atmosphere

MDPI

Article

Lightning Nowcasting Using Solely Lightning Data

Ehsan Mansouri [1], Amirhosein Mostajabi [1], Chong Tong [2], Marcos Rubinstein [3] and Farhad Rachidi [1,*]

[1] Electromagnetic Compatibility Laboratory, Swiss Federal Institute of Technology (EPFL), 1015 Lausanne, Switzerland; ehsan.mansouri@epfl.ch (E.M.); amirhossein.mostajabi@epfl.ch (A.M.)
[2] State Grid JiangSu Electric Power Co., Ltd. Suzhou Branch, Suzhou 215000, China; chong.tong@smartpdg.com
[3] Institute for Information and Communication Technologies, University of Applied Sciences of Western Switzerland (HES-SO), 1400 Yverdon-les-Bains, Switzerland; marcos.rubinstein@heig-vd.ch
* Correspondence: farhad.rachidi@epfl.ch

Abstract: Lightning is directly or indirectly responsible for significant human casualties and property damage worldwide. A timely prediction of its occurrence can enable authorities and the public to take necessary precautionary actions resulting in diminishing the potential hazards caused by lightning. In this paper, based on the assumption that atmospheric phenomena behave in a continuous manner, we present a model based on residual U-nets where the network architecture leverages this inductive bias by combining information passing directly from the input to the output with the necessary required changes to the former, predicted by a neural network. Our model is trained solely on lightning data from geostationary weather satellites and can be used to predict the occurrence of future lightning. Our model has the advantage of not relying on numerical weather models, which are inherently slow due to their sequential nature, enabling it to be used for near-future prediction (nowcasting). Moreover, our model has similar performance compared to other machine learning based lightning predictors in the literature while using significantly less amount of data for training, limited to lightning data. Our model, which is trained for four different lead times of 15, 30, 45, and 60 min, outperforms the traditional persistence baseline by 4%, 12%, and 22% for lead times of 30, 45, and 60 min, respectively, and has comparable accuracy for 15 min lead time.

Keywords: lightning; nowcasting; machine learning; data-driven; satellite observations; lightning early warning; U-Net; ResUNet

Citation: Mansouri, E.; Mostajabi, A.; Tong, C.; Rubinstein, M.; Rachidi, F. Lightning Nowcasting Using Solely Lightning Data. *Atmosphere* **2023**, *14*, 1713. https://doi.org/10.3390/atmos14121713

Academic Editors: Kleber Pinheiro Naccarato and Ana Paula Paes Dos Santos

Received: 29 September 2023
Revised: 13 November 2023
Accepted: 15 November 2023
Published: 21 November 2023

1. Introduction

Lightning is an atmospheric phenomenon that transfers significant amounts of electrical charge over a short period of time, releasing energies of several gigajoules, and resulting in electromagnetic radiation over a very large frequency bandwidth. Lightning can cause human and livestock casualties, infrastructure damage, electromagnetic interference, forest fires, system failure in aviation systems, and power network disruption [1–8]. Therefore, a timely and accurate prediction of lightning occurrence combined with appropriate preventive actions can reduce and avert these unfortunate outcomes. Traditional prediction models attempt to characterize charge separation and electrification in the clouds and to predict lightning formation using numerical models [9–14]. Other traditional approaches rely on cloud-resolving models, i.e., convective cloud numerical simulations [15–21]. In [22], the weather research and forecasting (WRF) model is used to derive electrical potential energy, which is related to the probability of lightning occurrence. In [23,24], the product of precipitation and convective available potential energy, CAPE, is used as an input to a cloud-to-ground lightning prediction. These models suffer from the following two drawbacks: (1) Numerical weather models require a high amount of computational resources; this limits the computation speed, especially for nowcasting applications [25,26]. Moreover, (2) the available data are only used for calibration instead of modeling.

In order to utilize the historical recorded data, [27] presented a neural network-based approach to predict lightning using a combination of ground-based and satellite data and weather forecasting models. In [28] the authors combined numerical weather models (WRF) with an encoder-decoder model based on convolutional Long short-term memory (ConvLSTM [29]). In another model from the previous group, the Channel-wise attention mechanism is used in order to select between the WRF simulation features and observations [30]. In the encoder, observed data are passed through a convolutional feature extractor and ConvLSTM. The output of the encoder is fed to the decoder. The decoder also gets WRF simulations and passes them through an attention layer that selects the appropriate channels needed for the prediction. Although these models solve the issue of the efficient usage of data, they still suffer from relying on numerical weather models, which are computationally expensive for nowcasting purposes. In [31], a combination of satellite data recorded over Europe and lightning data recorded by ground-based lightning detection networks are used to train a residual U-Net model for lightning activity prediction [32–35]. In [36], the GLM data in combination with aerosol features are used to train a Gradient-boosted decision tree model called LightGBM to perform hourly forecasts.

