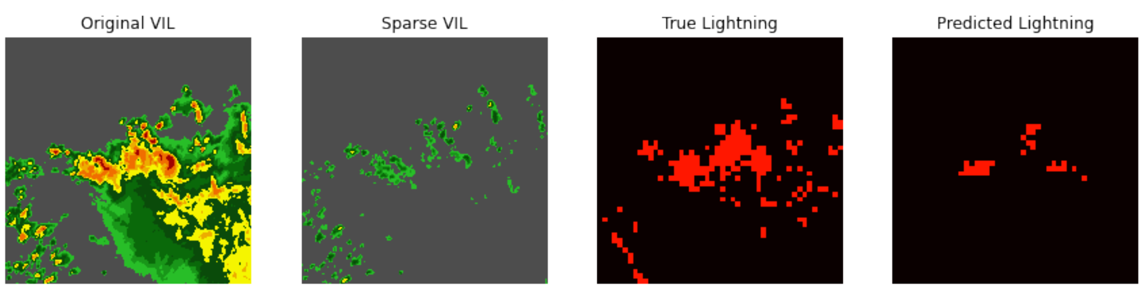


Figure 3: Successful Examples of RTD Applied to VIL Images

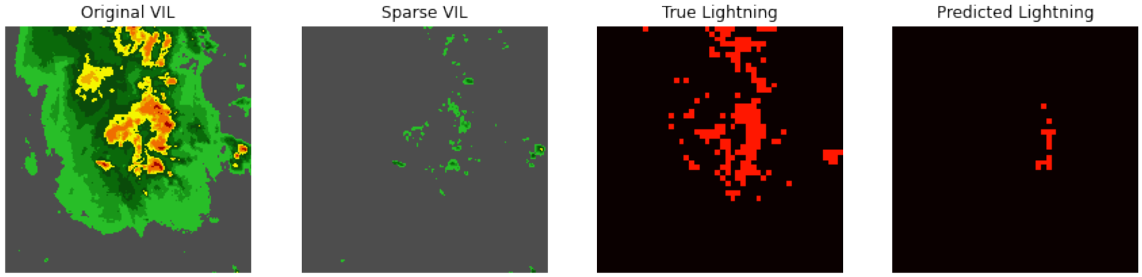
However, RTD also extracted some features that were anomalous respective to their surrounding pixels even if they were not regions with a high concentration of VIL. This resulted in images with many small sparse features which could give a false indication of lightning. Images with numerous sparse features often confused the model, resulting in predictions that did not capture the full scope of lightning within the image. Figures 4a and 4b present examples where the abundance of sparse features fail to highlight the regions with lightning, resulting in poor predictions. Additionally, RTD failed to identify regions where scattered lightning strikes occur that are isolated from the main concentrations of lightning. Figures 4c and 4d display examples where RTD strictly extracted the most intense regions, leaving the model no information that lightning could occur elsewhere. These figures highlight the limitations of extracting sparse features without the target information in mind and using as them as model inputs, as some relevant information will inevitably be thrown out.



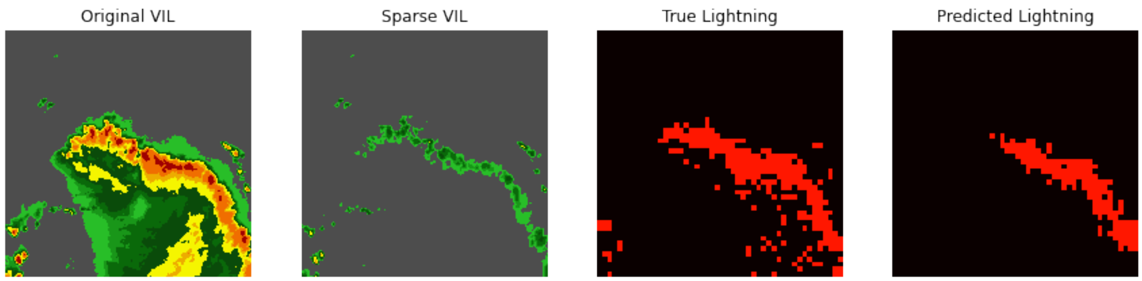
(a)



(b)



(c)



(d)

Figure 4: Poor Examples of RTD Applied to VIL Images

4.3 Model Comparison

This section presents the results of the four models generated to evaluate the performance of the sparse CNN and the sparse images created with RTD in the prediction of future lightning activity. The models tested included: the sparse CNN, the non-sparse CNN, the sparse CNN using RTD images, and the non-sparse CNN using RTD images. Each model was trained and tested 10 times on the same data for each of the eight time lags. The performance metrics were averaged over the 10 runs to account for model variation. Additionally, based on the results of the Bayesian optimization hyperparameter tuning, the sparse CNNs were trained using L1 regularization with a regularization parameter of 0.01.

Each model was prone to overfitting during model training due to the large amount of trainable parameters within these models relative to the size of the training sets. The validation losses would begin to increase after only a few epochs, while the training losses would continue to decrease. Similarly, the validation AUC values would begin to decrease after a few epochs, while the training AUC values would continue to increase. However, the regularization within the sparse CNNs delayed the effects of overfitting by reducing the complexity of the models. Regularization decreased the rates at which the training and validation losses and AUC values diverged from one another. Figures 5 and 6 display the loss and AUC curves for the training and validation sets of the sparse and non-sparse CNNs at the 5-minute time lag, respectively.

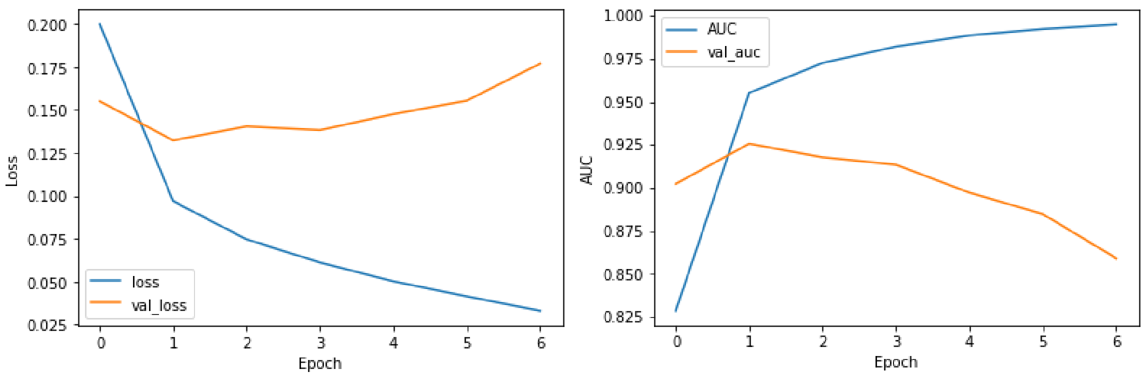


Figure 5: Sparse CNN Performance Curves

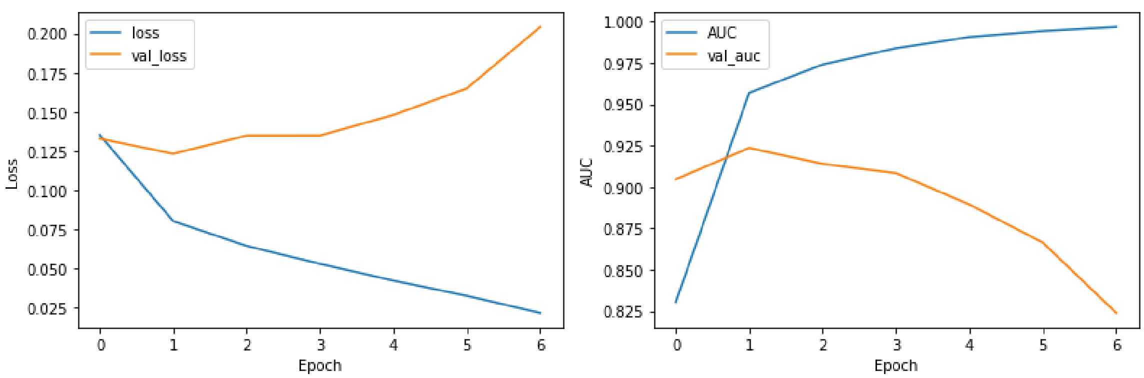


Figure 6: Non-Sparse CNN Performance Curves

4.3.1 AUC Performance

Figure 7 displays the average AUC performance and standard deviation for each of the models. All four models have similar rates of degradation as the time lag increases with a slightly sharper and more constant decline occurring after 30 minutes. This may result from the difficulty of predicting lightning past 30 minutes and from the greater time lag jump taken to reach to 45 minutes and 1 hour. Additionally, the sparse and non-sparse CNNs using the original VIL data produced higher average values of AUC for each time lag than the sparse and non-sparse CNNs using the RTD images. The sparse and non-sparse CNNs produced similar results over all times lags, while the non-sparse CNN using RTD performed slightly better than its sparse counterpart. However, the performance of the models using RTD appear to start converging after the 30-minute time lag. Lastly, the average AUC values for the sparse CNN using RTD were more sporadic over all the time lags, while the other models experienced a smoother decline as the time lags decreased. For further comparison, all average AUC values and standard deviations for each model and time lag combination are recorded in Table 4 in Appendix A.

