

Other

An Application of Image Processing Techniques in the Calibration of Catastrophe Models

Titus Kipkoech Rotich^{1a}, Artur Hambardzumyan^{2b}, Eliud Koech^{3c}, Olga Poghosyan⁴, Joseph Mungatu⁵

¹ Curtin University, ² BetConstruct, ³ Alupe University, ⁴ University of Canberra, ⁵ Jomo Kenyatta University of Agriculture & Technology, Nairobi, Kenya

Keywords: Artificial intelligence, Catastrophe modeling, Image processing, Machine learning, Neural networks, Semantic segmentation

<https://doi.org/10.66573/001c.146426>

Variance

Vol. 18, 2025

Actuaries depend primarily on simulations to build catastrophe (cat) models. By applying image processing techniques such as the novel convolutional neural network (CNN), it is possible to use both numeric and map data to improve modeling. To this end, we illustrated applying CNN to calibrate a cat model, using the more efficient U-Net architecture, which has been shown to perform well with limited data because of its localized predictive ability. We evaluated our CNN model using real-life data obtained from the National Oceanic and Atmospheric Administration. We also used these data to build a more traditional generalized linear model and compared the results with those from the CNN model. We found that even with limited data, the CNN model performed well.

Address for Correspondence: rktitus@uoeld.ac.ke

1. INTRODUCTION

Climate change¹ is one of humanity's most significant and adverse challenges of the 21st century—one of the "biggest threats modern humans have ever faced" (United Nations 2020). The resulting increase in catastrophic weather events over the last few decades has led to previously rare events now occurring more often. Without intervention, these natural disasters will increase to catastrophic levels by 2100 (Fawzy et al. 2020; Prudential Regulation Authority 2015). A recent study by Thistlethwaite et al. (2018) showed that, without intervention, the prevailing climate scenario could increase annual flood losses in Canada by as much as 300% by the end of the century. And this finding did not consider potential increased exposure.

Born (2019) showed that climate change is adversely affecting polar bears, but its impact is also evident in more expansive areas, such as observed increases in the prevalence of waterborne and vector-borne diseases and the im-

pact of natural disasters on human socioeconomic well-being (Watts, Amann, Arnell, et al. 2019). Every natural disaster has direct and indirect impacts on the physical infrastructure, including buildings and road networks, as well as on the social infrastructure, including health services, educational facilities, and the economy.

Additionally, varying exposures render some segments of the population, such as the elderly, children, and people with underlying health conditions, more vulnerable to climate change effects (Whitmore-Williams et al. 2017). Recent research has also found that psychological responses to climate change—what Clayton (2020) refers to as "climate anxiety"—are associated with increased mental health issues (Morganstein and Ursano 2020).

Insurance companies have a primary role in protecting against losses that may be devastating to individual entities. They do so by capitalizing on the law of large numbers—pooling individual risks allows them to indemnify their clients against devastating costs from covered perils.

a Kip Rotich is a senior lecturer in actuarial science at Curtin University. Formerly a lecturer at the University of Eldoret, his research spans mortality modeling and health policy, with a focus on applying actuarial methods in nontraditional areas.

b Artur Hambardzumyan is an analyst at BetConstruct, where he develops machine learning and AI algorithms to optimize and enrich the gaming experience.

c Eliud Koech is a biostatistics lecturer at Alupe University. His research focuses on longitudinal and survival analysis for type 2 diabetes. He is acting chair of the department of science mathematics education and a member of KNSS and the Bioinformatics Hub.

1 We use the United Nations' definition of climate change: "long-term shifts in temperatures and weather patterns, mainly caused by human activities, especially the burning of fossil fuels" (United Nations 2020).

Casualty actuaries have historically used generalized linear models (GLMs) to model weather-related event frequencies. GLMs are considered the best alternative for modeling count data (Wood 2006) and, consequently, a natural choice when modeling weather-related event frequencies. Research has illustrated their advantage over other models (O'Hara and Kotze 2010; Ives 2015). However, such numerical prediction models for weather-event frequencies are not only inappropriate when data are limited (Warton et al. 2016) but are also overly complex. They use high computing power and expensive hardware, making them an expensive venture (Trebing, Stańczyk, and Mehrkanoon 2021).

Most catastrophe models² are designed to simulate the physical phenomenon of various natural disasters by simulating several events and their associated probabilities and occurrence scenarios and then using these outcomes to estimate the expected loss for a given portfolio. This involves tracking a weather event's entire life cycle. For example, simulating typhoons requires developing a catastrophe model that begins tracking a typhoon as soon as a depression system forms in the ocean, then continuing to track as the typhoon travels toward land and across several countries. Data collection must include the type of impact, potential damages, and financial losses incurred. In addition to the financial risk aspects of catastrophes, it is essential to consider other factors, such as resiliency, emergency planning, and asset management. To do this, scientists rely on climate science, data quality, and catastrophe modeling.

However, catastrophic weather-related events are now occurring more often than the current catastrophe (cat) models can predict (Rotich 2017), leading to a capital base within the reinsurance sector that has increased to a record high of \$605 billion (Seekings and Anderson 2017). Despite this high capital base, the losses incurred from catastrophes have been immense. For example, the aftermath of hurricanes Irma and Harvey, which were devastating, dealt a sharp blow to the reinsurance industry, with the worst losses ever recorded, totaling slightly under \$200 billion. Despite the over-capitalization of the reinsurance sector, it will be challenging to cover losses from such extreme events if reinsurers continue using their current models (Seekings and Anderson 2017).

It is essential to note three things. First, the ballooning insured losses have not emanated solely from the increased frequency of catastrophes; new entrants into the insurance sector, such as agriculture, have also contributed (Jørgensen, Termansen, and Pascual 2020). Second, the increased frequency of catastrophes can be attributed to both climate change and improved reporting (World Meteorological Organisation 2021); however, the number of deaths has

decreased due to improved early warnings and better disaster management. Finally, the increasingly devastating effects of these disasters are primarily driven by population growth and distribution. For example, high-density urban cities are more vulnerable to devastating disasters (Donner and Rodriguez 2011).

Even with sophisticated cat models, temporal and spatial uncertainty are still significant factors (Thistlethwaite and Wood 2018), partly because of limited data. Even with increasing frequencies, rare events are not frequent enough to generate sufficient data. The biggest challenges for the cat models are climate change hazards, including accurate representation of climate change uncertainty, modeling correlations between perils, modeling turbulent processes in the atmospheric boundary layer and other weather-related perils, and so forth (Jewson, Herweijer, and Khare 2016). Therefore, methodologies that can produce accurate results with limited data are vital assets for an insurer's claim management arsenal. These methodologies can be used to build internal models, which, in turn, can be used to calibrate the vendor-supplied cat models; or the outputs can be used for validation to help establish an accurate representation of the probability distribution of losses.

Sousounis (2021) established that industry concerns are rooted in three main areas:

1. Climate change leads to significant alterations of weather-event frequencies and intensity. However, cat models are built on long-term historical datasets, which do not address recent changes.
2. Cat models also lack a straightforward method, such as a scaling factor, that can account for the impact of climate change.
3. Climate change will continue to drive adverse weather effects.

Therefore, models that can be used to generate early warnings play an increasingly significant role in disaster and loss management.

Recent studies have alluded to the potential for advances in artificial intelligence to solve some of these shortcomings. In a comprehensive study on how machine learning (ML) techniques can be used to tackle the climate change problem, Rolnick, Donti, Kaack, et al. (2019) presented possible model architectures³ for modeling weather-related event frequencies. The authors explored ML applications in various areas, including modeling, forecasting, and building predictive ML models. They also illustrated various applications for different ML techniques, such as convolutional neural networks (CNN) (LeCun, Boser, Denker, et al. 1989).

² A catastrophe model is a computer simulation of potential disasters that estimates the losses likely to be inflicted by an event. The model analyzes how frequent and severe the predicted disasters/catastrophes are likely to be, where they are likely to occur, and the magnitude of damages they are likely to cause. We use the terms *catastrophes* and *disasters* interchangeably.

³ Neural network architecture (or just *architecture* for short) refers to the arrangement of the deep learning tools used for modeling.

CNN models use images as inputs (see Section 2.2 and Appendix C for more details on CNN). Table B1 in Appendix B summarizes previous studies that investigated using ML applications for weather-event modeling, including the inputs used to build the respective models, the network models used, and the performance measures used to evaluate the models.

Briefly, CNNs comprise a branch of deep learning with multilayered neural networks that use ImageNet datasets with millions of labeled images and abundant computing resources to analyze visual imagery. CNN uses convolution, the mathematical operation of two functions that produce a third function, to show how the shape of one is modified by another (Mandal 2021).

Studies have shown CNN's potential effectiveness for thresholding and clustering satellite images into grids to identify disaster-affected areas (Doshi, Basu, and Pang 2018). Liu, Racah, Correa, et al. (2016) found that CNN accurately characterizes extreme weather events using a broad set of patterns from labeled data.

Our study aimed to illustrate the application of image processing techniques in modeling weather-event frequencies. In particular, we illustrate CNN's ability to model weather-event frequency when numerical data are unavailable or limited. Additionally, as shown by Scher and Messori (2018), assigning a scalar value of confidence to medium-range forecasts initialized from an atmospheric state allows us to estimate future forecast uncertainty from past forecasts. Actuaries can use CNN techniques to calibrate commercial or third-party cat models and depict increased frequencies based on more recent, expensed, and focused satellite images. They can also use the results to validate output from cat models, especially where data are limited, as they are in most cases. Additionally, these techniques have a potential application for providing early warnings to bolster disaster preparedness, which, as highlighted earlier, plays a significant role in minimizing disaster-related deaths.

Introducing ML techniques into the catastrophe models used in weather-event frequency studies may lead to improved models. Ghosh, Bergman, and Dodov (2020) reported that an exceptional feature of ML, particularly CNN, is its potential to detect and classify features in imagery. The authors found that the CNN algorithm can be trained on a set of previously identified high-rise buildings to execute visual identification in aerial imagery by identifying shadows, patterns, and shapes found in the training data. This suggests that CNN models can be designed to use satellite imagery in the absence of sufficient raw data. Based on this premise, we built CNN models by applying Ronneberger et al.'s (2015) U-Net architecture and compared our model's results with those from more traditional GLMs.

Our study's contribution is threefold. First, to the best of our knowledge, this is the first study to apply image pro-

cessing using U-Net architecture to build a model for catastrophic events. Also, to the best of our knowledge, our study is the first to evaluate the predictive power of U-Net and GLMs. We not only evaluated the U-Net architecture but also investigated its effectiveness using sparse data to test the claim that it performs significantly better than other models.

Second, given the increasingly broad impact of natural disasters on communities, accurate modeling of such events will increase financial cushioning and preparedness. Our study results may therefore reduce the impact of such events on humans' socioeconomic well-being. Finally, by applying a CNN architecture, our model extracts and uses the temporal and spatial dependencies within the data, improving the model's effectiveness in predicting weather-related catastrophic events.

This study does not explore the complete cycle of catastrophe modeling, which would involve scenario analysis and stress tests to establish the bounds of an event's impact. The study used weather-event frequency and severity⁴ as target variables and longitude and latitude data as inputs. To measure model performance, we used performance measures such as mean absolute error (MAE), accuracy false alarm rate (FAR), and F_1 scores, among other measures, as discussed in Section 5.5.