During the past decade, important progress has been made in the field of weather prediction using machine learning which can be either directly or indirectly transferred to lightning prediction as lightning and precipitation are shown to be tightly related [37]. Ref. [38] developed a U-net based model for precipitation nowcasting for 1 to 3 h in the future. The multi-radar multi-sensor (MRMS) system for precipitation measurement provided the training and testing dataset for their model. The model was shown to outperform the numerical model used at that time. Refs. [39,40] presented a new model called MetNet for 8 h prediction which outperforms the model used by the National Oceanic and Atmospheric Administration (NOAA). In these models, in addition to MRMS, GEOS-16 channels, latitude, longitude, and elevation are also provided to the model. The model uses a deep convolutional neural network for feature extraction in the spatial domain followed by a ConvLSTM for temporal encoding. In the end, two axial attention mechanisms are used for the final precipitation prediction.

In this work, we propose to utilize a residual U-Net structure for lightning nowcasting and treat it as a video processing problem. We are using as input lightning data observed by the GOES-16's Geostationary Lightning Mapper (GLM) sensor, without utilizing numerical weather models.

Although there is no simple way to compare the physics-based, numerical models to our ML-based models directly due mainly to the appreciable difference in the lead times, the following observations can be made: The prediction of lightning using physics-based numerical models is not necessarily more accurate since convective storms generating lightning develop rapidly and occur in limited areas. By solving physical equations, numerical weather forecast tools provide essential planetary-scale predictions several days in advance. However, to ensure that the processing time does not exceed the prediction lead time, these computationally intensive tools can only make predictions for lead times longer than about two hours. Our approach, which is targeted to nowcasting, fills the gap in this crucial time interval.

2. Materials and Methods

Our method performs lightning activity prediction by considering only recent lightning activity through an autoregressive model. In this section, first, we formally introduce the problem and then we describe the data used in this study. After that, we present the baselines and proposed models. Finally, we explain the training process.

2.1. Problem Formulation

The desired target for a large area lightning predictor is a 2-D map of the area under study that expresses the probability of lightning occurrence in each location for the next hour. In order to reach this, the system needs to be fed with appropriate data based on

which it could make predictions. In this work, an autoregressive model has been utilized to capture the inherent patterns of lightning activity and build a baseline for future lightning prediction research. It has been shown that the model, despite only using lightning activity data, can achieve reasonable performance for lead times greater than 15 min.

Suppose that the two-dimensional map X_n of size $H \times W$ describes the lightning activity at time step n, where each pixel can take either the value 1 or the value 0 depending on whether or not any lightning activity occurred between time steps $n-1$ and n. The goal is to find the same map that describes the lightning activity at time step $n+1$. More formally, we want to find out a mapping $\mathcal{F} \in \mathbb{R}^{n \times H \times W} \rightarrow \mathbb{R}^{H \times W}$ such that:

$$X_{n+1} = \mathcal{F}(X_1, \ldots, X_{n-1}, X_n) \tag{1}$$

Finding the exact function is challenging due to the chaotic behavior of atmospheric phenomena, lack of contributing atmospheric factors, high dimensional data, discrete-time sampling, computational limits, and the complex nature of lightning. Therefore, an approximate function needs to be found, the output of which is a probability map describing the presence of lightning in each pixel for the next time steps as close as possible to actual observations. To that end, the following minimization problem needs to be solved:

$$\underset{\hat{\mathcal{F}} \in \mathbb{R}^{n \times H \times W} \rightarrow \mathbb{R}^{H \times W}}{minimize} \mathcal{L}(Y_{n+1}, X_{n+1}) \tag{2}$$

$$Y_{n+1} = \hat{\mathcal{F}}(X_1, \ldots, X_{n-1}, X_n) \tag{3}$$

where $\mathcal{L}$ is a loss function that measures the distance between the prediction Y_{n+1} and the target X_{n+1}. As it will be explained later in detail, the data are divided into time intervals of 15 min, which require 4 outputs in order to fulfill the one-hour lead-time prediction. Hence, the final optimization problem can be described as follows:

$$\underset{\hat{\mathcal{F}} \in \mathbb{R}^{n \times H \times W} \rightarrow \mathbb{R}^{4 \times H \times W}}{minimize} \mathcal{L}(Y_{n+1}, X_{n+1}, \ldots, Y_{n+4}, X_{n+4}) \tag{4}$$

$$[Y_{n+1}, Y_{n+2}, Y_{n+3}, Y_{n+4}] = \hat{\mathcal{F}}(X_1, \ldots, X_{n-1}, X_n) \tag{5}$$