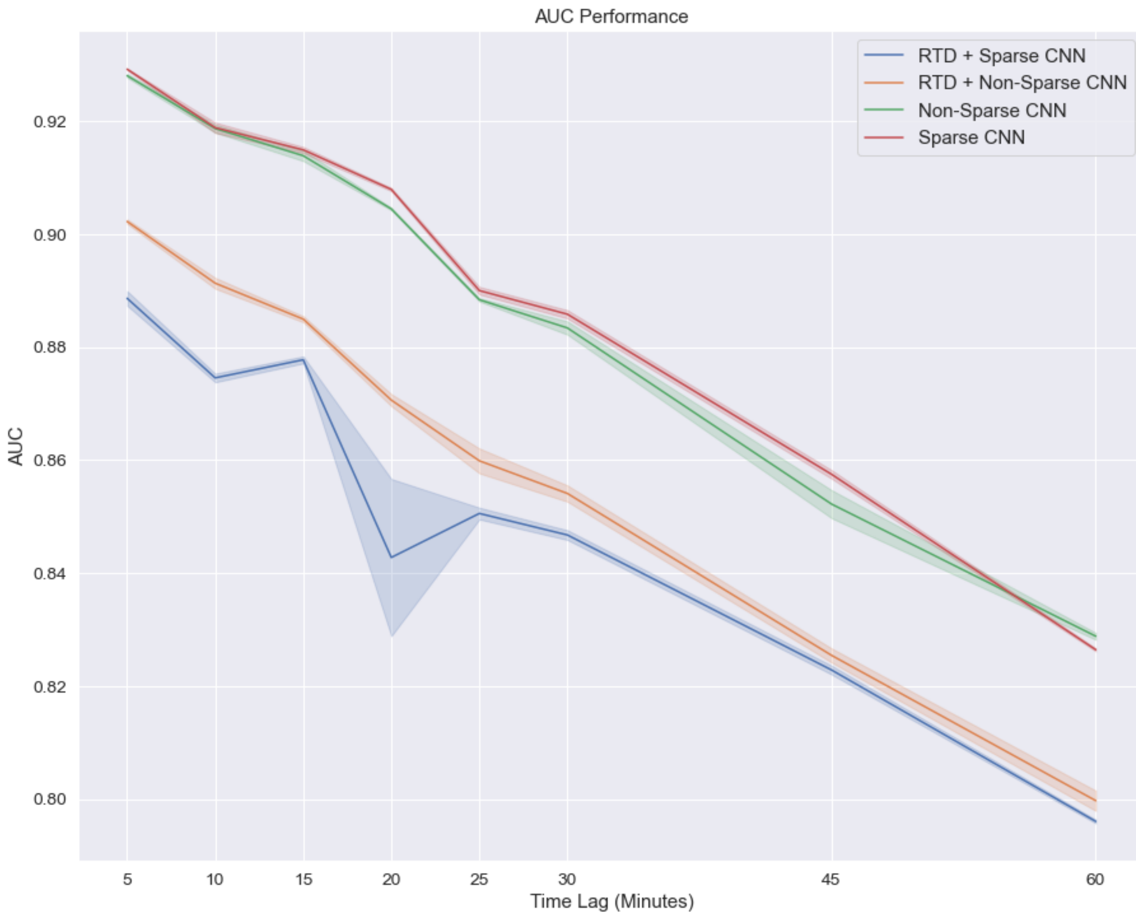


Figure 7: AUC Performance

In order to conclude which models performed best, a two-sample t-test was conducted to compare the AUC performances of both pairs of models at each time lag. The t-test determines whether the population mean of the two sample groups of 10 model evaluations are equal. An F-test was also conducted for each model pair to determine if equal variances could be assumed for each of the t-tests. The appropriate t-test was then performed based on the outcome of the associated F-test. Table 2 displays the p-values from the t-tests comparing the sparse and non-sparse CNNs using the original VIL images. All p-values are statistically significant at the 0.05 level, except at the 10-minute time lag, suggesting there is strong evidence in favor of the alternate hypothesis that the population means of the two models are different at these time lags. Therefore, it can be inferred the sparse CNN results in higher AUC values than the non-sparse CNN at all time lags, except 10 minutes and 1 hour. Table 3 displays the p-values from the t-tests comparing the sparse and non-sparse CNNs using the RTD images. All p-values in this table are statistically significant at the 0.05 level, indicating the non-sparse CNN using RTD images performs better in AUC than the sparse CNN using RTD images at all time lags.

Table 2: t-test: Non-Sparse CNN vs. Sparse CNN

Time Lag	**p-value**
5 minutes	2.47E-7
10 minutes	0.689
15 minutes	0.007
20 minutes	1.26E-16
25 minutes	1.63E-5
30 minutes	2.90E-5
45 minutes	4.94E-5
1 hour	4.70E-8

Table 3: t-test: RTD + Non-Sparse CNN vs. RTD + Sparse CNN

Time Lag	**p-value**
5 minutes	3.81E-12
10 minutes	1.26E-19
15 minutes	3.63E-16
20 minutes	0.001
25 minutes	1.49E-19
30 minutes	5.00E-11
45 minutes	2.86E-5
1 hour	8.17E-5

4.3.2 AUPRC Performance

In addition to AUC, Figure 8 displays the average AUPRC performance and standard deviation for each model over all time lags. The AUPRC performance exhibits similar trends to AUC, as all models degrade in similar fashion and experience a more constant decline after 30 minutes. Furthermore, the sparse and non-sparse CNNs using the original VIL images produced higher average values of AUPRC than the CNNs which used the RTD images at all time lags. Additionally, the sparse and non-sparse CNNs using the original images performed similarly as they did with AUC, while the predictions made by the non-sparse CNN using RTD input resulted in higher average values of AUPRC than its sparse counterpart. Lastly, the sparse CNN produced AUPRC values that were more sporadic over all the time lags than the other models which had more consistent slopes. Because AUPRC summarizes the tradeoffs between both precision and recall, only AUPRC is discussed thoroughly in this section. However, all average and standard deviation values for AUPRC, precision, and recall can be viewed in Appendix A. Most of the model variability was evident in the precision and recall performance compared to AUC and AUPRC which remained relatively constant throughout the 10 model runs.

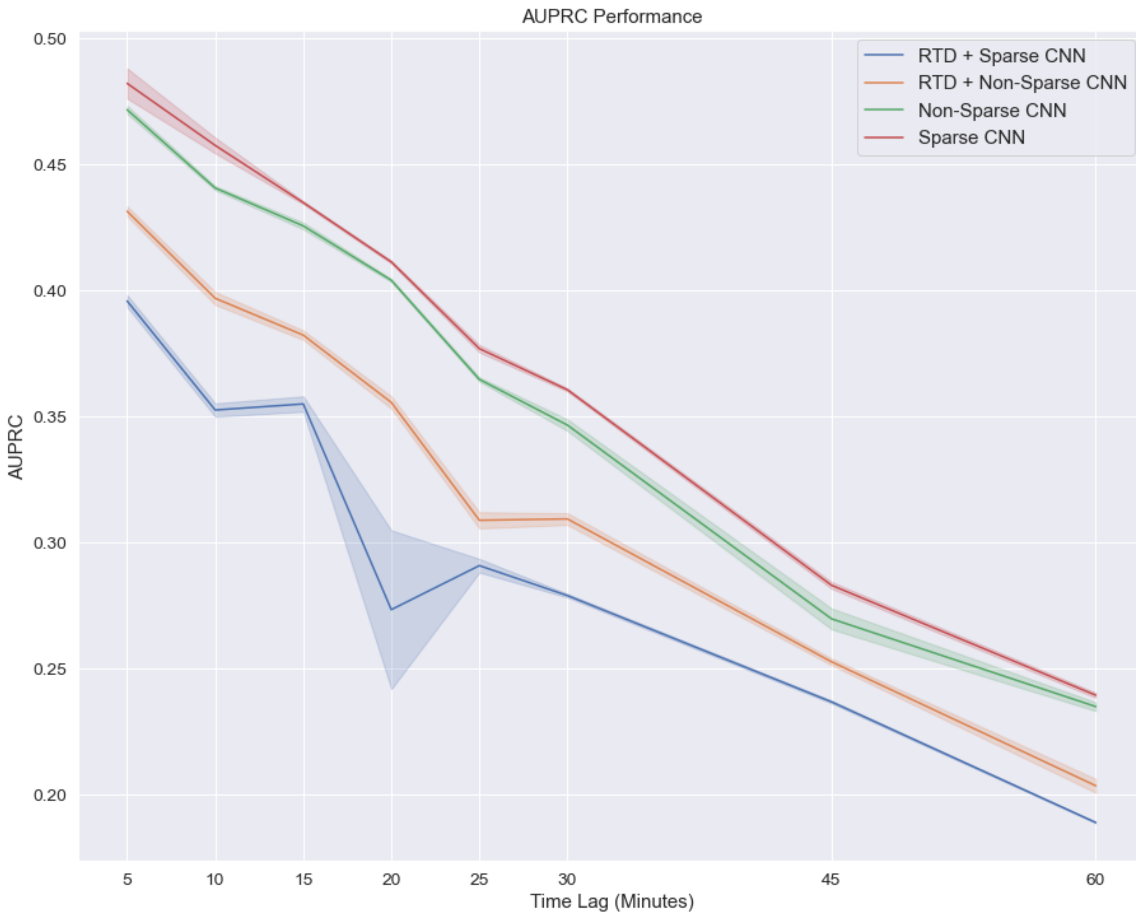
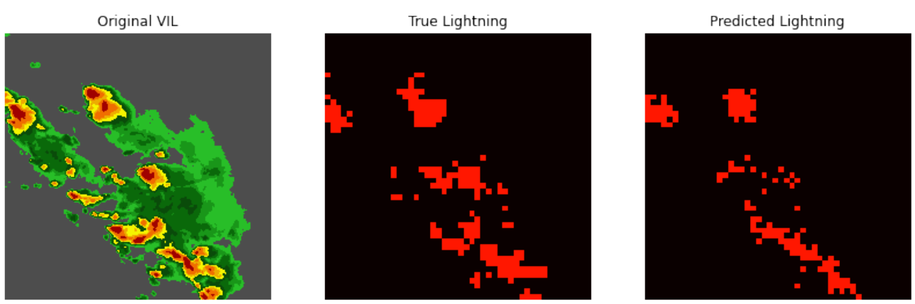


Figure 8: AUPRC Performance

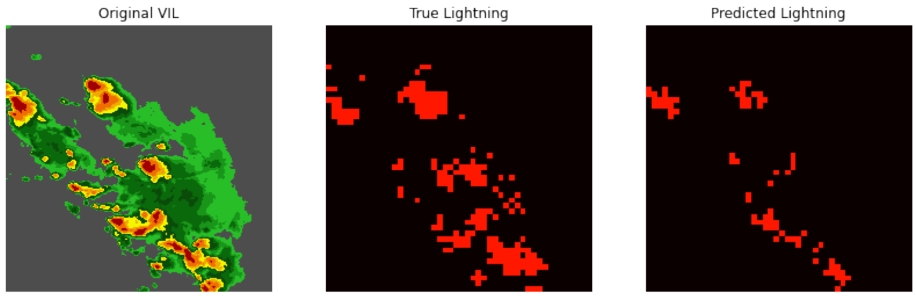
4.3.3 Predictions

In addition to evaluating the performance metrics of the models, it is important to examine their lightning predictions and how they degrade as the time-lag increases. In general, all four models were able to predict the general pattern of lightning activity at lower time lags from 5 minutes to 20 minutes. The patterns within the VIL and lightning sequences only slightly differ within the four frames accounting for this time period, especially within 5 and 10 minutes. Therefore, the models were able to more effectively learn from the information within the VIL data at lower time lags when the distribution of lightning closely matches the patterns within the VIL images.