The rest of the paper is organized as follows: Section 2 describes the study's methodologies, including neural network, CNN, U-Net, and GLM. Section 3 describes the study data. Section 4 describes the exploratory data analysis, followed by a description of the methods and models fitted and the model evaluation methods in Section 5. Results are then presented in Section 6, and the paper concludes with a discussion of the results and concluding remarks in Section 7.

2. METHODS

GLMs have long been part of an actuary's assortment of modeling tools for solving various problems (Goldburd et al. 2025). However, neural networks have been gaining traction owing to their ability to more closely mimic reality. This study evaluates the CNN model, particularly the U-Net architecture. Understanding how a basic artificial neural network (ANN) works is essential to understanding CNN. Therefore, we begin by discussing a simplified ANN architecture and how it is used for modeling. Since neural networks are often viewed as black boxes (Z. Zhang et al. 2018), we demystify them in Appendix A by providing a high-level summary of similarities and differences between ANNs and GLMs.

⁴ Whenever this information is available.

2.1. ARTIFICIAL NEURAL NETWORK

ANNs (or just neural networks for short), have been applied in many different fields after first being discussed in McCulloch and Pitts's (1943) study on computational models for neural networks. Werbos's (1974)⁵ study introducing back-propagation (i.e., backward propagation of errors) sparked renewed interest. Thought to belong to the nonparametric class, ANNs have been widely adopted because of their ability to "learn" the input data's behavior and thereby provide more accurate output (England 2008). [Figure 1](#) shows a simplified representation of how a neural network works (see Guresen and Kayakutlu 2011; Z. Zhang et al. 2018; Zupan 1994; and the references therein for a more detailed discussion on neural networks, including their mathematical formulations).

Briefly, a layer is made up of nodes. The arrows represent the associations and/or the direction of the association. The input layer receives data to be analyzed, a single layer where the number of nodes equals the number of input variables, which can associate with more than one node on the hidden layer. The hidden layer manipulates the data and relays the result to the output layer. The number of layers depends on the problem being analyzed, with the number of nodes in each layer equal to the number of data categories (factors).

In most cases, a single hidden layer is optimal (Lowe and Pryor 1996). Weights are then assigned to the different associations between adjacent layers. If there is no hidden layer, it is a linear regression model. The output layer is composed of a single node that relays the model's output. However, it is possible to have more than one ANN ensemble so that the output layer has more than one output.

In a backpropagation ANN model,⁶ modeling typically starts with an initial known output value called the "seed." Initial weights w_{ij} defining the relationship between nodes i and j of two adjacent layers are defined, including initial biases. Using the input data and the "seed," the model readjusts the weights and associated biases. This self-calibration is called model training. The training stops by reaching either a specified number of iterations or minimum variance. Making the error zero will likely lead to overfitting; therefore, best practice is to minimize the variance.

Despite the efforts to explain what happens in the hidden layer, ANN models remain a "black box" to many (Z. Zhang et al. 2018)—often because there is no clear relationship between the inputs and the model's output. This makes it difficult to understand the output and to interpret the results. Consequently, checking the model's validity and deciding on its appropriate structure (e.g., the number of hidden layers) can be challenging. Some argue that con-

structing an ANN model is subjective. The main concern remains overfitting.

2.2. CONVOLUTIONAL NEURAL NETWORK

A CNN is another architecture used for deep learning exercises, where the inputs are images.⁷ First introduced by LeCun, Boser, Denker, et al. (1989), CNNs have outperformed standard feed-forward classification neural networks because they can extract and use the temporal and spatial dependencies in the input image. CNNs are anchored on the principle that any image can be considered a binary representation of features of underlying data. It can therefore be broken down into smaller building blocks (pixels), which are arranged on a grid and contain information about the image's characteristics (see [Figure 2](#)).

Just like ANN, CNN can assign weights and biases to the features defining the image; however, CNN does this with minimal preprocessing. Typically, CNN comprises convolutional, hidden, and fully connected layers, as shown in [Figure 3](#). The convolutional layer does most of the computation. First, it takes a matrix of features to be learned by the network and the matrix of the restricted receptive field section and performs a dot product of the two. The restricted receptive field section is also called the local receptive field. Contrary to ANN, where each node from the input layer is connected to another node in the hidden layer, with CNN, only the local receptive field connects to the hidden layer. To generate the local receptive field feature matrix, the image from the input layer is first converted to a feature map, which is then relayed to the hidden layer. This process is efficiently implemented through convolution.

Similar to ANN, CNN training involves estimating weights and biases, which are continuously updated during the training process. Unlike ANN, these weights and biases are unchanged for a given hidden layer; therefore, the model is effectively tolerant to any image feature translation and will recognize these features whenever they are present. This is possible because every node within the hidden layer is detecting (learning) the same image feature. Consequently, a model with more hidden layers can learn more features. This explains why a CNN model can have a substantially larger number of hidden layers than an ANN model.

Finally, the output from each neuron is transformed using an activation function by taking these outputs and mapping them to the highest value. However, if the highest value is negative, the transformation process is repeated with a condensed output. Through a process known as pooling, the output from smaller parts is condensed into a single output by reducing the dimension of the feature

⁵ Werbos's 1974 study has been lauded for making multilayer neural network training possible.

⁶ The backpropagation ANN model is also commonly referred to as *supervised learning*.

⁷ More generally, the inputs are typically data elements that have a grid-like topology. Images are a good example and are the most commonly used CNN inputs.

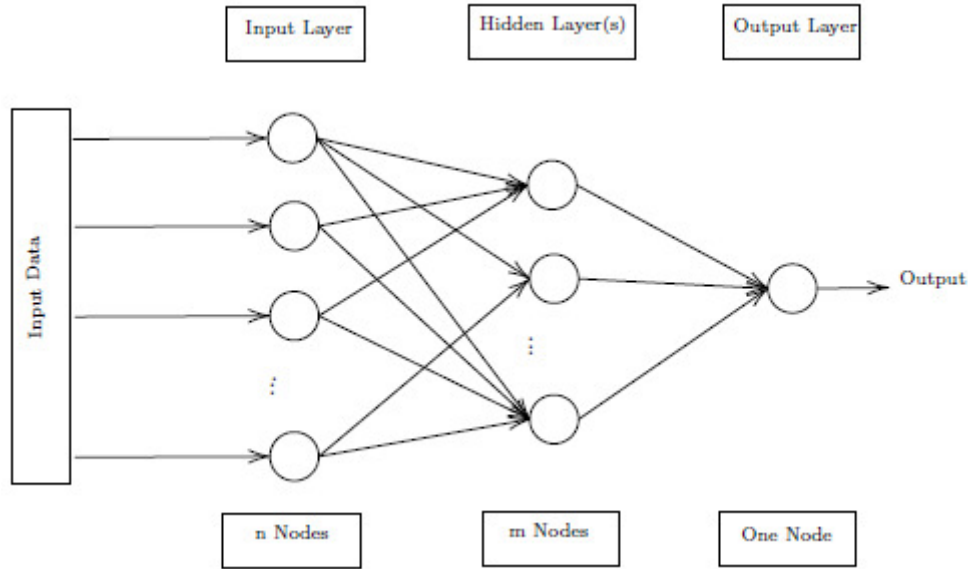


Figure 1. A simplified example of an ANN with a single hidden layer (adapted from Lowe and Pryor 1996).

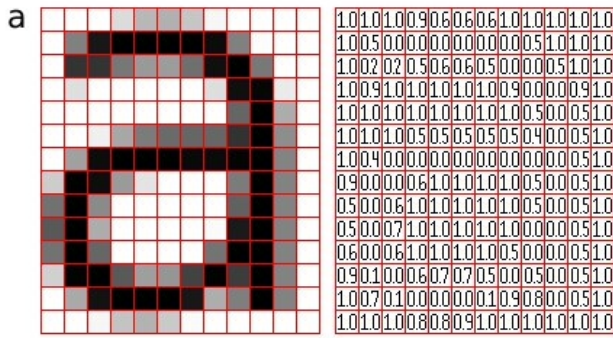


Figure 2. An illustration of an image as a pixel. CNN works by localizing its prediction to each square of the image grid (pixel) (adopted from Mishra 2020).

map, which helps reduce the parameters to be learned and hence the run time. The final hidden layer then connects all its nodes to the final layers, collating the feature output to form the final image (see H hlele et al. 2020; Liu, Racah, Correa, et al. 2016; Stankovic and Mandic 2021; and their accompanying references for a detailed discussion on CNN, including their mathematical formulations and their application in extreme weather modeling).

2.3. CONVOLUTION NEURAL NETWORK (U-NET ARCHITECTURE)

U-Net architecture is specific to CNN and designed for semantic segmentation.⁸ Developed by Ronneberger, Fischer, and Brox (2015) to be used for biomedical image segmentation, CNN U-Net (henceforth referred to as U-Net) has since been widely adopted in various fields (see, for example, Du et al. 2020; Sadeghi et al. 2020; Yao et al. 2018). As CNN architecture, U-Net has all the CNN properties plus the ability to localize and differentiate borders by classifying images based on each pixel, making the input and output have the same properties. Figure 4 shows a U-Net architecture. (For a more detailed description of the architecture and the accompanying mathematical formulations, see Ronneberger, Fischer, and Brox 2015.)

U-Net architecture is symmetrical with two main parts: the encoder⁹ and decoder¹⁰ paths. The encoder path comprises a convolutional process described in Section 2.2. The decoder path comprises two-dimensional convolutional transposition.

First, the image enters the input layer from the left side of Figure 4. Convolutional operations are applied using two convolutional layers. Finally, the number of features to be mapped is specified. A process similar to the CNN model uses a three-dimensional feature matrix, commonly re-

⁸ Semantic segmentation uses pixels to paint and identify specific image features; this differs from object detection and instance segmentation. Object detection is essentially a cropping exercise, where the objects are detected by cropping out the parts of the image not occupied by the object in the image. Instance segmentation goes beyond semantic segmentation to identify the individual objects in an image.

⁹ Also known as the contracting path.

¹⁰ Also called the expansive path.

ferred to as the kernel. The final convolutional layer comprises padding.¹¹ Next, pooling is implemented, reducing the image size by half. Pooling enables the model to focus on the most critical features of the input images by generating a two-by-two kernel, then using the previous output to select the max value to populate the two-by-two kernel. This process is repeated three times to reach the bottom of the architecture, which concludes the encoder path. The bottom convolutional operation differs from the previous operations in that the pooling operation is not applied, so the image size remains the same.

The decoder path is the reverse of the encoder path. Here, the image sizes are expanded through fractionally strided convolution by applying a padding exercise and then a convolutional operation. After the image size has been expanded, it is concatenated with the image from the encoder path.¹² Combining the information in the expanded image with that from the previous layer allows the U-Net to produce a prediction superior to other CNN architectures. Similar to the encoder path, the process is repeated three times. The last step involves reshaping the final image to meet the prediction requirements specified by the modeler. The final layer is always a single filter convolutional layer (see Ronneberger, Fischer, and Brox 2015, for a more detailed layer-by-layer description).