2.2. Data

The input and output of the training/prediction system are lightning occurrence maps over the Americas which are collected by the Geostationary Lightning Mapper (GLM) installed on the GOES-16 satellite in a geostationary orbit. The Geostationary Lightning Mapper is a single-channel, near-infrared optical transient detector that can detect the momentary changes in an optical scene, indicating the presence of lightning. GLM measures total lightning activity (in-cloud, cloud-to-cloud and cloud-to-ground) continuously over the Americas and adjacent ocean regions with a nearly-uniform spatial resolution of approximately 10 km. The raw data are records of energy, area, latitude, and longitude of lightning events during 2019. These data are converted to a map of lightning flashes in the region shown in Figure 1. The figure also shows a snapshot of the lightning flashes at any arbitrary time in 2019. The data have been discretized in time by time steps 15 min apart and in space by squares of around 10 km $\times$ 10 km, hence forming a series of two-dimensional matrices of dimension 1086 $\times$ 1086 corresponding to spatial locations where each pixel can take either the value 1 or the value 0 depending on whether or not any lightning activity occurred in the past 15 min.

In order to feed the data to the system, some adjustments need to be made to the raw data. First of all, we limited our analysis to a region in the northern part of South America (covering Brazil) characterized by a relatively high lightning activity compared to the rest of the continent [41]. Hence, instead of matrices of size 1086 by 1086, the considered region is described by matrices of size 500 by 500, highlighted with red boundaries in Figure 1. Also, due to the computational limits, the data were cropped to avoid out-of-memory problems. Cropping is also needed to increase the training speed. The final cropped area is

a random region inside the mentioned highlighted area and is different from one sample to another in order to reduce the statistical dependency of the model on the region. Therefore, the matrices were further cropped to a size of 256 by 256.

Figure 1. GOES-16 field of view, region of interest, and sample lightning occurrences seen by GLM installed on GEOS-16, red dots represent the lightning events in an arbitrary time interval of 15 min.

In the next step, two matrix sequences corresponding to the input and the target are generated. As the input, a sequence of 8 matrices which correspond to 2 h is selected [42]. Based on this sequence, the predictors need to generate lightning probability maps for the four upcoming time steps, namely one hour lead time. This lead time has been selected as it has been used in previous works (e.g., [43,44]). An overview of the data preparation is illustrated in Figure 2 and a sample of input/target sequence pairs of the lightning maps is shown in Figure 3.

A short discussion on the limitations associated with the used data is in order. First of all, the spatial resolution is limited to 10 km. Moreover, due to the extremely long time required to download the data used in the present study (of the order of months), as well as the time required to convert records of lightning data into images, we were obliged to limit the data to single year. Converting lightning records into images that are processable by the model requires several considerations. Each lightning record in the dataset consists of several parameters, including latitude, longitude, amplitude, and area. In order to convert all the records into an image, we need to snap each record into a two-dimensional grid and assign its value to the corresponding pixel. This requires a huge amount of processing as, for each image, we need to map lightning records that occurred in the past 15 min to the pixels of the image. Another limitation is the 15-min granularity of the images. Although the GLM sensor provides lightning data every 20 s, the ABI (Advanced Baseline Imager) sensor, which provides images at 16 different wavelengths, is limited to a granularity of 15 min. Since we would like to integrate these two data types together in our future work, we have adopted the same time resolution to make our future models comparable to the present study. Finally, we are using zero-one lightning maps despite the fact that several other attributes of the lightning are also reported by the GLM

instrument, such as the flash energy and the flash area (which is the area on the Earth's surface where lightning flashes are detected and recorded by GLM). This is again due to the computation limitation of converting several lightning flashes into images. We have tried to make images considering each lightning flash as a circle with the area mentioned in the record entry and painting the images with a circle instead of a single pixel based on the ratio of energy to area. However, this is a computationally expensive process that will be considered in our future work.