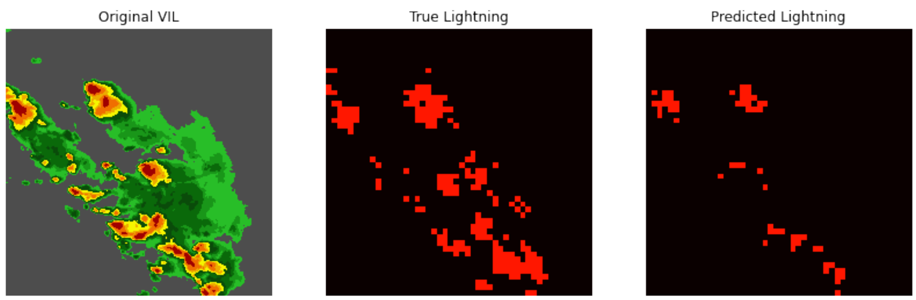
Movement starts to become more apparent after 25 minutes, as lightning moves farther from its original location within the VIL images and morphs into different shapes. Furthermore, there may be instances in storm events that have few strikes where lightning is present in one frame and not in the next, making it very difficult to determine if and where the lightning will occur in greater time lags. As a result, the models become increasingly restrictive with positively classifying a pixel as a lightning strike. However, these models still achieved higher levels of precision for the few lightning strikes they did identify. Predictions at the 1-hour time lag exhibited higher values of recall, but lower values of precision. The models were unable to learn the behavior of lightning an hour in advance from the current VIL images, and would therefore, cast a wider, less precise net of positive predictions. This is because images from recent observations provide little information on the trends of storm developments that far into the future. Figure 9 displays sample predictions at all time lags made by the sparse CNN that exhibit these overarching trends. Figure 12 in Appendix B is provided as an additional example.



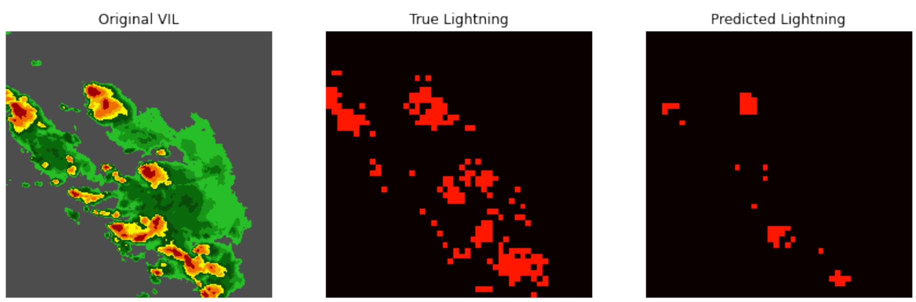
(a) 5-Minute Time Lag



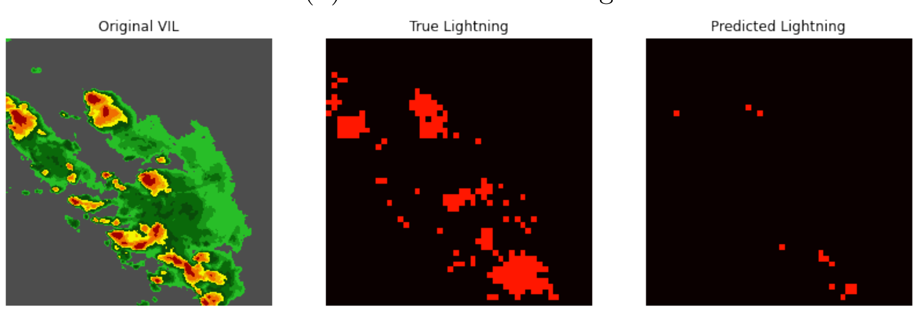
(b) 10-Minute Time Lag



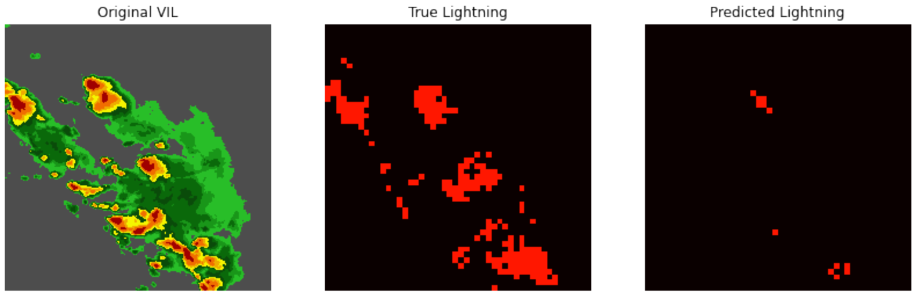
(c) 15-Minute Time Lag



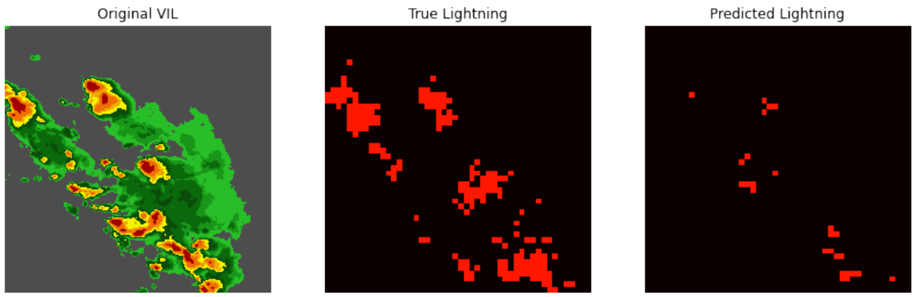
(d) 20-Minute Time Lag



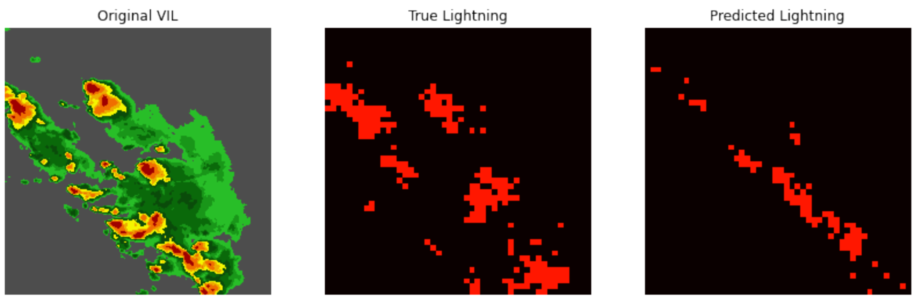
(e) 25-Minute Time Lag



(f) 30-Minute Time Lag



(g) 45-Minute Time Lag



(h) 1-Hour Time Lag

Figure 9: Example Prediction 1

Another trend that occurred in many of the predictions was the inability to detect scattered strikes that are isolated from the main concentrations of lightning activity. Figure 13 in Appendix B presents an example made by the sparse CNN that was able to predict the location of the main cluster of lightning, but failed to predict the outlying strikes throughout the image. This trend was especially apparent in the predictions made by the CNNs using the RTD images, where areas of isolated clusters and strikes were less likely to be identified by RTD and thus, not input into the model.

The models also demonstrated the ability to predict frames that included zero lightning strikes. This is an important capability as many applications require the prospect of no lightning to execute operations. However, correctly predicting the occurrence of no lightning is a difficult task, as VIL instances with zero lightning strikes appear very similar to instances which do contain lightning. Figure 14 in Appendix B displays an example where the sparse CNN was able to consistently avoid predicting lightning over all time lags. In general, the models were able to accurately predict storm events containing no lighting in the majority of their frames. However, there were many instances where the models falsely predicted the occurrence of no lightning in images with few and scattered lightning strikes. This may result because the behavior of lightning within these frames is less related to the apparent patterns within the VIL images. These events also have fewer high intensity regions which the models learned typically indicate the presence of lightning. Furthermore, the amount of lightning in their frames fluctuates with some frames containing no lightning at all. These instances can confuse and mislead the model into falsely predicting no lightning. Figure 15 in Appendix B presents sample predictions made by the sparse CNN where the model falsely predicted no lightning under these circumstances.

V. Discussion

This thesis explores the effects of applying sparse methods within convolutional neural networks (CNN) to accurately predict the location of future lightning strikes. A combination of regularization and tensor decomposition techniques were used to call attention to important features within HD remote sensing modalities and reduce the complexity of the models. L1 regularization induces sparsity within the CNNs during model training, while robust tensor deposition (RTD) extracts sparse features within the input images themselves. Four different models were trained and tested to evaluate the performance of these methods and the different sensing modalities in the prediction of lightning activity.

5.1 Key Findings

Initial model results indicate the superior performance of vertically integrated liquid (VIL) compared to 10.7 μm brightness temperature modality (IR 10.7) in the prediction of lightning. This finding aligns with previous studies which suggest VIL measurements are more highly correlated to the density of lightning strikes within thunderstorms than other commonly observed sensing modalities. As a result, the VIL modality was used to conduct the remaining model testing to evaluate the performance of the different sparse methods.