2.4. GENERALIZED LINEAR MODELS

GLMs, popularized by McCullagh and Nelder (1989), model the relationship between a response variable and a predictor variable. GLMs have become a staple in the casualty actuary's toolbox. The models assume that systematic and random components drive the response variable, so a GLM's purpose is to explain as much of the systematic component as possible. It is assumed that the dependent vari-

able is drawn from a distribution of the exponential family with a mean

$$E[\mathbf{Y} | \mathbf{X}] = g^{-1}(\mathbf{X}\boldsymbol{\beta}), \quad (2.1)$$

where $\mathbf{Y}$ and $\mathbf{X}$ are the response and predictor variables, respectively. The modeling then reduces to the specification of the link function, g , and the modeling of the linear predictor

$$\eta = \mathbf{X}\boldsymbol{\beta}. \quad (2.2)$$

Our study focused on modeling weather-event frequency, which was our response variable. We assumed that these events follow a Poisson distribution, so we could model the mean using equation (2.1). If we assume a log link function, the linear predictor can then be modeled as

$$\log \lambda_i = \beta_0 + \beta x_i, \quad (2.3)$$

where λ is the Poisson distribution mean. Or equivalently,

$$g(\lambda_i) = \beta_0 + \beta x_i. \quad (2.4)$$

For a comprehensive review of GLMs, including their extensions, see Dobson (2002), Dunn and Smyth (2018), Hardin and Hilbe (2006), and McCullagh and Nelder (1989).

3. DATA

We obtained the study data from the National Oceanic and Atmospheric Administration (NOAA 2020), a government agency within the US Department of Commerce that provides information on the state of the oceans and atmosphere, including weather warnings and forecasts. We obtained historical data for storm events from 1950 to 2020 and used the Python programming language (Van Rossum and Drake 2009) to scrape data in a zipped format and convert it to comma-separated value (CSV) files, the most commonly used format due to ease of use (Hernández, Somnath, Yin, et al. 2020). Three files were generated for each year:

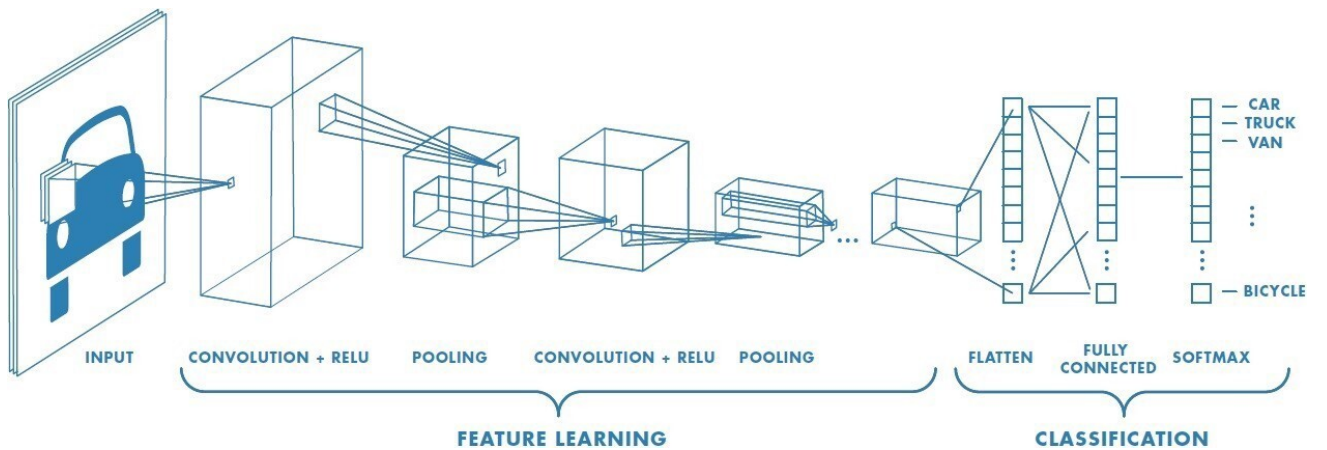


Figure 3. An illustration of CNN architecture using feature segmentation to identify a car in an input image (adopted from Saha 2018).

¹¹ Essentially, this is cropping out unnecessary parts of the input images.

¹² This process is commonly known as concatenation of feature maps.

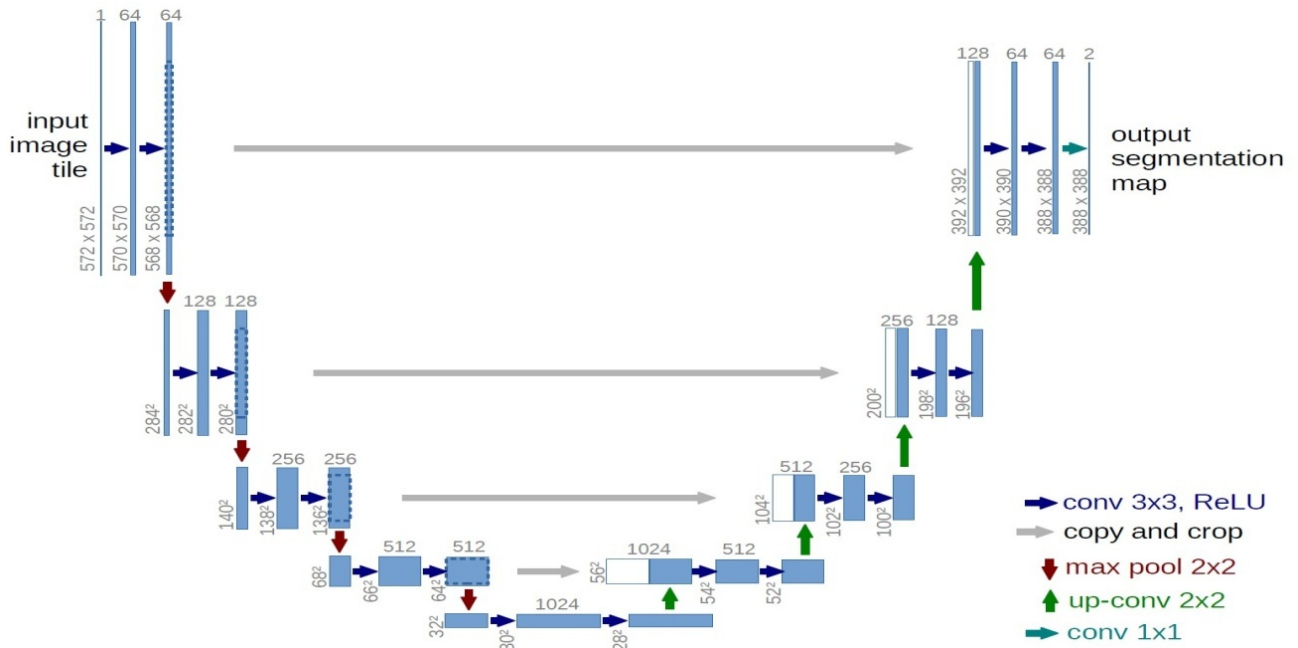


Figure 4. U-Net architecture (adopted from Ronneberger, Fischer, and Brox 2015).

1. Storm event details
2. Storm event locations
3. Storm event losses, including fatalities caused by the storm event

We collated the three files, excluded data with missing location information, and generated an all-years dataset, which was filtered to only include the data needed for modeling. We used the data to generate exact event dates for modeling purposes.

All analyses were conducted using Python, which provides modern tools developed for ML problems (Sarkar, Bali, and Sharma 2018). All work presented in this paper is reproducible; the codes, data, and analyses we used are available on GitHub.¹³

4. EXPLORATORY DATA ANALYSIS

The downloaded data contained a range of information relating to each weather event (see NOAA 2020, for a complete list and description of the data types). However, we only retained the information relevant to this study. One challenge we faced was inconsistency in data types and format over multiple years. For example, a million dollar amount was recorded as \$1,000,000 in some years and \$1000000.00 or \$1m in other years, which required sub-

stantial data cleaning and correction prior to data visualization and analysis. Since we intended to use heatmaps to visualize and analyze the data, latitude and longitude data were critical because they provided each event's exact location. Therefore, events that were missing latitude and/or longitude data were excluded from the analyses. We used Python's DateTime library to generate the exact date of each storm event and its duration.

The next step involved data visualization. Figure 5 shows how the total number of yearly events evolved over the study period. Figure 5a shows that the number of annual storm events increased substantially between 1950 and 2020.¹⁴ Additionally, the event numbers demonstrate a cyclical pattern, where an above-average number of events occurred every three to six years, but this cyclical pattern appears to be decreasing over time—that is, increased numbers of annual events occurred more frequently in recent years compared with earlier years.

In other words, the frequency of extreme weather events has consistently increased over the years. Furthermore, the number of annual events has more than doubled over the last two decades.

Figure 5b shows the cumulative number of events occurring in each state. It shows that most extreme weather events occurred in four US states: Texas, Kansas, Oklahoma, and Missouri, and that the eastern states are more suscep-

¹³ <https://github.com/rktitus/Variance>

¹⁴ As noted in Section 1, this area of research has been subject to reporting bias. Prein and Holland (2018) found that a combination of increased population density and increased wealth is responsible for escalating property losses from hailstorms, and that report-based observational data have failed to discriminate between climate and societal changes. However, when the focus is disaster management and response, whether increasing losses are driven by societal change, population change, climate change, or a combination of these is not a primary concern from a modeling perspective.

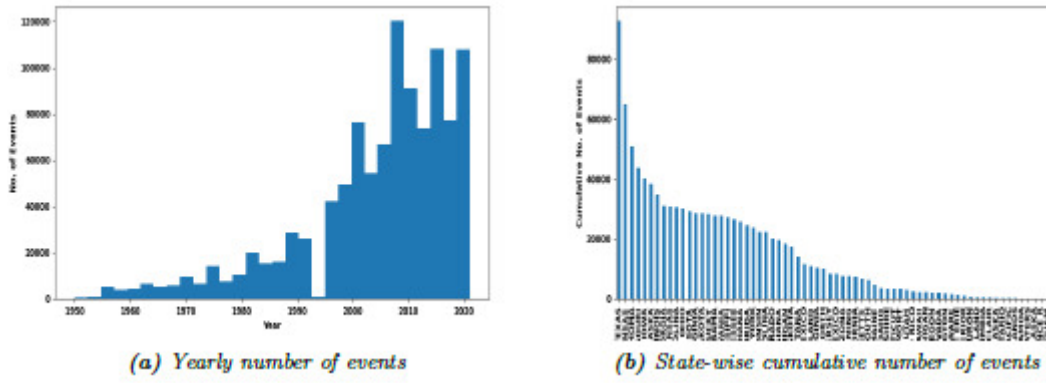


Figure 5. Total number of events occurring per year (a), and cumulative numbers of events that occurred in each US state between 1950 and 2020 (b).