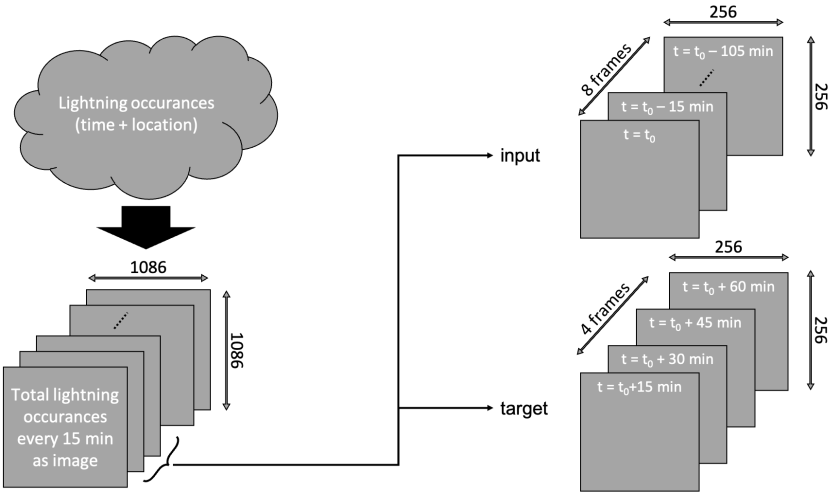

Figure 2. An overview of the data preparation procedure.

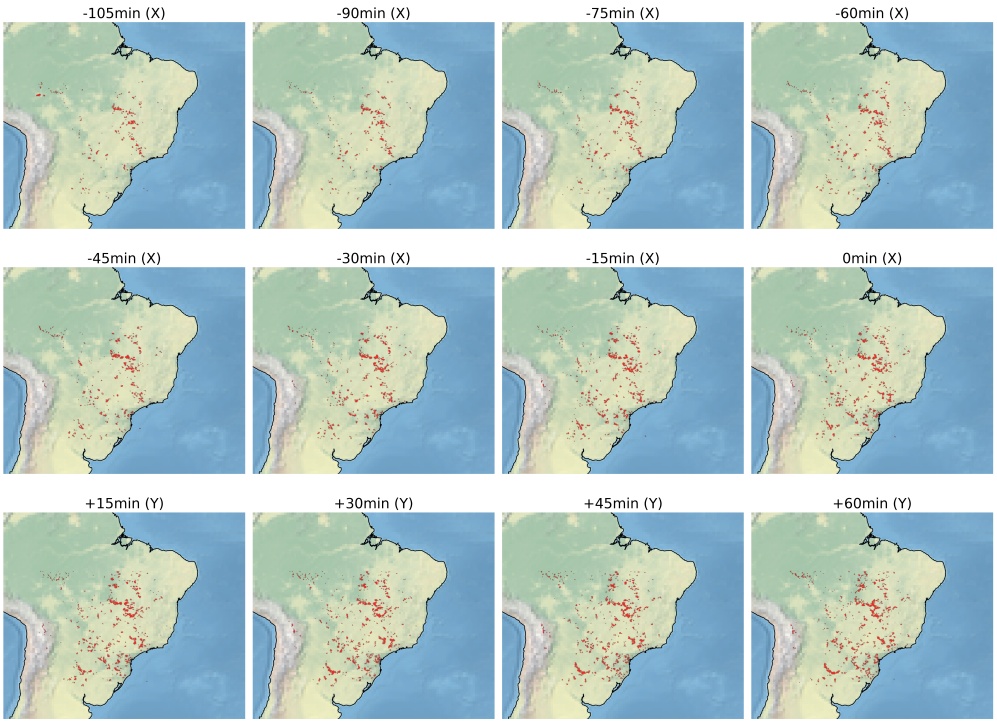

Figure 3. Sample data. The first and second rows correspond to input and the third row corresponds to the target output. Red dots represent the lightning occurrences between time t and t-15 min for each timestep t.

2.3. Baselines

In order to evaluate the performance of our model and assess its relative effectiveness, a simple but frequently used baseline called Eulerian persistence is used. The Eulerian persistence, from now onward referred to as persistence, states that the best prediction for

the next time frame is a replica of the current situation. Suppose that $X_1, X_2, \ldots, X_n$ are available. The persistence model predicts lightning activity Y as follows:

$$Y_{n+1}^{pred} = sgn(X_n) \tag{6}$$

where $sgn()$ is the sign function, equal to one for positive values and equal to zero for zero input. Since X_n is the last available data point (lightning map), it is used in generating the persistence model output. For our four-output system, we have:

$$Y_{n+1}^{pred}, Y_{n+2}^{pred}, Y_{n+3}^{pred}, Y_{n+4}^{pred} = sgn(X_n) \tag{7}$$