The sparse CNN using L1 regularization and the original VIL images outperformed the other models at most time lags by producing higher area under the ROC curve (AUC) and area under the precision-recall curve (AUPRC) values. This indicates that encouraging sparsity via regularization to extract important features from non-sparse images improves the prediction of future lightning strikes compared to the other methods presented in this thesis. The results also suggest sparse inputs do not contain enough relevant information to capture the behavior of lightning activity as the original, non-sparse images. Additionally, using a sparse CNN with sparse inputs performed the worst, as it appears to eliminate too many relevant features from the inputs to accurately predict future lightning. Finally, the results show a degradation of performance for each model as the time lag of prediction increases. The models accurately predicted the approximate location of lightning strikes at smaller time-lags. This demonstrates the potential of using deep learning methods that learn spatial information, such as CNNs, in combination with sparse methods which call attention to important features for the prediction of close-range lightning activity.

Furthermore, all models performed best when predicting lightning in events that displayed consistent patterns throughout their many frames. For example, events that moved slowly and did not drastically change shape through time produced more accurate predictions, especially as the prediction window grew larger compared to other events. The models generated predictions that better matched the general pattern of the true lightning in events with consistent distributions of lightning as well. Lastly, the models struggled to predict outlying and scattered strikes that are isolated from the main clusters of lightning activity. These observations highlight the limitations of using remote sensing images to identify and predict the relatively unpredictable behavior of lightning. There is no modality or combination of modalities that can fully capture the scope of lightning because lightning strikes may still occur with little indication from the available information within remote sensing modalities.

5.1.1 Sparse Method Comparison

The different sparse methods utilized in this thesis aim to call attention to the most important features of the input data and reduce model complexity to generate predictions of future lightning activity. In this application, inducing sparsity within

CNNs, via L1 regularization, performed better than using sparse input data created with RTD. RTD extracts sparse features from the remote sensing images without any consideration of the target variable (i.e. lighting data) which consequently, prevented valuable information from being input into the models. In contrast, regularization encourages sparsity within the model weights during models training, effectively selecting a subset of important features. Because regularization takes place during mode training, the sparse CNN selects the relevant features while taking into account the information within the lighting images. This difference may be the reason the sparse CNN using the original VIL images outperformed the non-sparse CNN using the RTD images. The sparse CNN could effectively select the sparse features by zeroing out specific model weights that were also most indicative of the associated lightning.

5.2 Future Work

There are many potential avenues for future work building upon the results presented in this thesis. With the increasing demand for accurate weather forecasts, there is a need for robust and reliable models able to predict lightning activity with high accuracy. The results of this thesis demonstrate promising results in predicting lightning activity. However, there is still room for improvement and various directions for exploration. This section discusses various methods with the potential to be developed and applied to extend the capabilities of this thesis. These recommendations could improve the prediction accuracy of incoming lightning farther in advance and may help address some of the limitations presented in the results.

5.2.1 Probability Predictions

The methodology of this thesis transforms lightning prediction into a binary classification problem. The CNNs output the predictions where each pixel contains a probability from 0 to 1 on whether it contains lightning. The possibility of generating probability maps from these results were explored briefly during this research. However, the probabilities output from the model did not display a great representation of the true lightning activity. Therefore the probabilities were rounded to one which produced results that better resembled the patterns of the true lightning.

However, presenting a map displaying the probabilities of whether lightning will occur in each pixel could be an extremely useful tool for a variety of different missions requiring the advanced knowledge of imminent lightning to plan and execute operations. These probability maps could drastically improve mission safety as they would provide a buffer around areas most likely to experience lightning. Additionally, they would provide more confidence in the predictions of regions potentially free of future lightning strikes. Exploring different techniques to improve the predicted probabilities could greatly improve the functionality of the predictions presented in this thesis for use in a multitude of different fields.

Figures 10 and 11 display example probability maps produced by the Sparse CNN. The probabilities in these maps are inflated to emphasize the distribution of pixels with larger probabilities relative to the majority of pixels (black regions) in the images. Figure 10 presents a probability map for the sample prediction in Figure 13a. In the original prediction, where the probabilities are rounded to one, the model was unable to predict the isolated cluster of lightning. However, this map reveals the model is actually capable of predicting the isolated cluster, just at a lower probability. Similarly, Figure 11 presents a probability map for the sample prediction in Figure 15a where the model falsely predicted the occurrence of no lightning. However, the map reveals that the model's highest predicted probabilities in the frame aligned with the region containing the most lightning. This demonstrates the model's capacity to predict the presence of lightning in frames with few and scattered lightning strikes.

These examples highlight the potential of scaling the probabilities from the predictions to create functional probability maps. They also unveil the possibility of lowering the threshold at which probabilities are rounded up to be classified as lightning strikes. A combination of these potential alternatives could be explored and tuned to find the probability maps that best represent true distribution of future lightning. This would help address the limitations and improve upon the predictions presented in this thesis.

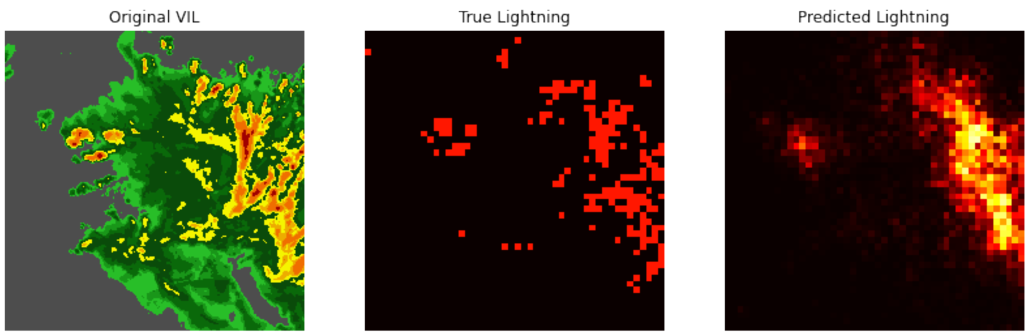


Figure 10: Example Probability Map of Figure 13a

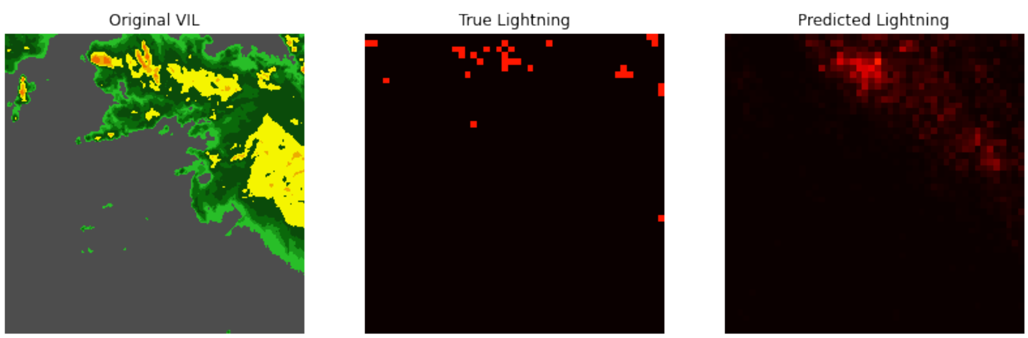


Figure 11: Example Probability Map of Figure 15a

5.2.2 Spatio-Temporal Methods

Because only single frames are input into the CNN, they have a difficult time learning the future patterns and developments of storm events. As a result, predictions of lightning increasingly degrade as the time lag increases. Storm events move rapidly through space and change shape over time. Therefore, methods that consider the temporal element in addition to the spatial elements of the entire remote sensing modality sequences are necessary to predict lighting farther in advance. The convolutional LSTM (ConvLSTM) is a deep learning technique that combines the CNN and the Long Short-Term Memory (LSTM) architectures, allowing it to process both spatial and temporal data. Shi et al. demonstrate the power of ConvLSTMs in capturing spatio-temporal data by accurately forecasting future precipitation maps, showcasing its ability perform well in weather applications, such as the one presented in this thesis [27]. The transformer is another deep learning method built to process sequential data such as natural language or time series data. It uses self-attention mechanisms to weigh the importance of each element in a sequence, enabling the model to process information in parallel, leading to improved processing times and performance [28]. The transformer could improve the prediction of lightning activity at greater time lags through its ability to process large amounts of spatio-temporal data.

In addition to using spatio-temporal methods, the inclusion of more data input into the models could help improve the prediction of lightning activity. The computational constraints in this thesis limited the amount of storm events used in model training. However, the SEVIR data set contains upwards of 10,000 storms events available for analysis. The utilization of more data in model training would supply the models with a plethora of diverse storm events that could provide relevant insights and improve the prediction of the behavior of lightning.

5.2.3 Supervised RTD

Another method that could be implemented to improve upon the methods presented in this thesis is a supervised version of RTD. Because RTD provided promising results in identifying the regions where lightning was most likely to occur, the development of a supervised version could be a productive way forward. RTD decomposes the remote sensing sequences into their low-rank and sparse components without considering the information in the associated lightning images, resulting in sparse images that disregarded information valuable to the prediction of lightning. A supervised version of RTD would take into account the information within the lightning images while extracting the sparse features of the remote sensing modalities. This could reduce instances where RTD performed poorly, leaving out relevant information that could inform certain patterns with the ability to indicate the development of storm events and the locations of lightning strikes. The development of supervised RTD would also extend the current RTD framework to make predictions of the target variable. Extracting sparse features in remote sensing modalities under the supervision of the associated lightning activity could vastly improve the prediction of close-range lightning strikes.