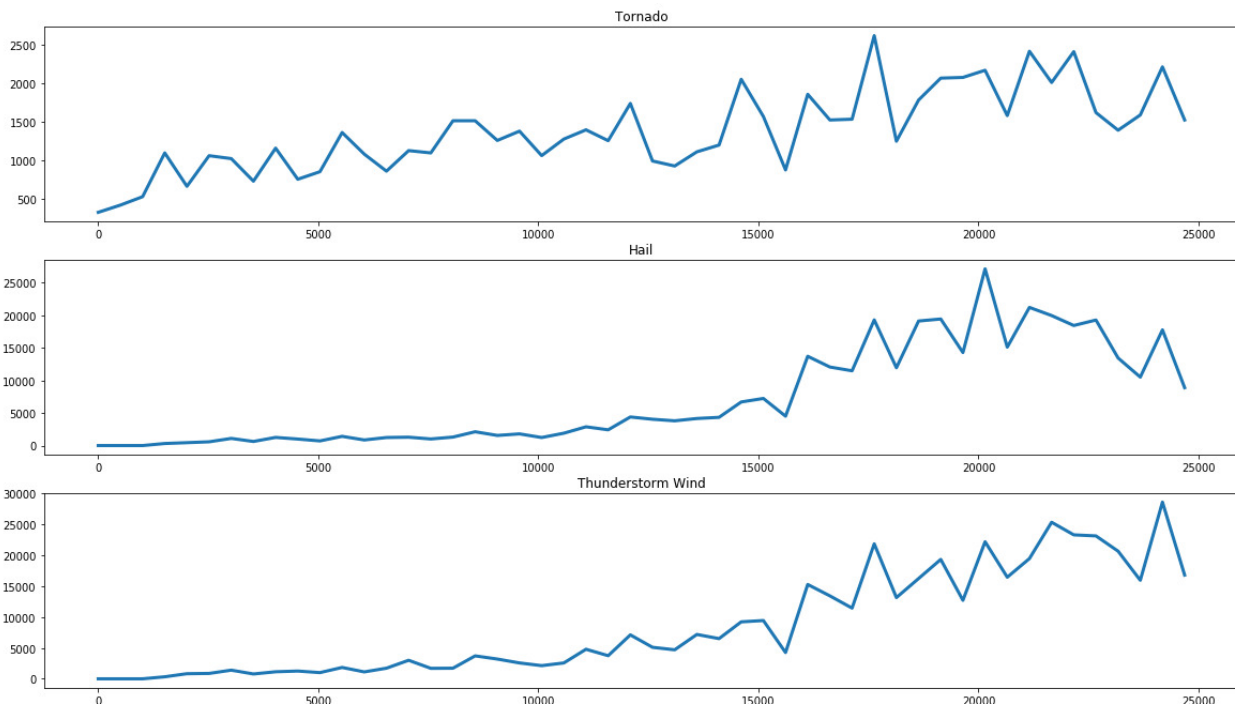


Figure 6. Plot of historical frequencies for the three major weather events in the US from 1950 to 2020.

tible to extreme weather events. However, it is essential to note that the intensity of these weather events were not considered in Figure 5b. Therefore, events that may occur over several days, such as wildfires, are counted as a single event, without considering the number of days a fire continued to burn, for example.

Figure 6 shows weather-event frequencies between 1950 and 2020 by type (tornado, hail, and thunderstorm wind). Conclusions drawn from Figure 6 are similar to those drawn from Figure 5.

Figure 7a shows that thunderstorm winds and hail are the two most prominent extreme events in the US. Figure 7b illustrates hail severity using a heatmap generated from

the longitude and latitude coordinates in the dataset that provided the exact location of weather events on a US map.¹⁵ This process required linking map limit coordinates to image pixel coordinates. For this study, we used 1 LON degree = 28 pixels and 1 LAT degree = 37 pixels, and we used a small point of 4×4 pixels to plot the events on a US map to produce a very detailed event map.

The plot shades reflect event intensity, where darker areas indicate higher event intensity. For example, Figure 7b illustrates the distribution of hail in the US over the study period, showing that hail events are concentrated in the eastern part of the US, corroborating the observation in Figure 5b. Maps can be generated to show the distribution

¹⁵ Note that the coordinates may vary on different map images.

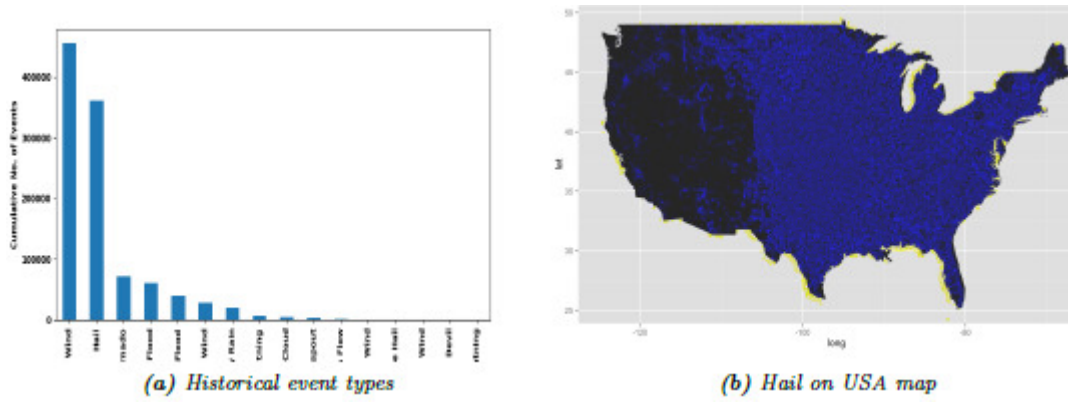


Figure 7. Total number of events occurring per year (a), and cumulative number of events that occurred in a US state between 1950 and 2020 (b).

of events by various factors, (e.g., per state/per year, per year/per event type/per state, etc.).¹⁶ Heatmaps can also be generated for any unit of time per event and can use animated display libraries to generate dynamic heatmaps for each event type.¹⁷

To visualize these weather events, we generated heatmaps using the longitude and latitude coordinates to place the event's exact location and magnitude¹⁸ to capture event intensity. These outputs can be found on GitHub.

5. THE MODELS

This study illustrates how image processing¹⁹ can be used to model the frequency and severity of catastrophic weather events when data are minimal or when only satellite images are available. Since cat models use data to produce simulations, the output data from our model can be used by actuaries to compare with external cat model output and hence can be used to calibrate cat models. Our model inputs are, therefore, images (maps) of the weather events to be modeled, which were generated using weather-event data to plot two-dimensional histograms.

5.1. TWO-DIMENSIONAL HISTOGRAMS

To set the scene, we introduce the concept of two-dimensional (2D) histograms. These are generalizations of the usual one-dimensional histograms with an extra dimension to show data density at any point in the usual (x, y) plane, also known as density map, density heatmap, or heatmap. These density maps are produced by grouping the (x, y) data points into bins based on their coordinates and then apply-

ing an aggregation function²⁰ to these groupings to calculate the tile color used to represent this bin on the map.

Essentially, a 2D histogram cuts the space into narrow ranges, like $[x1, x2]$, $[y1, y2]$, and makes a pixel for each range, putting there all the data points located in the given range, which results in a "heatmap image" with size $[\text{number_of_ranges_x}, \text{number_of_ranges_y}]$. For example, Figure 8 shows 2D histograms for thunderstorm wind and hail. These are $[100, 100]$ sized histograms with the coordinates constrained to $([-130, -64], [24, 50])$. Comparing Figures 8b and 7b shows that a 2D histogram can be a good alternative for visualizing such events. The two graphs closely resemble each other, except that Figure 7b was generated by superimposing the event data points on a blank US map.

Using artificial intelligence advances in image processing, these images can be fed into a model as inputs to model and project future (next) image(s). A CNN model is an example of this. Given that "a picture is worth a thousand words," such model outputs would be a lot easier to communicate. Additionally, the process is more effective because it negates the need to build a separate model to predict when the events will occur; only the number of events occurring over the prediction period need to be predicted. This makes the problem relatively easy to solve using ML and is less resource intensive, because most insurers are more interested in aggregate claims not exceeding a specific limit.

5.2. MODEL STRUCTURE

We constructed six-month rolling periods for model validation. For example, we fitted the model using data from January 1, 1950, to December 31, 2000 to predict the next six

¹⁶ See GitHub code for more details.

¹⁷ This has also been done. See the animation here: <https://rktitus.github.io/index.html>

¹⁸ Where magnitude data were available.

¹⁹ Using a CNN model with a U-Net architecture.

²⁰ For example, the summation function.

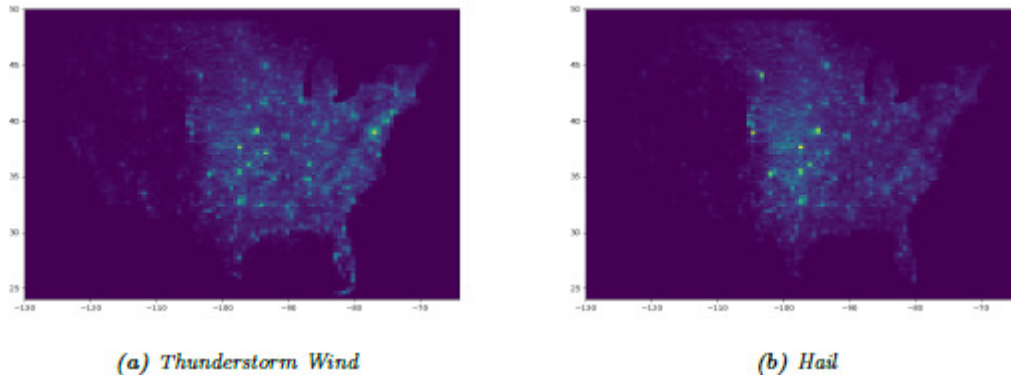


Figure 8. Total numbers of events occurring per year (a), and cumulative numbers of events that occurred in a US state between 1950 and 2020 (b).

months, then refitted the model using data from January 1, 1950, to January 31, 2001 to predict the next six months, and so on. For each of these periods, we generated density maps for the significant event types,²¹ such as those shown in Figure 8, which were then used to predict future density maps. It is important to note that the bigger the density map,²² the more accurate and detailed the map would be, but the more difficult it would be for the model to predict because of the additional pixels, and accuracy is also affected.

5.3. FITTING CONVOLUTION NEURAL NETWORK (U-NET ARCHITECTURE)

To fit a U-Net, we used the 2D histograms as model inputs. A total of 25,192 2D histograms were produced for each event type, corresponding to the number of days within the study period under consideration. We used image preprocessing to crop the images. Each pixel corresponds to the cumulative incidence of a specific event on a specified day within approximately two square miles.

For events such as earthquakes, the third dimension can represent the intensity. We used 20,153 images to train the model (training dataset), while the remaining 5,039 were used to test the trained model (testing dataset). A maximum of 1,000 epochs²³ were used to train the U-Net model. We then defined a rule to establish optimal epochs for each model, using a stopping rule if there was no increase in the loss function over the preceding 10 epochs. To achieve this, we used an Adam optimizer (Kingma and Ba 2014) with an initial learning rate of 0.001. To further improve model training, we used randomly selected images for every epoch to validate the model.

Three events in our dataset had complete data for all the years: tornadoes, hail, and thunderstorm wind (see Figure 6), so we used these events to fit the model. Therefore, the

input to the U-Net model presented in Figure 9 is a set of three images corresponding to these three events, which forms the three channels. A U-Net advantage is its ability to take more than a single layer as input.

We used the training dataset to fit the model. The density maps model input had a dimension of 50×20 . As shown in Figure 9, our model consisted of two convolutional operations. We then used four feature extractors (filters) to extract the essential features from the density maps. Each filter²⁴ was $3 \times 3 \times 3$, where the last 3 is the number of channels in the model.

We used a rectified linear unit (ReLU) activation function and a batch size of 32. No padding was applied to avoid overcropping the density maps. We also used a stride size of 2. After the two convolutional operations were implemented, we performed a max pool operation to provide a focus on “what” is being modeled. In our density maps, this equates to zooming in on specific regions of the map to accurately model the number of events occurring within each pixel. In doing this, we lost the “where” of the occurrences of the events. By zooming in, we reduced the size of the density map. This operation was repeated until the last encoder operation of the model was reached.

The symmetric expanding path does the opposite of what the encoder path does. The concatenation operation shown by the gray arrow in Figure 9 is essential; it recovers the “where” of the density map. By up sampling, we walked back to the original size of the density map. Therefore, our model output was also a 50×20 density map.