A more common baseline for atmospheric phenomena predictions is Lagrangian persistence. In Lagrangian persistence, the prediction for the next state is the current observed state shifted by the amount of predicted movement [27]. This method uses dense maps like clouds or precipitation to calculate the movement. However, in Leinonen et al., 2022, the performance gain from using Lagrangian persistence was negligible compared to simple Eulerian persistence. Moreover, The calculation of storm movement is not computationally stable if one only uses sparse lightning maps, as is the case in our study. Hence, in this paper, we do not use Lagrangian persistence. The Vanilla autoencoder is the second and more sophisticated baseline used in this work [45–48]. Autoencoders consist of (i) an encoder that reduces the dimension of the input gradually through its layers and produces a compressed representation, and (ii) a decoder that uses this compressed representation and reconstructs the output by gradually expanding it. In the first part, a combination of convolution and pooling reduces the dimensions of the input while extracting features. In the second part, transposed convolution is used along with convolution for up-sampling and recovering the desired output.

In this work, three different models with various numbers of layers have been trained. One is a baseline (Autoencoder) and the other two are our proposed models (U-Net and ResUnet). All convolutional layers have a kernel size of 3×3 except the last layer, which generates prediction by combining extracted features from the previous layer using a 1×1 convolutional layer. Moreover, Maxpooling 2×2 is used for spatial dimension reduction [49]. A detailed description of the models is presented in Appendix A.

2.4. Proposed Models

Our first attempt towards finding a reasonable lightning prediction model is U-Net [50]. Compared to autoencoder's architecture, U-Net provides a highway path, a direct connection without network processing from the input to the output of the corresponding layers, for the information and forces the model to use the input and learn necessary modifications in order to generate the output. Since no abruptness is present in the nature of atmospheric phenomena, an adjustment to the previous state would be sufficient to predict the future state as opposed to fully regenerating it from scratch, hence the U-Net has an apparent advantage over autoencoder thanks to the highway path.

The second model used in this work is a residual U-Net. Residual U-Net has the same general structure as U-Net consisting of a contraction path followed by an expansion path as well as direct copying of each level's input to its output. However, each level of the encoder and decoder is itself a residual sub-network. In other words, the network is made of blocks with a highway path from their inputs to their outputs in parallel to a sub-network of alternating convolutional layer, batch normalization, and Relu activation layer. These blocks are connected together by upsampling and downsampling layers and copy-through connections [32–35]. Figure 4 shows the base autoencoder architecture as well as the modifications that lead to U-net, and then ResUnet.

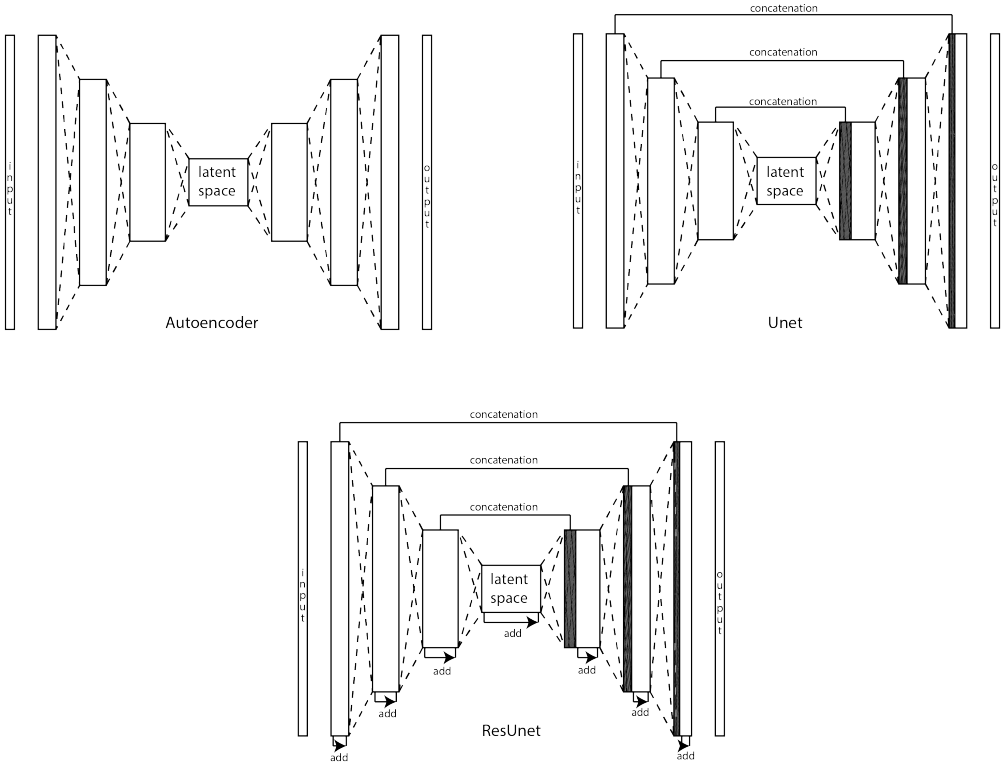

Figure 4. Schematic representation of the autoencoder, Unet and ResUnet architecture.