Appendix A. Performance Tables

Table 4: Average AUC Performance

Time Lag	Sparse CNN	Non-Sparse CNN	RTD + Sparse CNN	RTD + Non-Sparse CNN
5 minutes	0.9292 ± 0.0002	0.9281 ± 0.0003	0.8887 ± 0.0013	0.9023 ± 0.0004
10 minutes	0.9189 ± 0.0009	0.9187 ± 0.0007	0.8746 ± 0.0008	0.8913 ± 0.0010
15 minutes	0.9149 ± 0.0005	0.9139 ± 0.0009	0.8778 ± 0.0006	0.8850 ± 0.0005
20 minutes	0.9079 ± 0.0003	0.9045 ± 0.0002	0.8428 ± 0.0139	0.8707 ± 0.0011
25 minutes	0.8900 ± 0.0007	0.8884 ± 0.0003	0.8506 ± 0.0010	0.8599 ± 0.0022
30 minutes	0.8859 ± 0.0007	0.8834 ± 0.0012	0.8468 ± 0.0009	0.8541 ± 0.0014
45 minutes	0.8575 ± 0.0008	0.8522 ± 0.0025	0.8205 ± 0.0008	0.8255 ± 0.0013
1 hour	0.8265 ± 0.0003	0.8289 ± 0.0006	0.7961 ± 0.0003	0.7997 ± 0.0018

Table 5: Average AUPRC Performance

Time Lag	Sparse CNN	Non-Sparse CNN	RTD + Sparse CNN	RTD + Non-Sparse CNN
5 minutes	0.4820 ± 0.0060	0.4716 ± 0.0016	0.3957 ± 0.0024	0.4313 ± 0.0020
10 minutes	0.4574 ± 0.0031	0.4405 ± 0.0008	0.3525 ± 0.0027	0.3968 ± 0.0026
15 minutes	0.4348 ± 0.0006	0.4255 ± 0.0014	0.3549 ± 0.0031	0.3822 ± 0.0019
20 minutes	0.4112 ± 0.0006	0.4040 ± 0.0006	0.2734 ± 0.0315	0.3555 ± 0.0025
25 minutes	0.3769 ± 0.0015	0.3647 ± 0.0009	0.2909 ± 0.0027	0.3088 ± 0.0032
30 minutes	0.3606 ± 0.0007	0.3466 ± 0.0023	0.2790 ± 0.0007	0.3094 ± 0.0023
45 minutes	0.2830 ± 0.0014	0.2698 ± 0.0042	0.2325 ± 0.0006	0.2527 ± 0.0013
1 hour	0.2395 ± 0.0010	0.2350 ± 0.0018	0.1890 ± 0.0003	0.2036 ± 0.0027

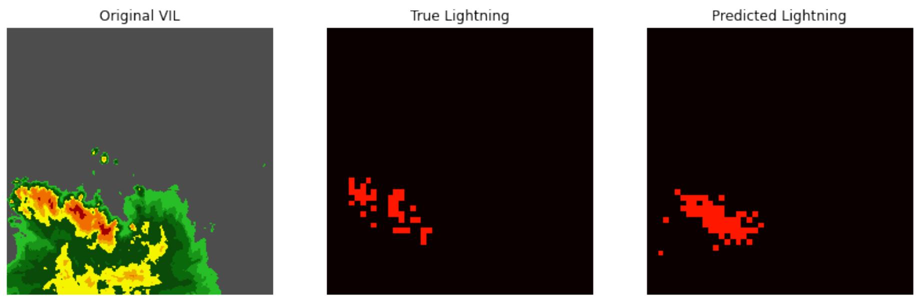
Table 6: Average Precision Performance

Time Lag	Sparse CNN	Non-Sparse CNN	RTD + Sparse CNN	RTD + Non-Sparse CNN
5 minutes	0.5830 ± 0.0082	0.5717 ± 0.0103	0.5208 ± 0.0007	0.5551 ± 0.014
10 minutes	0.5849 ± 0.0146	0.5846 ± 0.0060	0.5561 ± 0.0165	0.5316 ± 0.0943
15 minutes	0.5965 ± 0.0058	0.5803 ± 0.0073	0.5285 ± 0.0181	0.5504 ± 0.0131
20 minutes	0.5877 ± 0.0044	0.5875 ± 0.0030	0.4514 ± 0.0607	0.5597 ± 0.0155
25 minutes	0.5728 ± 0.0021	0.5517 ± 0.0030	0.4040 ± 0.0113	0.4162 ± 0.0080
30 minutes	0.5311 ± 0.0094	0.5237 ± 0.0139	0.4567 ± 0.0035	0.4917 ± 0.0102
45 minutes	0.4474 ± 0.0054	0.4357 ± 0.0087	0.3700 ± 0.0325	0.4069 ± 0.0031
1 hour	0.3482 ± 0.0051	0.3533 ± 0.0044	0.2774 ± 0.0213	0.3060 ± 0.0059

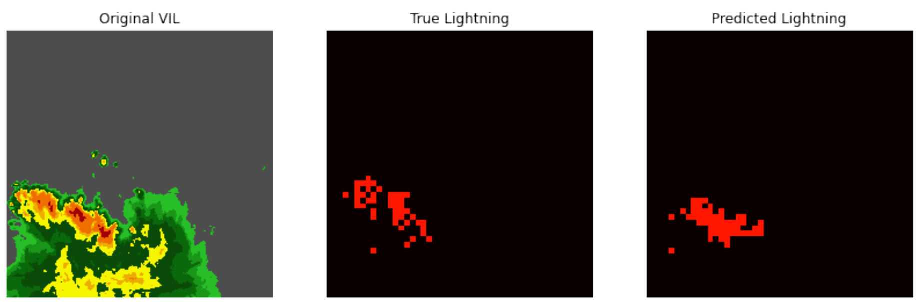
Table 7: Average Recall Performance

Time Lag	Sparse CNN	Non-Sparse CNN	RTD + Sparse CNN	RTD + Non-Sparse CNN
5 minutes	0.3576 ± 0.0200	0.3648 ± 0.0285	0.3154 ± 0.0040	0.3331 ± 0.0263
10 minutes	0.3114 ± 0.0272	0.2567 ± 0.0147	0.1588 ± 0.0214	0.2376 ± 0.0154
15 minutes	0.2168 ± 0.0135	0.2267 ± 0.0155	0.1891 ± 0.0219	0.2141 ± 0.0211
20 minutes	0.1618 ± 0.0089	0.1523 ± 0.0055	0.0516 ± 0.0309	0.1386 ± 0.0230
25 minutes	0.1207 ± 0.0038	0.1258 ± 0.0054	0.2627 ± 0.0246	0.2837 ± 0.0120
30 minutes	0.1601 ± 0.0257	0.1256 ± 0.0245	0.1016 ± 0.0067	0.1347 ± 0.0132
45 minutes	0.1031 ± 0.0123	0.0813 ± 0.0210	0.0960 ± 0.0032	0.1210 ± 0.0063
1 hour	0.1886 ± 0.0107	0.1418 ± 0.0091	0.1634 ± 0.0322	0.1602 ± 0.0089

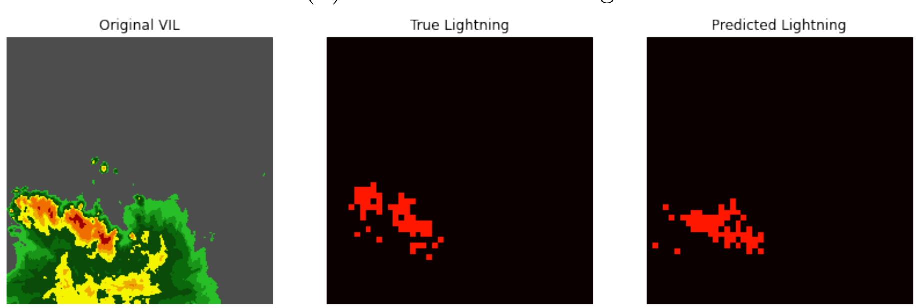
Appendix B. Sparse CNN Example Predictions