5.4. FITTING A GLM

We fitted a traditional GLM to model event count, which we compared with the CNN model. Numerical data were used to fit a GLM model and the model used for forecasting, while the CNN is fitted using the generated images and the

²¹ That is, thunderstorm winds, hail, and tornadoes. See Figures 5 and 6.

²² In terms of resolution.

²³ In machine learning, *epoch* refers to the number of iterations applied to the training dataset.

²⁴ Also known as *kernel*.

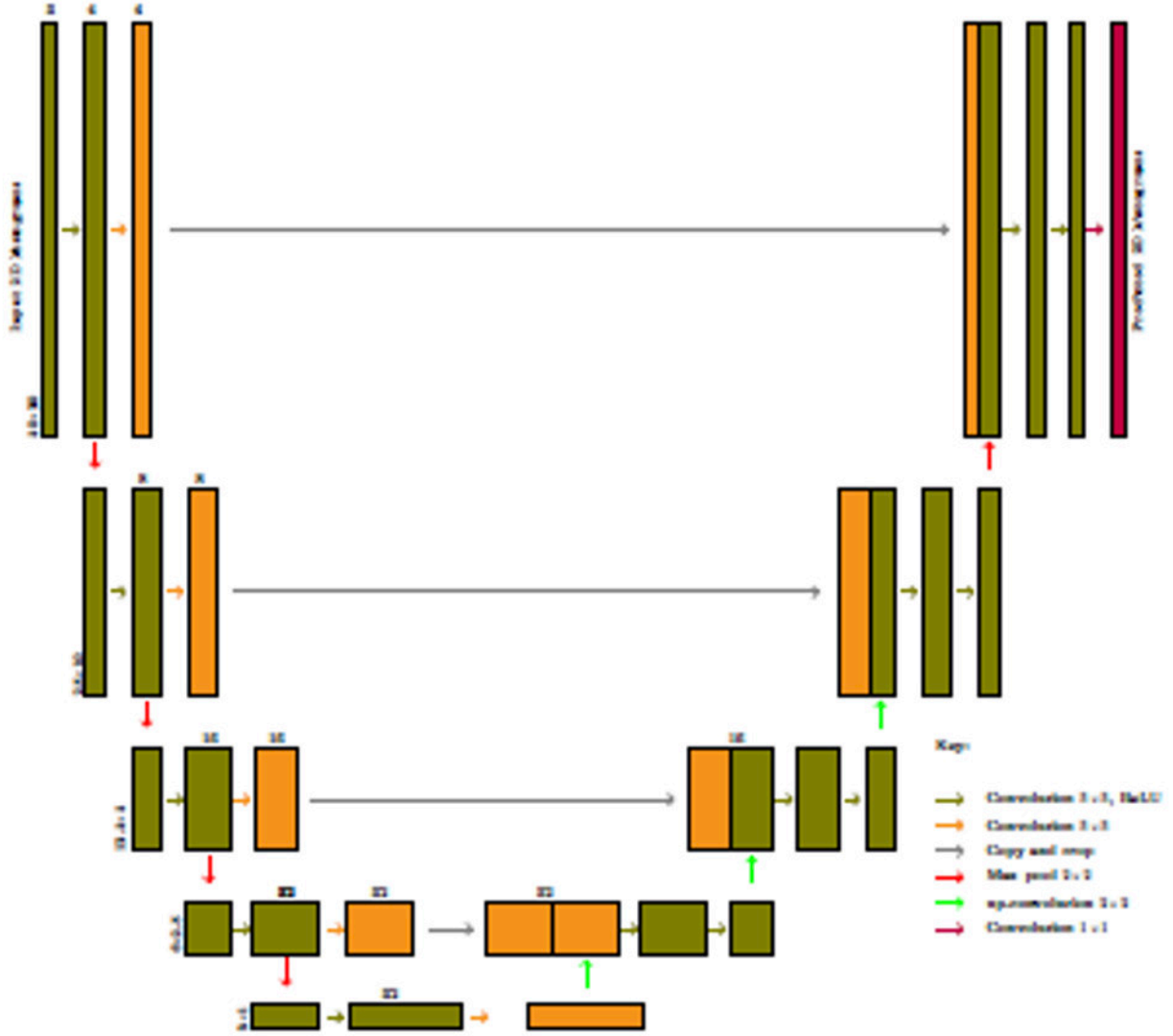


Figure 9. A visual representation of the U-Net fitted.

fitted model used for forecasting. For the GLM, we modeled event occurrences per day, per event type. It is impossible to model the number of events for exact locations, so the data were aggregated based on state. Consequently, we needed to build (number of event types) $\times$ (number of states) GLMs for event counts.

5.5. MODEL EVALUATION

We adopted MAE as the loss function, which calculates the average magnitude of the error between the predicted image outputs and the actual images. MAE was calculated as

$$MAE = \frac{\sum_{i=1}^n y_i - \hat{y}_i}{n}, \quad (5.1)$$

where n is the number of images used to fit the model, y_i are the actual images, and $\hat{y}_i$ are the image outputs predicted by the model. In addition, we calculated the pre-

cision,²⁵ accuracy FAR, F_1 score, and recall of the models. Precision was defined as

$$\text{Precision} = \frac{TP}{TP + FP'} \quad (5.2)$$

where true positives are correct predictions, while false positives are where the model predicts an event that actually does not occur. Accuracy was defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.3)$$

where true negative is when the model truly predicts the absence of an event, and false positive is where the model predicts the absence of an event when an event actually occurs.

Similarly, the FAR was calculated as

$$FAR = \frac{FP}{TP + FP} \quad (5.4)$$

The recall was estimated as

²⁵ In machine learning, *precision* measures the quality of the positive predictions produced by the model.

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (5.5)$$

Finally, the F_1 score was calculated as

$$\begin{aligned} F_1 &= \frac{TP}{TP + 0.5(FP + FN)} \\ &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \end{aligned} \quad (5.6)$$

6. RESULTS

Using Keras’s least absolute deviations regularization loss function (L1) to prevent overfitting, we fitted a U-Net model using the generated density maps. The input was an image, and the output was also an image of the same size and shape. A notable advantage of U-Net is that any image size can be used as an input since it is an end-to-end, fully convolutional network. This U-Net versatility is handy when modeling weather events where the images cropped to include only nonempty parts form an essential part of the modeling process.

We split the input data into test and training data, where the training data were used to train the model, and the test data were used to cross validate the model. This paper only presents results for hail. Similar conclusions can be drawn for both thunderstorm wind and tornadoes. The model’s accuracy can be improved by expanding the maps to contain cumulative events within a larger region, say a state. This would be useful if we were more interested in the aggregates of, say, state rather than the event’s actual location. This could provide useful data for state government preparedness.

MAE does not penalize outliers as heavily as root mean squared error (RMSE) or MSE. Therefore, we chose MAE to evaluate the fitted models. We selected the models that achieved the lowest validation loss. [Figure 10](#) shows how the two models performed on the training dataset.

The first row represents the actual density maps, the second row represents the U-Net predictions, and the last row represents the GLM predictions. The images show that both models performed well. We then used these two models on the test dataset and produced similar graphs (see [Figure 11](#)). Comparing [Figures 10](#) and [11](#) shows that the models seem to perform similarly on both training and test datasets. This indicates an absence of overfitting the data.

We used these models to calculate various measures, as summarized in [Table 1](#). The U-Net model outperformed the GLM on all measures except F_1 score. We then used the fitted U-Net model to predict the frequency of hail and compared it with the test data. [Figure 12](#) shows the results.

While there was a lag in predicting the number of events, the model captured the cyclical nature of these events and predicted the peaks and troughs, specifically where the event counts were close to zero or zero. Generally, we can conclude that the fitted model accurately predicted the number of events. The cumulative number of events predicted by the model would be close to the actual occurrence within a cycle. One notable downside of the fitted GLM was that it consistently predicted the presence of events even when none occurred. Additionally, it did not perform well

in mimicking the cyclical nature of the events. This was expected, especially given our chosen GLM.

7. CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

7.1. CONCLUSIONS

Our study shows that, in general, the U-Net model performs better than the traditional GLM, even with limited data. Notably, the GLM we fitted predicted day-by-day, while it is possible to predict a period block using U-Net density maps. Additionally, U-Net allows multichannel use, making it possible to input more than one stack of density maps into the model. This allows incorporating the dependencies, if any, within the various images. It also allows faster and more efficient training time with fewer computing resources.

7.2. LIMITATIONS

A significant limitation of this study was an incomplete dataset. Most raw data entries were incomplete, with many missing latitude and longitude data. Using estimates for the missing data would have adversely affected the model’s accuracy. Therefore, a substantial volume of data had to be excluded, leaving a smaller sample size, which adversely affected model training and testing. For example, hurricanes, which cause substantial losses, were missing longitude and/or latitude data—the most critical data elements for this modeling exercise. Complete hurricane data were only available beginning with 2016, so most hurricane data from earlier years had to be excluded. Second, ML algorithms often work best when the data points are continuous and tightly close to each other, which limited the techniques that could be evaluated. Additionally, the events were sparse and unbalanced; therefore, for some periods, the generated heat maps were just blanks, which affected the model’s predictive performance. Only three event types had complete data for the entire study period. However, with more granular data, such limitations can easily be overcome.

7.3. FUTURE WORK

With more data available, similar research can be done with a broader range of weather events. Additionally, mindful of the number of parameters to be estimated, a similar model with fewer parameters would allow less computer-intensive training and forecasting. It would also be interesting to compare U-Net with other CNN model extensions. For example, Liu, Racah, Correa, et al. (2016) promoted using deep CNN to accurately characterize extreme weather events and weather fronts. Hou et al. (2022) tested the ability of the CNN-LSTM (i.e., long short-term memory) model to predict hourly air temperature. Lee et al. (2020) asserted that ML techniques can be used to predict heavy rain damage. Doshi, Basu, and Pang (2018) recommended that CNN be applied to detect disaster-affected areas. It would be in-