In order to compare the results, the same number of layers with the same filter sizes as autoencoder is used. By doing this, the number of parameters is kept approximately identical across corresponding networks. Therefore, a fair comparison can be made.

2.5. Loss, Metrics, and Training Process

In the following, the details of the training process including optimization goal (i.e., losses), evaluation parameters (i.e., metrics), hyperparameters, and codes are explained.

The models are trained with the aim of minimizing the total loss, which is a weighted sum of the Dice loss and cross-entropy loss. The cross-entropy loss is the common loss used in classification problems and is defined as:

$$\mathcal{L}_{ce} = \frac{1}{N}\Sigma_{n=1}^{N} X_n \log Y_n \tag{8}$$

where it captures the pixel-wise inconsistency of the prediction and the ground truth, and by minimizing it, the leaning process can be done more efficiently. The Dice loss (continuous Dice loss) is defined as in [51]:

$$\mathcal{L}_{dice} = 1 - \frac{2 \times |X_n \cap Y_n|}{c|X_n| + |Y_n|} \tag{9}$$

where $|\bullet|$ is the summation over all the elements of the operand, $\cap$ is the intersection of the two sets, which in this case is element-wise product of the elements, and finally:

$$c = \frac{\Sigma X_n^i Y_n^i}{\Sigma X_n^i sign(Y_n^i)} \tag{10}$$

where the index i identifies each pixel in X_n and Y_n. Dice loss is a region-based loss and can capture the cumulative discrepancy between prediction and ground truth. Therefore, by minimizing it, the model improves its prediction capability by optimizing the overlap and alignment of the segmented regions. The total loss is:

$$\mathcal{L} = \beta \mathcal{L}_{ce} + \mathcal{L}_{dice} \tag{11}$$

where a value of $\beta = 5$ has been selected through hyper-parameter tuning. In order to track the performance of the model during training, precision, recall, and f1-score are tracked. Since the model predicts four lead times, these values are measured for each lead time separately. The definition of these values are given as follows:

$$precision = \frac{TP}{TP + FP} \tag{12}$$

$$recall = \frac{TP}{TP + FN} \tag{13}$$

$$f1 - score = \frac{2 * precision \times recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \tag{14}$$

where TP (true positive) is the number of correctly predicted lightnings, FP (false positive) is the number of false alarms i.e., pixels falsely detected to contain lightning, and FN (false negative) is the number of missed lightning detections. Precision, recall, and f1-score are all between 0 and 1, with higher values indicating better performance. Precision is an indicator of how often the model incorrectly predicts a lightning event, meaning that the predicted lightning did not happen. On the other hand, recall is an indicator of how often the model misses predicting a lightning event, meaning the lightning discharge occurred but the model failed to predict it. In general, there is always a trade-off between a high precision score and a high recall score. In an extreme case, a model that predicts the occurrence of lightning for every time slot and position would not miss any lightning and the recall would be equal to 1. However, there would be excessive amounts of false positives, which would lead to the degradation of precision. On the other hand, a model that does not predict any lightning event would have no false positives and the precision would be 1, but at the cost of increasing false negatives which, in turn, results in a low recall score. Therefore, the f1-score is used to indicate a balance between precision and recall since it is the harmonic mean of the two former scores. For the f1-score to have a high value, both precision and recall should be high. The equivalent metrics used in the IEC-62793 standard on "Protection against lightning—Thunderstorm warning systems" are POD (probability of detection) and FAR (false alarm ratio) which are related to recall and precision as follows:

$$POD = \frac{TP}{TP + FN} = recall \tag{15}$$

$$FAR = \frac{FP}{TP + FP} = 1 - precision \tag{16}$$

One important remark is that the predictor network outputs continuous values between zero and one indicating the probability of lightning in each pixel. Therefore, a thresholding process is needed to convert probability values to zero/one so that the mentioned metrics can be calculated. In Appendix B, the threshold used in this work is presented and justified.