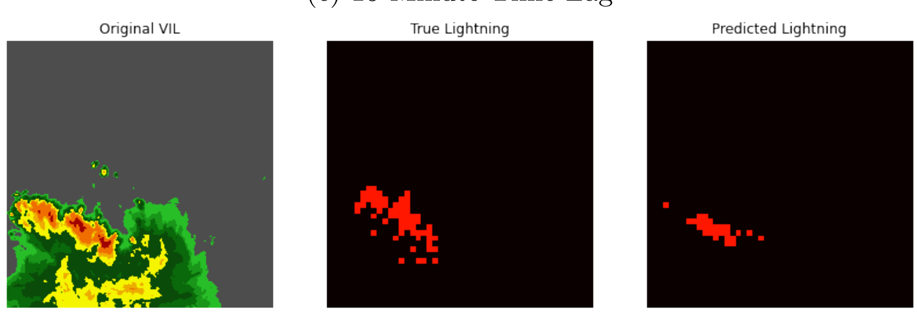
(a) 5-Minute Time Lag



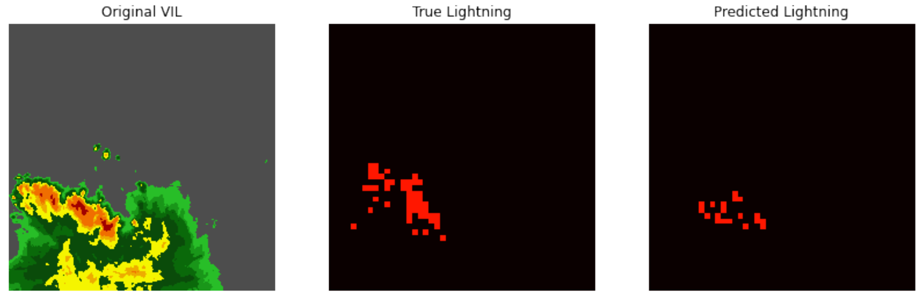
(b) 10-Minute Time Lag



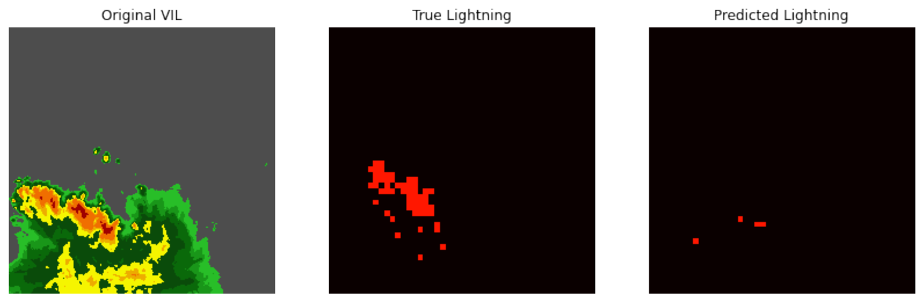
(c) 15-Minute Time Lag



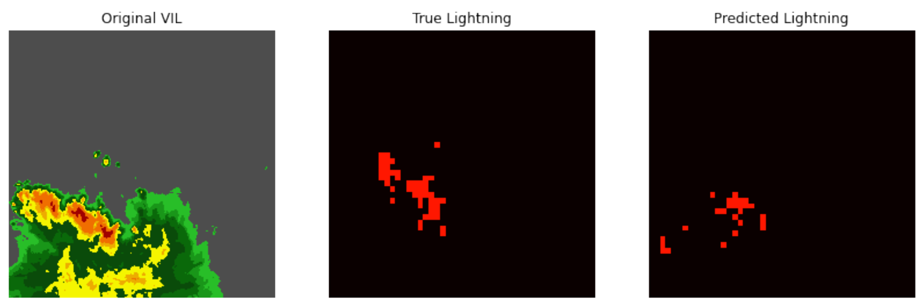
(d) 20-Minute Time Lag



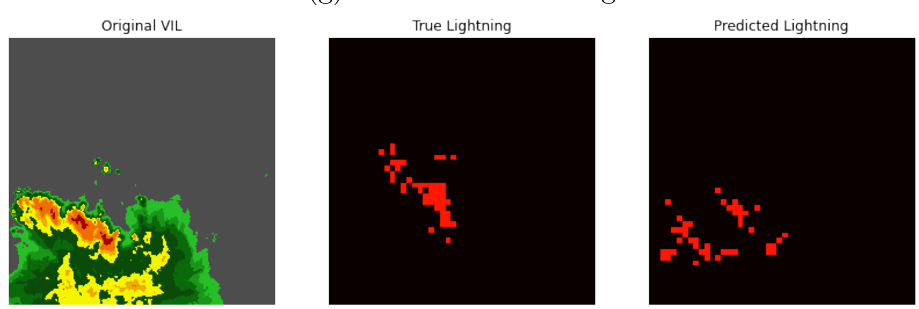
(e) 25-Minute Time Lag



(f) 30-Minute Time Lag



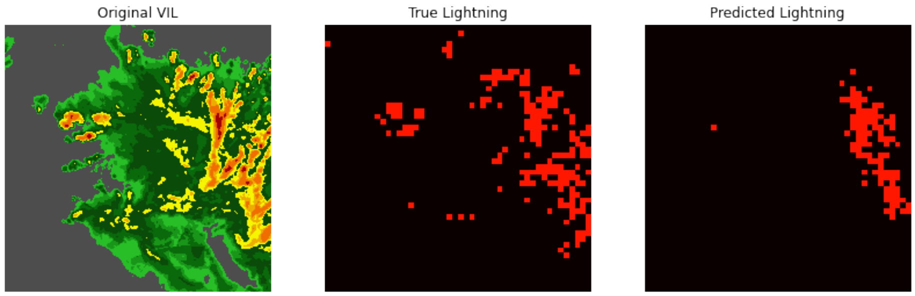
(g) 45-Minute Time Lag



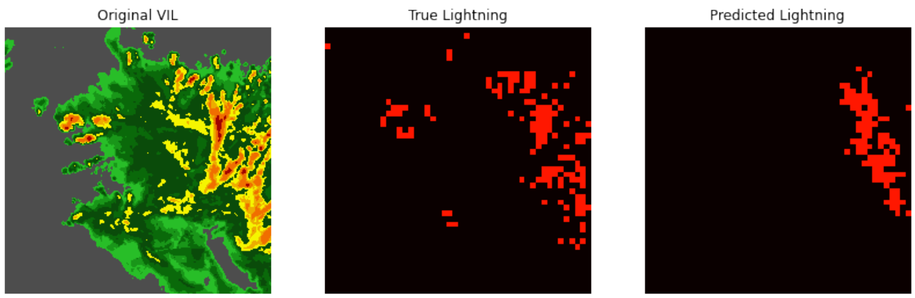
(h) 1-Hour Time Lag

Figure 12: Example Prediction 2

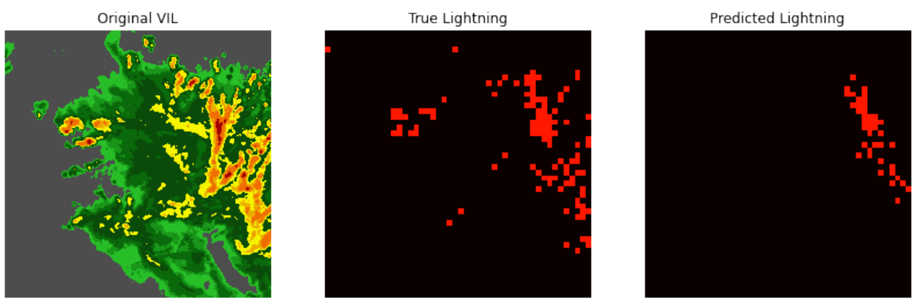
This figure displays a typical sample prediction where the model accurately predicts the general pattern of future lightning activity for the first 20 minutes and progressively degrades as the time lag increases.