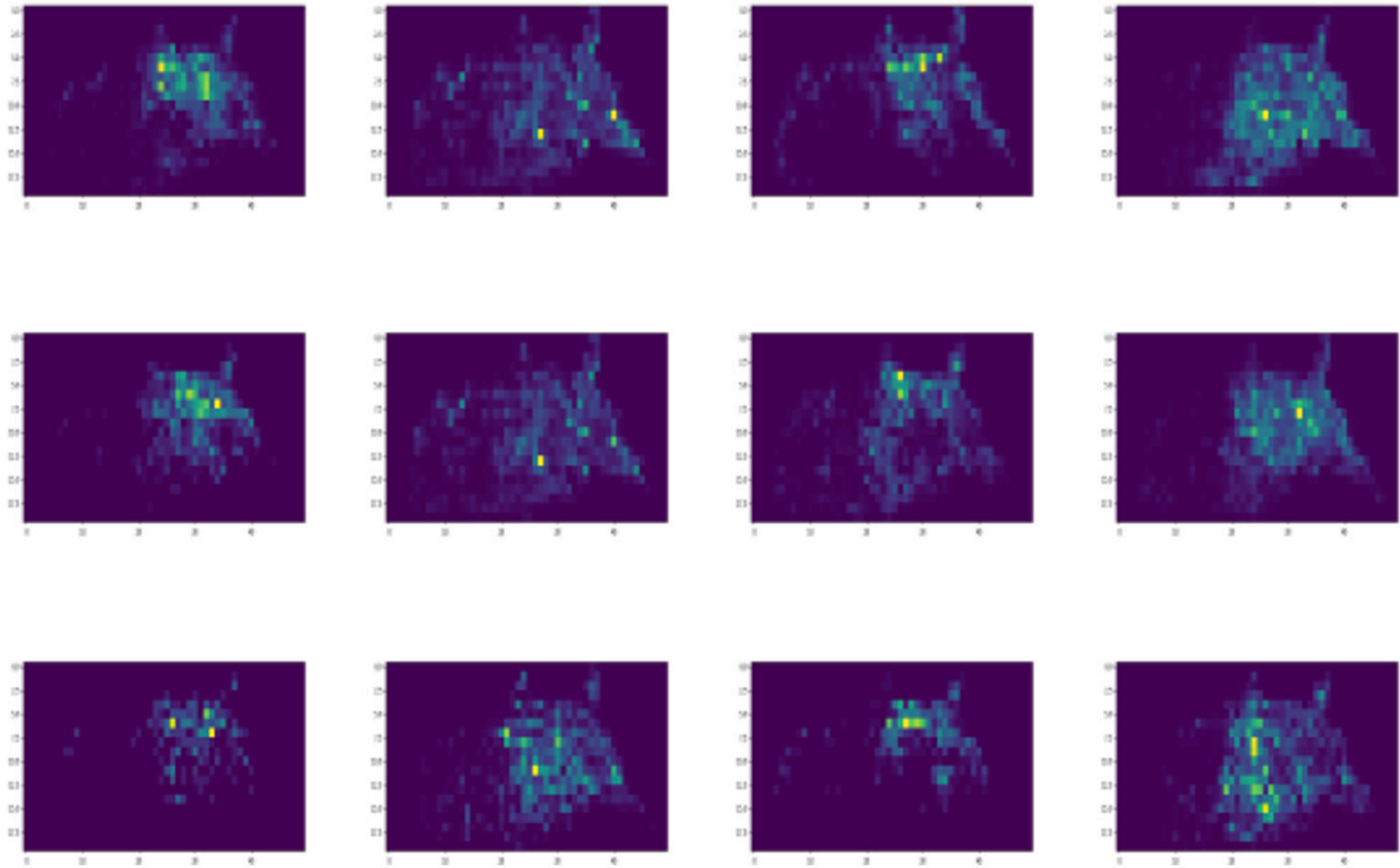


Figure 10. Actual versus predicted maps for the training dataset.

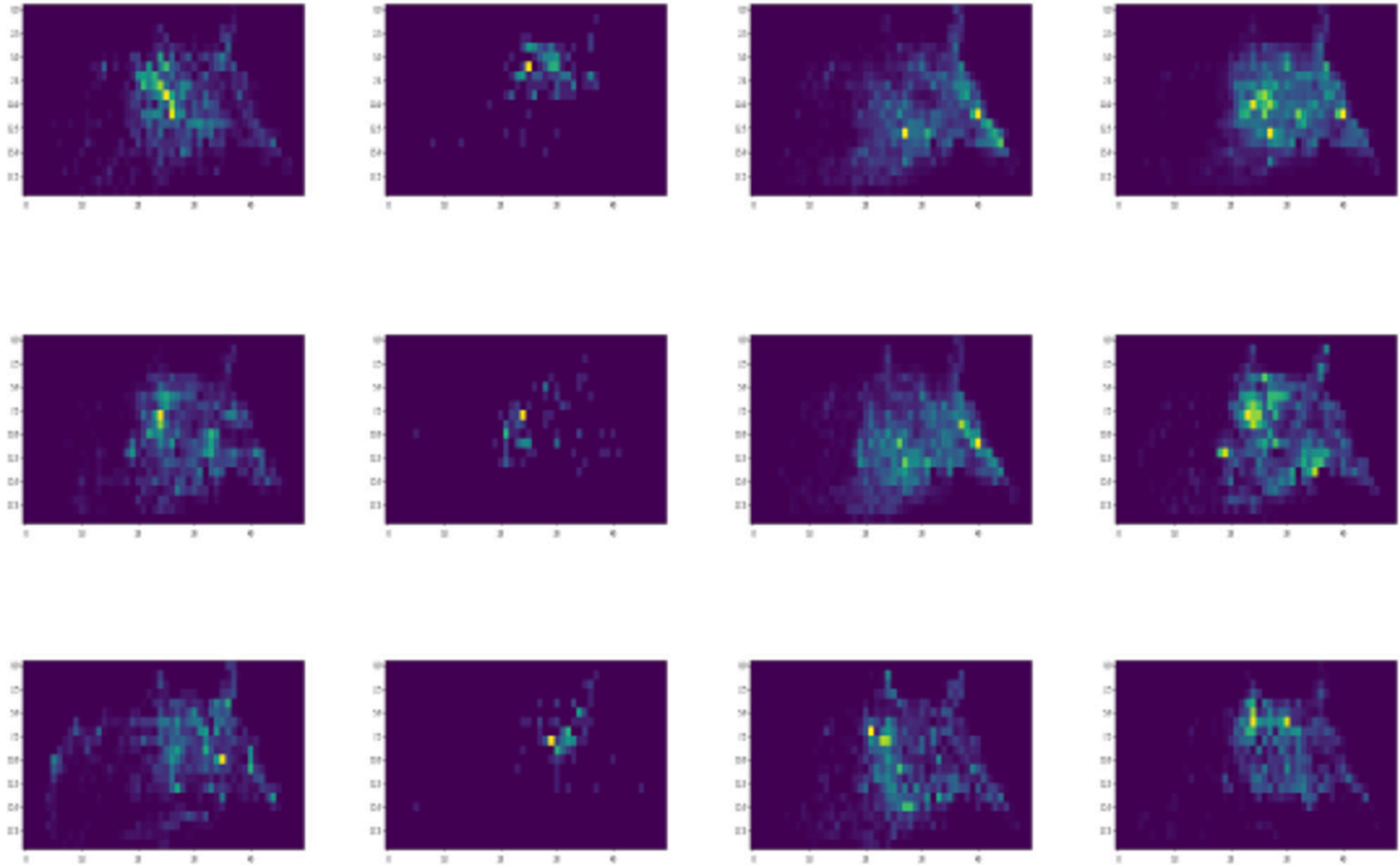
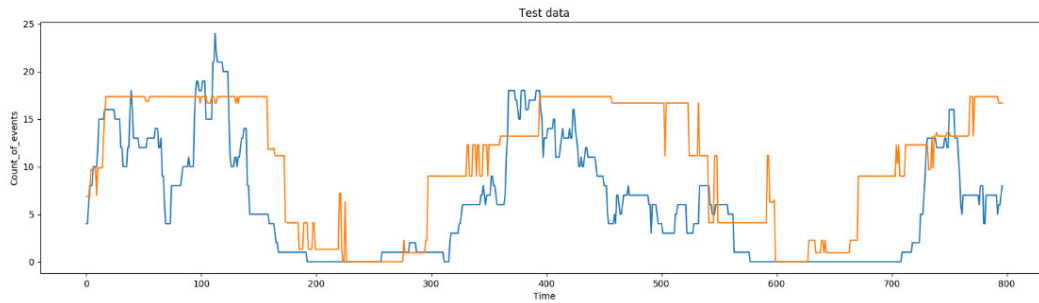


Figure 11. Actual versus predicted maps for the testing dataset.

Table 1. A summary of the fitted GLM and CNN (U-Net) model performance measures

Model	MAE	Precision	Accuracy	FAR	Recall	F ₁ Score
U-Net	1.7175	0.7010	0.7104	0.2989	0.8497	0.7604
GLM	1.9797	0.6578	0.6701	0.3401	0.6775	0.7832

**Figure 12. Actual versus predicted hail severity using testing dataset.**

interesting to compare the performance of these models with that of the U-Net architecture, especially with limited data.

Further, there is no one else who satisfies the criteria of this manuscript's authorship who has not been listed. Finally, we wish to confirm that all the named authors approve of the authorship order.

COMPETING INTERESTS

There were no known conflicts of interest at the time of submitting this paper. We also confirm that all the named authors have read and approved the submitted manuscript.

Submitted: January 09, 2023 EDT. Published: December 17, 2025 EDT.

REFERENCES

- Aleman, S., J. Beltran, A. Perez, and S. Ganzfried. 2019. "Predicting Hurricane Trajectories Using a Recurrent Neural Network." *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (01): 468–75. <https://doi.org/10.1609/aaai.v33i01.3301468>.
- Athira, V., P. Geetha, R. Vinayakumar, and K. P. Soman. 2018. "DeepAirNet: Applying Recurrent Networks for Air Quality Prediction." *Procedia Computer Science* 132:1394–1403. <https://doi.org/10.1016/j.procs.2018.05.068>.
- Barbounis, T., J. Theocharis, M. Alexiadis, and P. Dokopoulos. 2006. "Long-Term Wind Speed and Power Forecasting Using Local Recurrent Neural Network Models." *IEEE Transactions on Energy Conversion* 21 (1): 273–84. <https://doi.org/10.1109/tec.2005.847954>.
- Bhattacharyya, Siddhartha. 2011. "A Brief Survey of Color Image Preprocessing and Segmentation Techniques." *Journal of Pattern Recognition Research* 6 (1): 120–29. <https://doi.org/10.13176/11.191>.
- Born, Dorothea. 2019. "Bearing Witness? Polar Bears as Icons for Climate Change Communication in National Geographic." *Environmental Communication* 13 (5): 649–63. <https://doi.org/10.1080/17524032.2018.1435557>.
- Clayton, Susan. 2020. "Climate Anxiety: Psychological Responses to Climate Change." *Journal of Anxiety Disorders* 74 (102263): 1–7. <https://doi.org/10.1016/j.janxdis.2020.102263>.
- Dobson, Annette. 2002. *An Introduction to Generalized Linear Models*. Chapman and Hall. <https://doi.org/10.1201/9781420057683>.
- Donner, William, and Havidan Rodriguez. 2011. "Disaster Risk and Vulnerability: The Role and Impact of Population and Society." Population Reference Bureau. 2011. <https://www.prb.org/resources/disaster-risk/>.
- Doshi, Jigar, Saikat Basu, and Guan Pang. 2018. "From Satellite Imagery to Disaster Insights." Preprint, arXiv: Computers and Society. <https://doi.org/10.48550/ARXIV.1812.07033>.
- Du, Getao, Xu Cao, Jimin Liang, Xueli Chen, and Yonghua Zhan. 2020. "Medical Image Segmentation Based on U-Net: A Review." *Journal of Imaging Science and Technology* 64 (2): 1–12. <https://doi.org/10.2352/j.imagingsci.technol.2020.64.2.020508>.
- Dunn, Peter K., and Gordon K. Smyth. 2018. *Generalized Linear Models with Examples in R*. Springer. <https://doi.org/10.1007/978-1-4419-0118-7>.
- Elhoseiny, M., S. Huang, and A. Elgammal. 2015. "Weather Classification with Deep Convolutional Neural Networks." In *International Conference on Image Processing*, 2–5. <https://doi.org/10.1109/ICIP.2015.7351424>.
- England, Peter D. 2008. "The Actuary's Toolkit: A View from EMB." In *38th International ASTIN Colloquium*.
- Fawzy, Samer, John Doran, David W. Rooney, and Ahmed I. Osman. 2020. "Strategies for Mitigation of Climate Change: A Review." *Environmental Chemistry Letters* 18 (6): 2069–94. <https://doi.org/10.1007/s10311-020-01059-w>.
- Ghosh, S., A. Bergman, and B. Dodov. 2020. "How Machine Learning Is Taking Catastrophe Modeling to a New Level." Edited by J. Kinghorn. Verisk. December 16, 2020. <https://www.air-worldwide.com/publications/air-currents/2020/how-machine-learning-is-taking-catastrophe-modeling-to-a-new-level/>.
- Goldburd, Mark, Anand Khare, Dan Tevet, and Dmitriy Guller. 2025. "Generalized Linear Models for Insurance Rating, 2025 Revision." Casualty Actuarial Society, CAS Monograph Series 5. <https://www.casact.org/sites/default/files/2021-01/05-Goldburd-Khare-Tevet.pdf>.
- Guresen, Erkam, and Gulgun Kayakutlu. 2011. "Definition of Artificial Neural Networks with Comparison to Other Networks." *Procedia Computer Science* 3:426–33. <https://doi.org/10.1016/j.procs.2010.12.071>.
- Hardin, James W., and Joseph M. Hilbe. 2006. *Generalized Linear Models and Extensions*. Stata Press.
- Hernández, Benjamín, Suhas Somnath, Junqi Yin, et al. 2020. "Performance Evaluation of Python Based Data Analytics Frameworks in Summit: Early Experiences." *Communications in Computer and Information Science* 1315:366–80. https://doi.org/10.1007/978-3-030-63393-6_24.
- Höhlein, Kevin, Michael Kern, Timothy Hewson, and Rüdiger Westermann. 2020. "A Comparative Study of Convolutional Neural Network Models for Wind Field Downscaling." *Meteorological Applications* 27 (6): 1–31. <https://doi.org/10.1002/met.1961>.