Regarding the training process, the following steps have been performed and, if one wishes to reproduce the results, these remarks should be taken into account. The data were divided into three different sets, "train", "validation", and "test" with ratios 70%, 15%, and 15%, respectively. The train dataset is used during the feed-forward and back-propagation steps of training, the validation dataset is used for hyperparameter tuning, and the test set is used for the final evaluation of the system. The choice of train, validation, and test datasets is of crucial importance in the development of data-driven models. The datasets should be independent of each other and in the case of time-series meteorological data, they should not overlap. However, these datasets should reflect the same statistical distribution of the data. Therefore, the training, validation, and test sets were selected periodically every 5 days (120 h) according to the mentioned ratio from the lightning data

of the year 2019. The first 3.5 days (84 h) of each cycle are for training, the next 0.75 days (18 h) for validation, and the last 0.75 days (18 h) for testing. [52] The model is trained using the ADAM optimizer [53], which uses a checkpointer for tracking the best model and utilizes the reduce on plateau learning rate scheduler based on the overall f1-score. The codes are written in Python and the deep learning framework of Keras with Tensorflow as its backbone is used. The codes and data used in this research are accessible through the links presented in the Data Availability Statement section.

3. Results

In order to evaluate the performance of the proposed model, two experiments have been devised. In the first experiment, all models, i.e., Autoencoder (baseline), U-Net, and ResUNet both proposed models) were trained with three different sets of layer composition (number of filters/channels) to have a better understanding of the performance dependency on model size. The different architectures, i.e., Autoencoder, U-Net, and ResUNet, share the same number of layers and parameters (except the extra parameters needed for each specific architecture) so that the performance comparison becomes legitimate. The model with the most number of parameters, which we denote by "-large", consists of layers that increase the depth dimension of the processed data by 16, 32, 64, 128, 256, 512, 1024, and 2048 sequentially until the data reach the model information bottleneck in the center and then decreases the data dimension in the reverse order to generate the prediction in the output (see Appendix A and Figure A1 for more details). The model indicated by "-medium" ("-small"), lacks the layer 2048 (layers 1024 and 2048).

In the first experiment, the last layer of the model is set to output four different images (predictions), each for one of the four different lead times, i.e., 15, 30, 45, and 60 min. Therefore, The objective of the model is to predict for these lead times and the training is done by minimizing the loss function between predicted lightning maps and the ground truth available from historical data. After the training, the performance of the trained models are evaluated using the unseen historical data during training. The same is done for the persistence model, although due to its definition, there is no training involved. The mentioned metrics (Section 2.5), i.e., f1score (f1), precision (p), and recall (r), which are evaluated for each lead time by comparing predicted lightning maps and the ground truth, are presented in Table 1 (indexed by '−15', '−30', '−45', and '−60'). Some cherry-picked examples are shown in Figures 5–8. As a first observation, it can be seen from Figures 5–8 that the autoencoder model tends to predict contiguous regions while U-Net and ResUNet can better capture the sparsity of lightning and predict maps with a lower false positive rate.

Table 1. Multi output models' comparison.

Model Name	Parameters	f1-15	p-15	r-15	f1-30	p-30	r-30	f1-45	p-45	r-45	f1-60	p-60	r-60
persistence	0	0.75	0.75	0.75	0.62	0.62	0.62	0.52	0.52	0.52	0.43	0.43	0.43
Autoencoder-small	7M	0.52	0.42	0.67	0.50	0.42	0.64	0.48	0.41	0.61	0.46	0.38	0.58
Autoencoder-medium	28M	0.40	0.32	0.57	0.40	0.32	0.55	0.39	0.31	0.54	0.38	0.30	0.53
Autoencoder-large	113M	0.30	0.22	0.46	0.29	0.22	0.46	0.29	0.22	0.45	0.29	0.22	0.44
U-Net-small	8M	0.69	0.59	0.85	0.64	0.55	0.76	0.59	0.52	0.69	0.55	0.48	0.63
U-Net-medium	31M	0.69	0.59	0.84	0.64	0.55	0.76	0.59	0.52	0.69	0.54	0.48	0.64
U-Net-large	126M	0.69	0.59	0.84	0.64	0.55	0.76	0.59	0.52	0.69	0.55	0.49	0.63
ResUNet-small	8M	0.70	0.59	0.85	0.64	0.55	0.77	0.59	0.52	0.71	0.55	0.48	0.64
ResUNet-medium	33M	0.70	0.59	0.85	0.64	0.55	0.78	0.59	0.51	0.71	0.55	0.48	0.65
ResUNet-large	133M	0.70	0.59	0.86	0.64	0.55	0.78	0.59	0.51	0.70	0.55	0.48	0.65