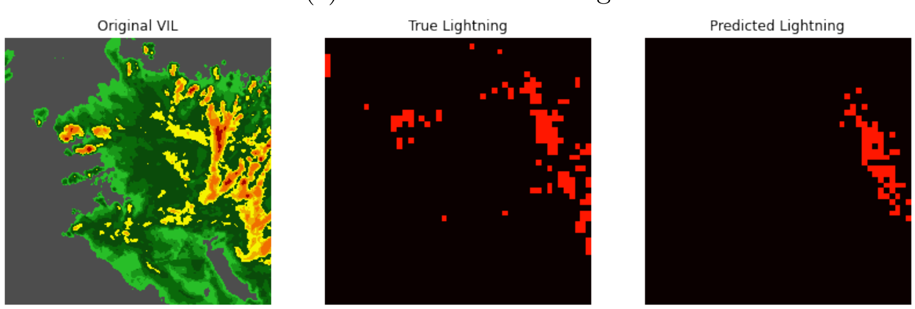
(a) 5-Minute Time Lag



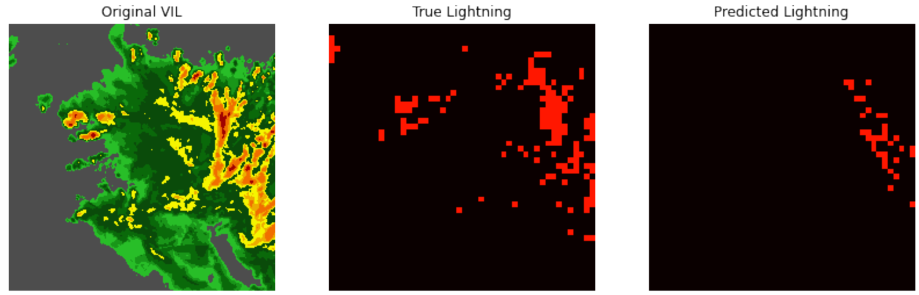
(b) 10-Minute Time Lag



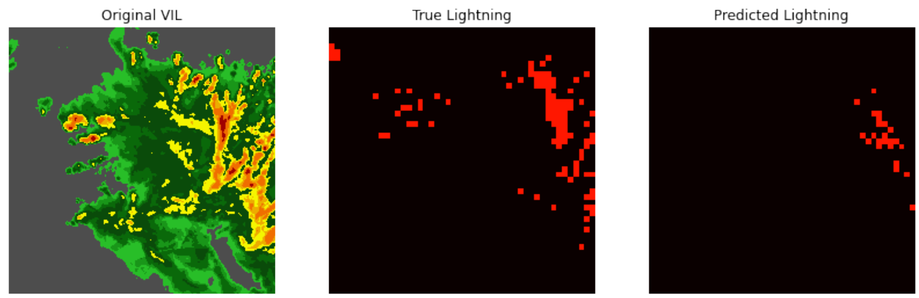
(c) 15-Minute Time Lag



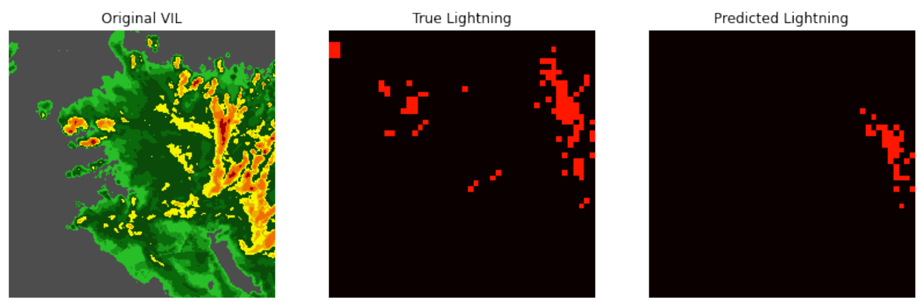
(d) 20-Minute Time Lag



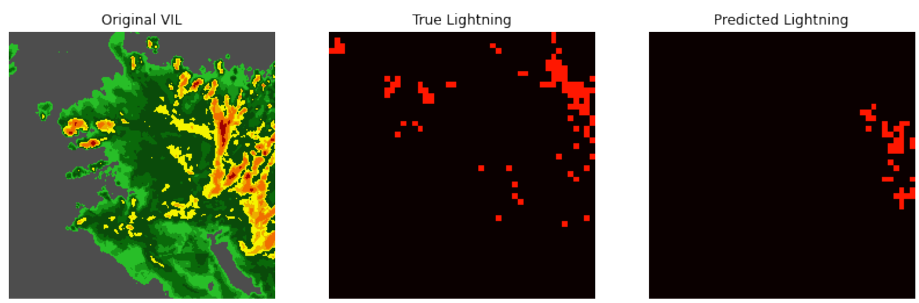
(e) 25-Minute Time Lag



(f) 30-Minute Time Lag



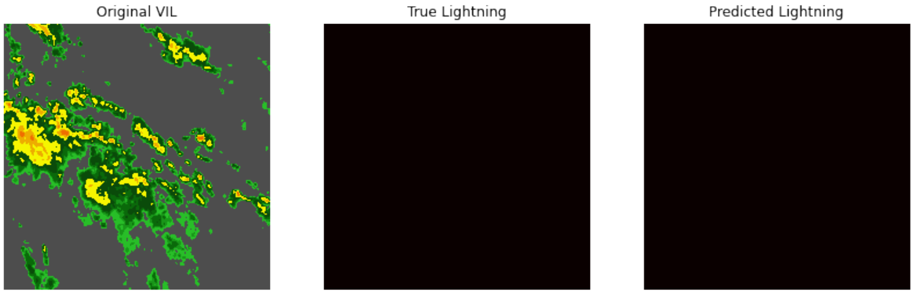
(g) 45-Minute Time Lag



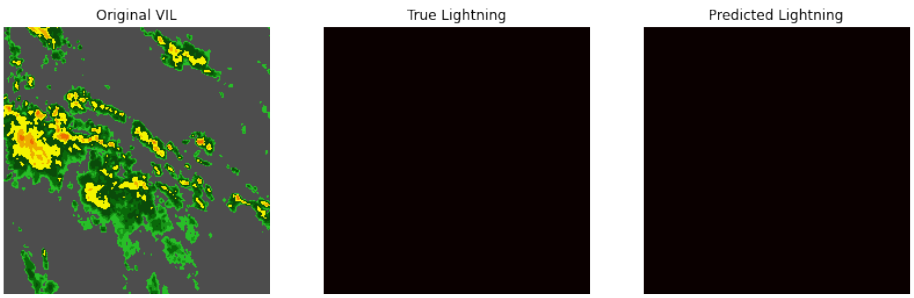
(h) 1-Hour Time Lag

Figure 13: Example Prediction 3

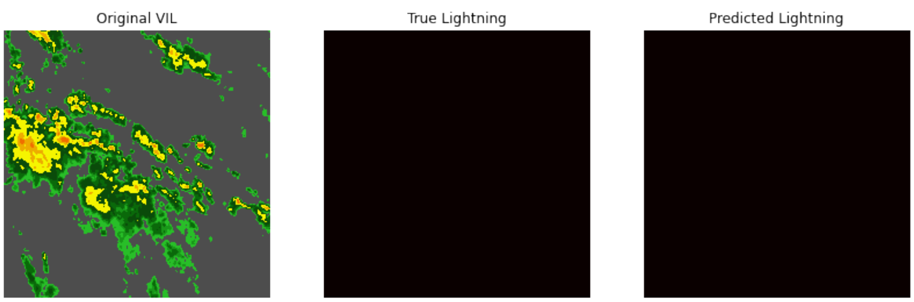
This figure displays a sample prediction where the model accurately identifies the location of the main concentration of lightning, but fails to detect the isolated clusters of surrounding lightning strikes.



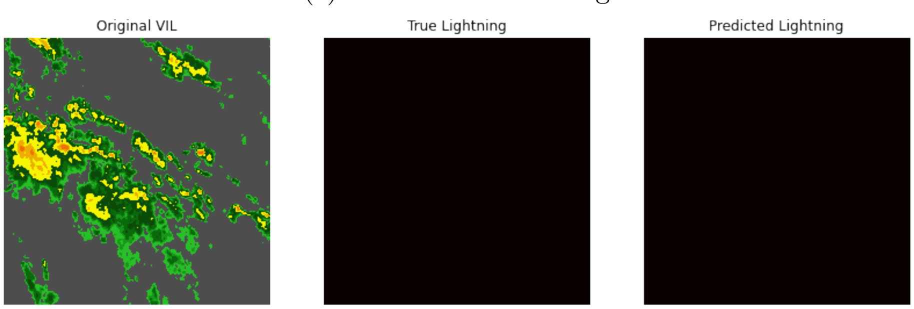
(a) 5-Minute Time Lag



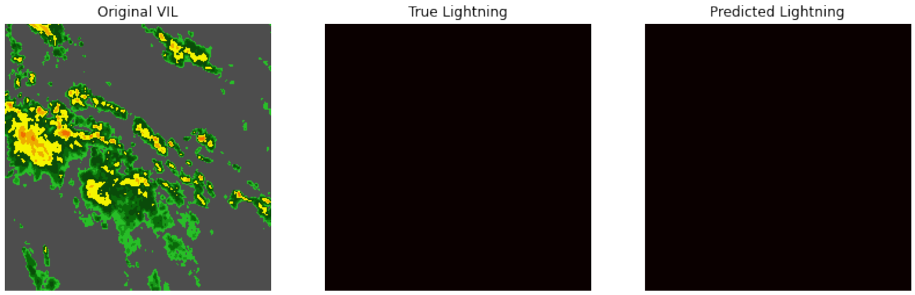
(b) 10-Minute Time Lag



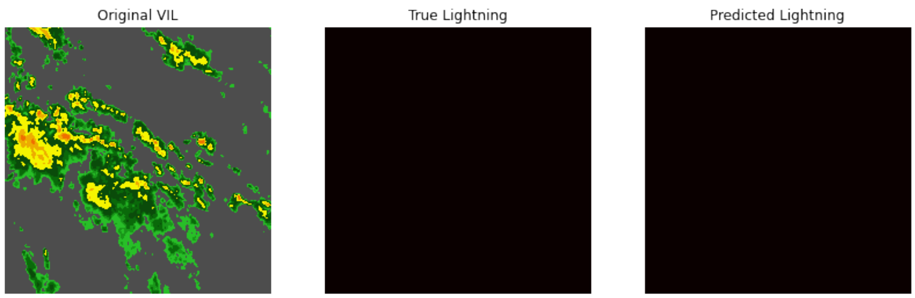
(c) 15-Minute Time Lag



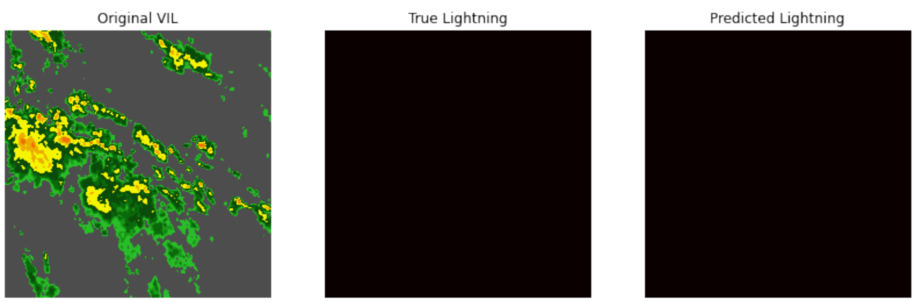
(d) 20-Minute Time Lag



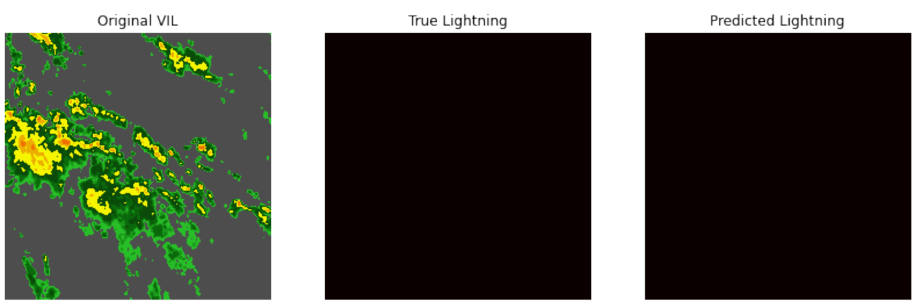
(e) 25-Minute Time Lag



(f) 30-Minute Time Lag



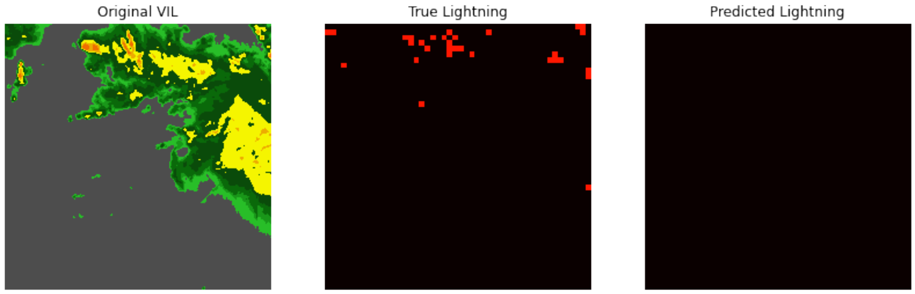
(g) 45-Minute Time Lag



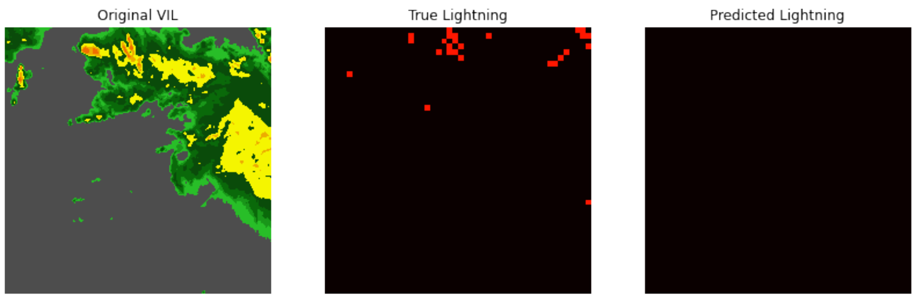
(h) 1-Hour Time Lag

Figure 14: Example Prediction 4

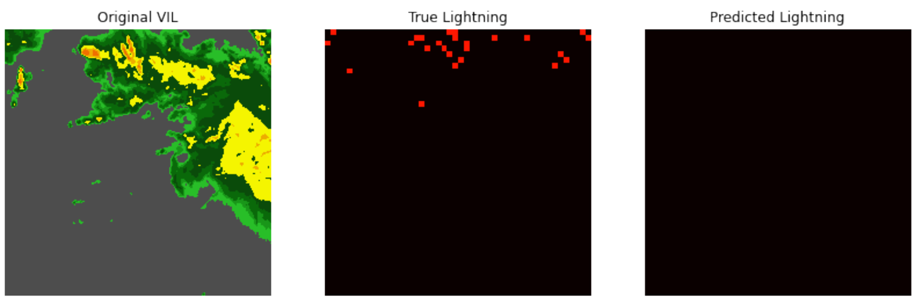
This figure displays a sample prediction where the model rightfully avoids predicting the presence of future lighting strikes in a storm event where no lightning occurs.