- Hou, Jingwei, Yanjuan Wang, Ji Zhou, and Qiong Tian. 2022. "Prediction of Hourly Air Temperature Based on CNN-LSTM." *Geomatics, Natural Hazards and Risk* 13 (1): 1962–86. <https://doi.org/10.1080/19475705.2022.2102942>.
- Hu, C., Q. Wu, H. Li, S. Jian, N. Li, and Z. Lou. 2018. "Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation." *Water* 10 (11): 1543. <https://doi.org/10.3390/w10111543>.
- Ives, Anthony R. 2015. "For Testing the Significance of Regression Coefficients, Go Ahead and Log-Transform Count Data." *Methods in Ecology and Evolution* 6 (7): 828–35. <https://doi.org/10.1111/2041-210x.12386>.
- Jewson, S., C. Herweijer, and S. Khare. 2016. *Catastrophe Modeling for Climate Hazards: Challenges and Climate Change*. Risk Management Solutions.
- Jørgensen, Sisse Liv, Mette Termansen, and Unai Pascual. 2020. "Natural Insurance as Condition for Market Insurance: Climate Change Adaptation in Agriculture." *Ecological Economics* 169 (106489): 1–8. <https://doi.org/10.1016/j.ecolecon.2019.106489>.
- Karamouz, M., S. Razavi, and S. Araghinejad. 2007. "Long-Lead Seasonal Rainfall Forecasting Using Time-Delay Recurrent Neural Networks: A Case Study." *Hydrological Processes* 22 (2): 229–41. <https://doi.org/10.1002/hyp.6571>.
- Katz, R. 2002. "Stochastic Modeling of Hurricane Damage." *Journal of Applied Meteorology* 41 (7): 754–62. [https://doi.org/10.1175/1520-0450\(2002\)041.754.2](https://doi.org/10.1175/1520-0450(2002)041.754.2).
- Kaur, A., J. K. Sharma, and S. Agrawal. 2011. "Artificial Neural Networks in Forecasting Maximum and Minimum Relative Humidity." *IJCSNS International Journal of Computer Science and Network Security* 11 (5): 197–98. http://paper.ijcsns.org/07_book/201105/20110529.pdf.
- Kingma, Diederik P., and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." Preprint, arXiv: Machine Learning. <https://doi.org/10.48550/arXiv.1412.6980>.
- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature* 436:436–44. <https://doi.org/10.1038/nature14539>.
- LeCun, Yann, Bernhard Boser, John Denker, et al. 1989. "Handwritten Digit Recognition with a Back-Propagation Network." In *Advances in Neural Information Processing Systems*. <https://proceedings.neurips.cc/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf>.
- Lee, Kanghyeok, Changhyun Choi, Do Hyoung Shin, and Hung Soo Kim. 2020. "Prediction of Heavy Rain Damage Using Deep Learning." *Water* 12 (7): 1942. <https://doi.org/10.3390/w12071942>.
- Li, Y., R. Yang, C. Yang, M. Yu, F. Hu, and Y. Jiang. 2017. "Leveraging LSTM for Rapid Intensifications Prediction of Tropical Cyclones." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4/W2*:101–5. <https://doi.org/10.5194/isprs-annals-iv-4-w2-101-2017>.
- Liu, Yunjie, Evan Racah, Joaquin Correa, et al. 2016. "Application of Deep Convolutional Neural Networks for Detecting Extreme Weather in Climate Datasets." Preprint. arXiv: Computer Vision and Pattern Recognition. <https://doi.org/10.48550/arXiv.1605.01156>.
- Lowe, Julian, and Louise Pryor. 1996. "Neural Networks v. GLMs in Pricing General Insurance." *Institute and Faculty of Actuaries Workshop*. <https://www.actuaries.org.uk/system/files/documents/pdf/0417-0438.pdf>.
- Mak, S., D. Bingham, and Y. Lu. 2016. "A Regional Compound Poisson Process for Hurricane and Tropical Storm Damage." *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 65 (5): 677–703.
- Mandal, Manav. 2021. "Introduction to Convolutional Neural Networks (CNN)." Data Science Blogathon. 2021. <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>.
- McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. Springer. <https://doi.org/10.1201/9780203753736/generalized-linear-models-mccullagh>.
- McCulloch, Warren S., and Walter Pitts. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *The Bulletin of Mathematical Biophysics* 5 (4): 115–33. <https://doi.org/10.1007/bf02478259>.
- Mishra, Mayank. 2020. "Convolutional Neural Networks, Explained." *Towards Data Science*. <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939/>.
- Morganstein, Joshua C., and Robert J. Ursano. 2020. "Ecological Disasters and Mental Health: Causes, Consequences, and Interventions." *Frontiers in Psychiatry* 11 (1): 1–15. <https://doi.org/10.3389/fpsy.2020.00001>.

- NOAA (National Oceanic and Atmospheric Administration). 2020. "NOAA Online Weather Data (NOWData): Interactive Data Query System: Public Fact Sheet." <https://www.weather.gov/climateservices/nowdatafaq>.
- O'Hara, Robert B., and Johan D. Kotze. 2010. "Do Not Log-Transform Count Data." *Methods in Ecology and Evolution* 1 (2): 118–22. <https://doi.org/10.1111/j.2041-210x.2010.00021.x>.
- Oquab, Maxime, Leon Bottou, Ivan Laptev, and Josef Sivic. 2014. "Learning and Transferring Mid-Level Image Representation Using Convolutional Neural Network." In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2014.222>.
- Prein, Andreas F., and Greg J. Holland. 2018. "Global Estimates of Damaging Hail Hazard." *Weather and Climate Extremes* 22:10–23. <https://doi.org/10.1016/j.wace.2018.10.004>.
- Prudential Regulation Authority. 2015. "The Impact of Climate Change on the Uk Insurance Sector: A Climate Change Adaptation Report by The Prudential Regulatory Authority." <https://www.bankofengland.co.uk/-/media/boe/files/prudential-regulation/publication/impact-of-climate-change-on-the-uk-insurance-sector.pdf>.
- Riley, D., and G. Venayagamoorthy. 2011. "Comparison of a Recurrent Neural Network PV System Model with a Traditional Component-Based PV System Model." In *37th IEEE Photovoltaic Specialists Conference*, 1–4. <https://doi.org/10.1109/PVSC.2011.6186438>.
- Rolnick, Davi, Priya L. Donti, Lynn H. Kaack, et al. 2019. "Tackling Climate Change with Machine Learning." Preprint, arXiv. <https://arxiv.org/abs/1906.05433>.
- Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation." Preprint, arXiv: Computer Vision and Pattern Recognition. <https://doi.org/10.48550/arXiv.1505.04597>.
- Rotich, Titus. 2017. "An Actuary and the Climate Change Elephant." *The Actuary*, 2017. <https://www.theactuary.com/opinion/2017/03/2017/03/01/actuary-and-climate-change-elephant>.
- Sadeghi, Mojtaba, Phu Nguyen, Kuolin Hsu, and Soroosh Sorooshian. 2020. "Improving Near Real-Time Precipitation Estimation Using a U-Net Convolutional Neural Network and Geographical Information." *Environmental Modelling & Software* 134. <https://doi.org/10.1016/j.envsoft.2020.104856>.
- Saha, Sumit. 2018. "A Comprehensive Guide to Convolutional Neural Networks—the ELI5 Way." *Towards Data Science*. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- Salman, A. G., Y. Heryadi, E. Abdurahman, and W. Suparta. 2018. "Single Layer & Multi-Layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting." *Procedia Computer Science* 135:89–98. <https://doi.org/10.1016/j.procs.2018.08.153>.
- Sarkar, Dipanjan, Raghav Bali, and Tushar Sharma. 2018. *Practical Machine Learning With Python*. Apress. <https://doi.org/10.1007/978-1-4842-3207-1>.
- Scher, Sebastian, and Gabriele Messori. 2018. "Predicting Weather Forecast Uncertainty with Machine Learning." *Quarterly Journal of the Royal Meteorological Society* 144 (717): 2830–41. <https://doi.org/10.1002/qj.3410>.
- Seekings, C., and E. Anderson. 2017. "Hurricanes Could Result in \$145 Billion of Insured Losses." *The Actuary*. <https://www.theactuary.com/news/2016/09/2017/09/26/hurricanes-could-result-145bn-insured-losses>.
- Sousounis, Peter. 2021. "Quantifying Physical Risk from Climate Change: How Cat Models Can Help." *Airmic*. <https://www.airmic.com/news/guest-stories/quantifying-physical-risk-climate-change-how-cat-models-can-help>.
- Stankovic, Ljubisa, and Danilo Mandic. 2021. "Convolutional Neural Networks Demystified: A Matched Filtering Perspective Based Tutorial." arXiv: Information Theory. <https://doi.org/10.48550/arXiv.2108.11663>.
- Szegedy, Christian, Vincent Vanhoucke, Sergey Loffe, Jon Shlens, and Zbigniew Wojna. 2016. "Rethinking the Inception Architecture for Computer Vision." In *2016 IEEE Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2016.308>.
- Thistlethwaite, John, Andrea Minano, Jordan A. Blake, Daniel Henstra, and Daniel Scott. 2018. "Application of Re/Insurance Models to Estimate Increases in Flood Risk Due to Climate Change." *Geoenvironmental Disasters* 5 (8): 1–13. <https://doi.org/10.1186/s40677-018-0101-9>.
- Thistlethwaite, John, and Michael O. Wood. 2018. "Insurance and Climate Change Risk Management: Rescaling to Look Beyond the Horizon." *British Journal of Management* 29 (2): 279–98. <https://doi.org/10.1111/1467-8551.12302>.

- Trebing, Kevin, Tomasz Stańczyk, and Siamak Mehrkanoon. 2021. "SmaAt-UNet: Precipitation Nowcasting Using a Small Attention-UNet Architecture." *Pattern Recognition Letters* 145:178–86. <https://doi.org/10.1016/j.patrec.2021.01.036>.
- United Nations. 2020. "What Is Climate Change?" <https://www.un.org/en/climatechange/what-is-climate-change>.
- Van Rossum, Guido, and Fred L. Drake. 2009. *Python 3 Reference Manual*. CreateSpace. <https://doi.org/10.5555/1593511>.
- Warton, David I., Mitchell Lyons, Jakub Stoklosa, and Anthony R. Ives. 2016. "Three Points to Consider When Choosing a LM or GLM Test for Count Data." *Methods in Ecology and Evolution* 7 (8): 882–90. <https://doi.org/10.1111/2041-210x.12552>.
- Watts, Nick, Markus Amann, Nigel Arnell, et al. 2019. "The 2019 Report of The Lancet Countdown on Health and Climate Change: Ensuring That the Health of a Child Born Today Is Not Defined by a Changing Climate." *The Lancet* 394 (10211): 1836–78. [https://doi.org/10.1016/s0140-6736\(19\)32596-6](https://doi.org/10.1016/s0140-6736(19)32596-6).
- Werbos, Paul J. 1974. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University.
- Whitmore-Williams, Susan Clayton, Christie Manning, Kirra Krygsman, and Meighen Speiser. 2017. "Mental Health and Our Changing Climate: Impacts, Implications, and Guidance." *American Psychological Association*. <https://www.apa.org/news/press/releases/2017/03/mental-health-climate.pdf>.
- Wood, Simon N. 2006. *Generalized Additive Models*. Chapman and Hall. <https://doi.org/10.1201/9781420010404>.
- World Meteorological Organisation. 2021. "Weather-Related Disasters Increase over Past 50 Years, Causing More Damage but Fewer Deaths." <https://wmo.int/media/news/weather-related-disasters-increase-over-past-50-years-causing-more-damage-fewer-deaths>.
- Yao, Wei, Zhigang Zeng, Cheng Lian, and Huiming Tang. 2018. "Pixel-Wise Regression Using U-Net and Its Application on Pansharpening." *Neurocomputing* 312:364–71. <https://doi.org/10.1016/j.neucom.2018.05.103>.
- Zagoruyko, Serge, and Nikos Komodakis. 2015. "Learning to Compare Image Patches via Convolutional Neural Network." In *2015 IEEE Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2015.7299064>.
- Zaytar, Mohamed Akram, and Chaker El Amrani. 2016. "Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks." *International Journal of Computer Applications* 143 (11): 7–11. <https://doi.org/10.5120/ijca2016910497>.
- Zhang, P., Y. Jia, J. Gao, W. Song, and H. Leung. 2018. "Short-Term Rainfall Forecasting Using Multi-Layer Perceptron." *IEEE Transactions on Big Data* 6 (1): 93–106. <https://doi.org/10.1109/tbdata.2018.2871151>.
- Zhang, Zhongheng, Marcus W. Beck, David A. Winkler, Bin Huang, Wilbert Sibanda, and Hemant Goyal. 2018. "Opening the Black Box of Neural Networks: Methods for Interpreting Neural Network Models in Clinical Applications." *Annals of Translational Medicine* 6 (11). <https://doi.org/10.21037/atm.2018.05.32>.
- Zhou, X., J. Xu, P. Zeng, and X. Meng. 2019. "Air Pollutant Concentration Prediction Based on GRU Method." *Journal of Physics Conference Series* 1168:032058. <https://doi.org/10.1088/1742-6596/1168/3/032058>.
- Zupan, Jure. 1994. "Introduction to Artificial Neural Network (ANN) Methods: What They Are and How to Use Them." *Acta Chimica Slovenica* 41 (3): 327.

APPENDICES

Appendix A

Table A1. Similarities and differences between the widely used GLMs and ANN models.

ANN Models	GLMs
Lower parameter risk because fewer parameters are estimated. The parameters are automatically recalibrated by the model.	Increased parameter risk because the estimated parameters are required before fitting the model.
Reduced statistical risk because no distributional and relational assumptions about the data and the model are required.	Increased statistical risk because assumptions about data distribution, error, and the relationships between dependent and independent variables are required before fitting the model.
Can be used for both linear and nonlinear analyses.	Best output for linear analyses.
Model can be trained to solve a new problem.	Models would need to be fitted to any other set of data.
Discontinuous variables can be modeled.	Assumes that the functional form of the data is continuous everywhere.
Outliers do not affect it.	Outliers adversely affect it.
Based on minimizing output error.	Based on minimizing output error.
Might not be easy to understand the model.	It is easier to understand the model and explain the output.

Appendix B

Table B1. Previous research that applied ML techniques to model weather-related events.

Reference	Inputs	Dependent variable	Model	Performance
Kaur, Sharma, and Agrawal (2011)	Indian meteorological department data	Humidity	MLP	RMSE: 4.258
P. Zhang et al. (2018)	Variables (13): height, temperature, temperature dew point difference, wind direction, wind speed, cloud amount, surface wind speed, surface wind direction, air pressure, pressure change, surface temperature dew point difference, surface temperature, rainfall over last 3 hours	Rainfall	MLP	Threat score: 0.87 RMSE: 2.73
			ARIMA	Threat score: 0.70 RMSE: 31.71
			SVM	Threat score: 0.72 RMSE: 20.16
Barbounis et al. (2006)	Wind direction and speed provided SKIRON for 4 positions up to 30 km farther from the turbine cluster	Wind speed (WS) and power (WP)	MLP	WS RMSE: 2.9762 WP RMSE: 1.9029
			RNN	WS RMSE: 2.9251 WP RMSE: 1.8471
Riley and Venayagamoorthy (2011)	Weather conditions, including irradiance, spectrum, and temperature	Photovoltaic system output	RNN	RMSE: 3.198 %
			SAPM	RSME: 3.050 %
Karamouz, Razavi, and Araghinejad (2007)	Precipitation in the southwest (SW), northwest (NW) and west (W) basins of Iran	Rainfall	RNN	SW RMSE: 139.66 NW RMSE: 64.67 W RMSE: 110.27
			ARMAX	SW RMSE: 103.41 NW RMSE: 73.67 W RMSE: 137.90
Aleman et al. (2019)	Longitude, latitude, pressure, and wind speed publicly provided by the National Hurricane Center	Trajectory of hurricanes (DEAN, SANDY and ISAAC)	RNN	DEAN RMSE: 0.0842 SANDY RMSE: 0.0800 ISAAC RMSE: 0.0592
Zaytar and El Amrani (2016)	Wind speed, humidity and temperature in 9 Moroccan cities	24 hours and the other one 72 hours of weather data	LSTM	24 hours average MSE: 0.00679
				72 hours average MSE: 0.00903
Li et al. (2017)	SHIPS database (contains most relevant predictors relevant to TC intensity fluctuations, such as sea surface temperature and sea level pressure)	Occurrence of tropical cyclones	LSTM	Accuracy: 68%
Hu et al. (2018)	Data were collected from 14 rainfall stations in Fen River in addition to one hydrologic station	Rainfall-runoff	MLP	RMSE: 25.6 R2: 0.75
			LSTM	RMSE: 14.4

Reference	Inputs	Dependent variable	Model	Performance
				R2: 0.83
Salman et al. (2018)	Time series data from Nadim Airport in Indonesia (pressure, temperature, dew point, and humidity)	Pressure (P), temperature (T), dew point (DP), and humidity (H)	LSTM	P RMSE: 0.0775 P accuracy: 0.8060 T RMSE: 0.0822 T accuracy: 0.7453 DP RMSE: 0.0778 DP accuracy: 0.8053 H RMSE: 0.0771 H accuracy: 0.8029
Athira et al. (2018)	Meteorological and pollution data from AirNet	Air quality	LSTM	MSE: 0.00058
			RNN	MSE: 0.00025
			GRU	MSE: 0.00023
Zhou et al. (2019),	Beijing's hourly particulate matter 2.5 weather and concentration information	Air pollutants concentration	GRU	RMSE: 0.0752
Elhoseiny, Huang, and Elgammal (2015)	50,000 weather relative images	Weather classification	CNN	Accuracy :82.2%
Liu, Racah, Correa, et al. (2016)	Image patterns relative to extreme weather events that occurred in the past	Extreme weather events	CNN	Accuracy: 89%–99%
Katz (2002)	Dataset of damage estimates for cyclones that occurred in North America	Total damage relative to hurricanes	CPP	MSE: 5.019 (bad performance)
Mak, Bingham, and Lu (2016)	Storm paths data and regional effects data	Tropical storm and hurricane prediction	CPP	No evidence of trend

APPENDIX C

This appendix provides an overview of the basic concepts of the CNN model to help readers better comprehend the fundamental data transmission and its components.

NEURAL NETWORK

The nervous system is a network made up of many neurons. Similarly, neurons are also the rudimentary processing units of artificial neural networks. The principle of operation is that several input values undergo mathematical transformation to obtain an output value. The mathematical transformation relationship between the input signal and output value can be expressed as

$$f(b + n\sigma_i = 1(x_i \times w_i))$$

Where f is the activation function. There are multiple activation functions, such as ReLU, sigmoid, and Tanh, among others (see Szegedy et al. 2016 for more details).

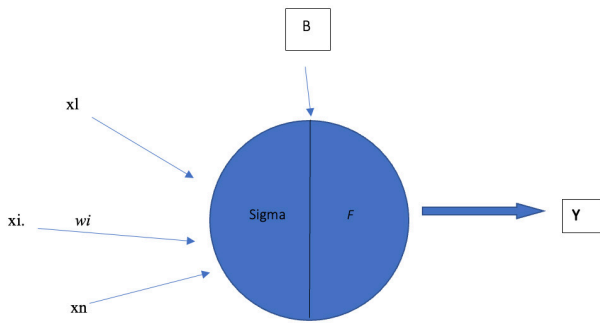


Figure C1. Neuron model, x_i is the input signal, n is the number of signals, the weight value of the input signal is w_i , bias is B , and the output neuron is Y (adapted from Szegedy et al. 2016).

CNN ARCHITECTURE

The integral CNN structure includes a convolutional layer, pooling layer, nonlinear activation layer, and fully connected layer. In general, the image is preprocessed and then fed into the network through the input layer. It is then processed by multiple oppositely arranged convolutional layers and pooling layers and then stratified by the fully connected layer (see Bhattacharyya 2011).

CNN adds a very attributive convolutional layer and pooling layer. With more pixels and larger datasets, CNN will have outstanding cost performance in model size and better performance. The convolutional layer has the attributes of a local receptive field, which keeps the input shape so that the correlation between the properties of the image pixels in the length and width orientation can be effectively identified (see Oquab et al. 2014).

CONVOLUTIONAL LAYER

For CNNs with a certain depth, the convolution operation of multiple convolution layers can obtain different input properties. The bottom layer convolution extracts feature such as texture, lines, and edges, while the higher layer extracts more abstract features (see Zagoruyko and Komodakis 2015). The convolutional layer contains several convolutional kernels with learnable parameters. It is a matrix made up of learnable weights, which are, in general, 3×3 , 5×5 , and 7×7 weight matrices with equal odd number length and width. The weight matrix of the convolution kernel corresponds to the local area of the convolution feature map, and the convolution kernel sequentially performs convolution operation on the area on the feature map by sliding (see LeCun, Bengio, and Hinton 2015 for more details).