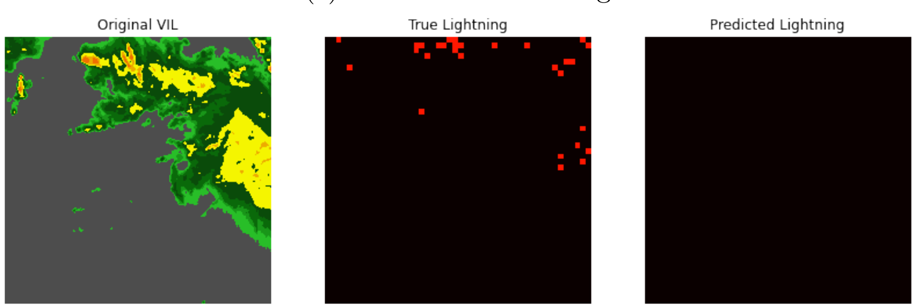
(a) 5-Minute Time Lag



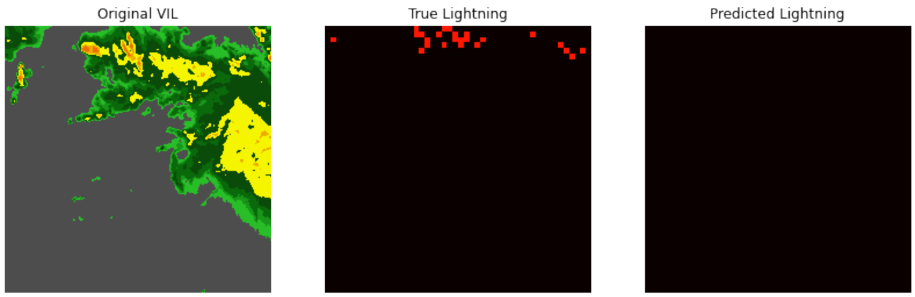
(b) 10-Minute Time Lag



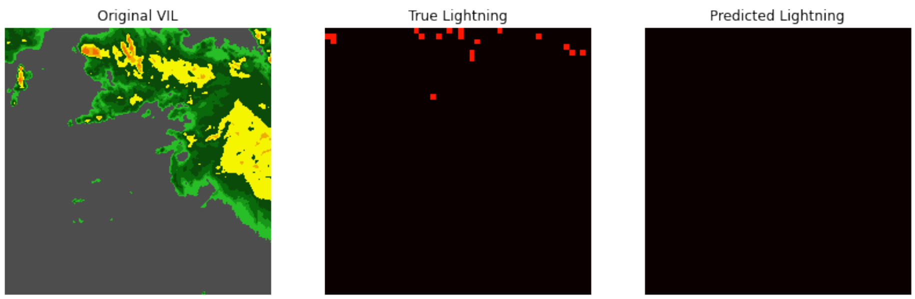
(c) 15-Minute Time Lag



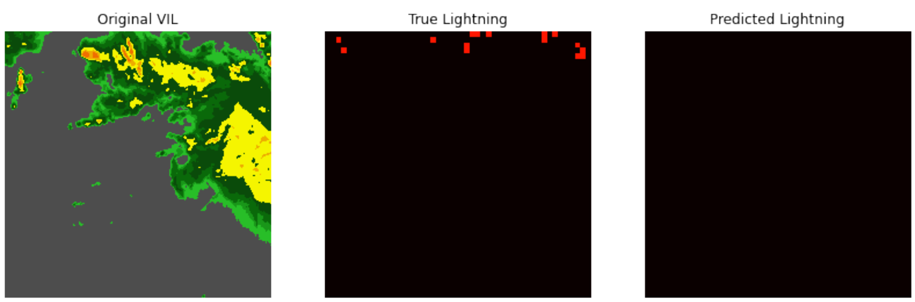
(d) 20-Minute Time Lag



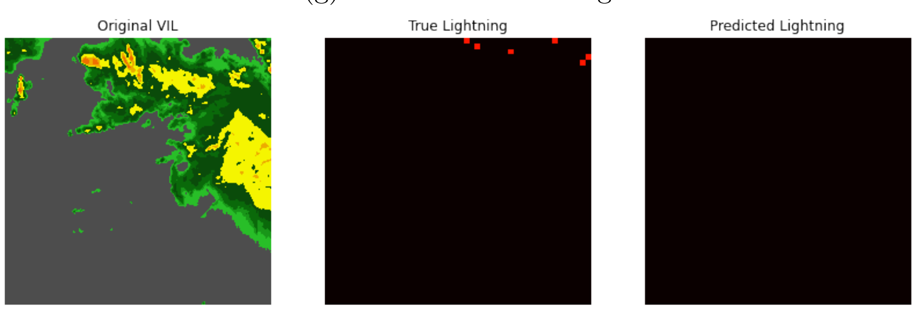
(e) 25-Minute Time Lag



(f) 30-Minute Time Lag



(g) 45-Minute Time Lag



(h) 1-Hour Time Lag

Figure 15: Example Prediction 5

This figure displays a sample prediction where the model falsely predicts the occurrence of no lightning in frames containing few and scattered lightning strikes.

Bibliography

1. Francis J Merceret, John C Willett, Hugh J Christian, James E Dye, E Phillip Krider, John T Madura, T Paul OBrien, W David Rust, and Richard L Walterscheid. A history of the lightning launch commit criteria and the lightning advisory panel for america's space program. Technical report, 2010.

2. Inclement weather. `https://www.faa.gov/newsroom/inclement-weather-0`, Mar 2021.

3. Air force doctrine publication 3-59 weather operations, Oct 2020.

4. Mark A Shafer, Donald R MacGorman, and Frederick H Carr. Cloud-to-ground lightning throughout the lifetime of a severe storm system in oklahoma. *Monthly Weather Review*, 128(6):1798–1816, 2000.

5. Iwan Holleman. *Hail detection using single-polarization radar*. Ministerie van Verkeer en Waterstaat, Koninklijk Nederlands Meteorologisch . . . , 2001.

6. Mingyue Lu, Yadong Zhang, Min Chen, Manzhu Yu, and Menglong Wang. Monitoring lightning location based on deep learning combined with multisource spatial data. *Remote Sensing*, 14(9):2200, 2022.

7. G Molinie and AR Jacobson. Cloud-to-ground lightning and cloud top brightness temperature over the contiguous united states. *Journal of Geophysical Research: Atmospheres*, 109(D13), 2004.

8. Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629, 2018.

9. Evan Racah, Christopher Beckham, Tegan Maharaj, Samira Ebrahimi Kahou, Mr Prabhat, and Chris Pal. Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. *Advances in neural information processing systems*, 30, 2017.

10. Lei Han, Juanzhen Sun, and Wei Zhang. Convolutional neural network for convective storm nowcasting using 3-d doppler weather radar data. *IEEE Transactions on Geoscience and Remote Sensing*, 58(2):1487–1495, 2019.

11. Shuchang Guo, Jinyan Wang, Ruhui Gan, Zhida Yang, and Yi Yang. Experimental study of cloud-to-ground lightning nowcasting with multisource data based on a video prediction method. *Remote Sensing*, 14(3):604, 2022.

12. Yangli-ao Geng, Qingyong Li, Tianyang Lin, Lei Jiang, Liangtao Xu, Dong Zheng, Wen Yao, Weitao Lyu, and Yijun Zhang. Lightnet: A dual spatiotemporal encoder network model for lightning prediction. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2439–2447, 2019.

13. Jaehong Yoon and Sung Ju Hwang. Combined group and exclusive sparsity for deep neural networks. In *International Conference on Machine Learning*, pages 3958–3966. PMLR, 2017.

14. Wei Wu, Qinwei Fan, Jacek M Zurada, Jian Wang, Dakun Yang, and Yan Liu. Batch gradient method with smoothing l1/2 regularization for training of feedforward neural networks. *Neural Networks*, 50:72–78, 2014.

15. Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

16. Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

17. Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.

18. Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

19. John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. *Advances in neural information processing systems*, 22, 2009.

20. Donald Goldfarb and Zhiwei Qin. Robust low-rank tensor recovery: Models and algorithms. *SIAM Journal on Matrix Analysis and Applications*, 35(1):225–253, 2014.

21. Yue Hu and Daniel B Work. Robust tensor recovery with fiber outliers for traffic events. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(1):1–27, 2020.

22. Hao Yan, Kamran Paynabar, and Jianjun Shi. Anomaly detection in images with smooth background via smooth-sparse decomposition. *Technometrics*, 59(1):102–114, 2017.

23. Hao Yan, Kamran Paynabar, and Jianjun Shi. Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition. *Technometrics*, 60(2):181–197, 2018.

24. Mark Veillette, Siddharth Samsi, and Chris Mattioli. Sevir: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. *Advances in Neural Information Processing Systems*, 33:22009–22019, 2020.

25. Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

26. Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

27. Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

28. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
23–03–2023	Master's Thesis	September 2021 — March 2023

4. TITLE AND SUBTITLE

Inducing Sparsity Within High-Dimensional Remote Sensing Modalities for Lightning Prediction

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Metzgar, Grace, 2nd Lt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENS-MS-23-M-146

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The uncertainty of lightning constantly threatens many weather-sensitive fields where the slightest presence of lightning can endanger valuable personnel and assets. The consequences of delaying operations have incited the research of methods that can accurately predict the location of future lightning strikes from the current weather conditions. High-dimensional remote sensing modalities contain information capable of detecting significant patterns and intensities within storms that could indicate the presence of lightning. This thesis induces sparsity into convolutional neural networks (CNNs) and remote sensing modalities through a combination of regularization and tensor decomposition techniques to call attention to sparse features that are most indicative of lightning activity. The developed models produce accurate predictions of the general pattern of true lightning strikes at lower time lags. The results demonstrate the potential of using CNNs in combination with sparse methods that focus on important features for the prediction of close-range lightning activity.

15. SUBJECT TERMS

convolutional neural network (CNN), deep learning, robust tensor decomposition (RTD), tensor decomposition, regularization, sparse CNN, lightning prediction

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Dr. Nathan Gaw, AFIT/ENS
					19b. TELEPHONE NUMBER *(include area code)*
U	U	U	UU	69	nathan.gaw@afit.edu