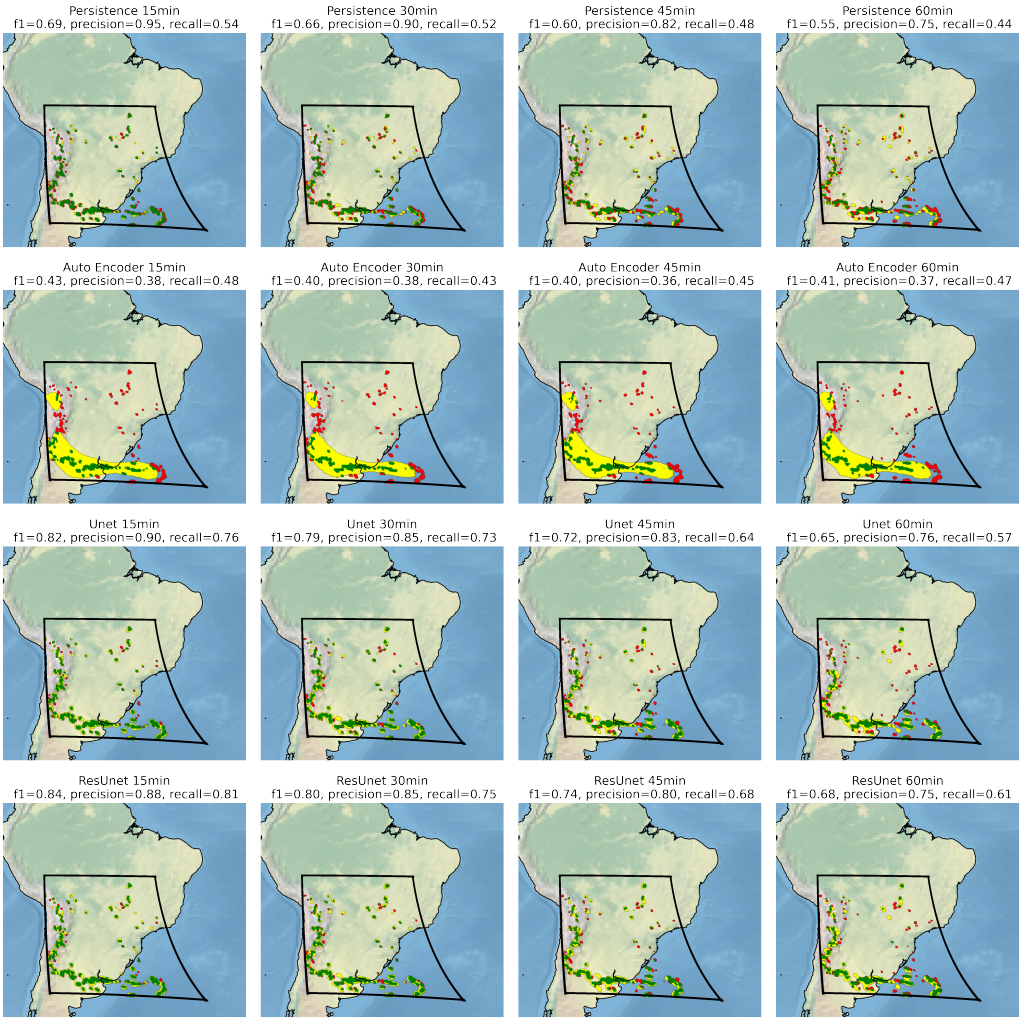

Figure 5. Prediction capability of the models, From top to bottom, (1) persistence, (2) Auto Encoder, (3) U-Net, and (4) ResUNet. Green dots represent the correctly predicted lightning, i.e., true positives. Yellow dots represent regions where the model falsely predicted a lightning occurrence, i.e., false positives. Finally, red dots show the occurred lightnings missed by prediction, i.e., false negatives.

Based on the scores presented in Table 1, one can see that the performance of the proposed models, i.e., U-Net and ResUNet, is almost independent of the model size and it appears that the performance has been saturated. Considering the amount of computation and memory usage required for operating medium and large models and the fact that there is no significant performance gain by their deployment, the small size model is the most practical solution for our problem. Moreover, by comparing the U-Net and ResUnet models, one can see that there is a small performance gain in using ResUnets. Therefore, the ResUnet-small has been chosen as the best working model for lightning prediction. From now on, we only consider the small models in the comparisons.

In order to better visualize the values presented in Table 1, in Figure 9, the metrics are presented against lead time for four models, namely: persistence, autoencoder-small, U-Net small, and ResUnet-small. Examining Table 1 and Figure 9, three observations can be made. First of all, as expected, the quality of prediction drops as one looks further into the future. The same can be seen for lightning prediction as the metrics show smaller values for larger lead times.

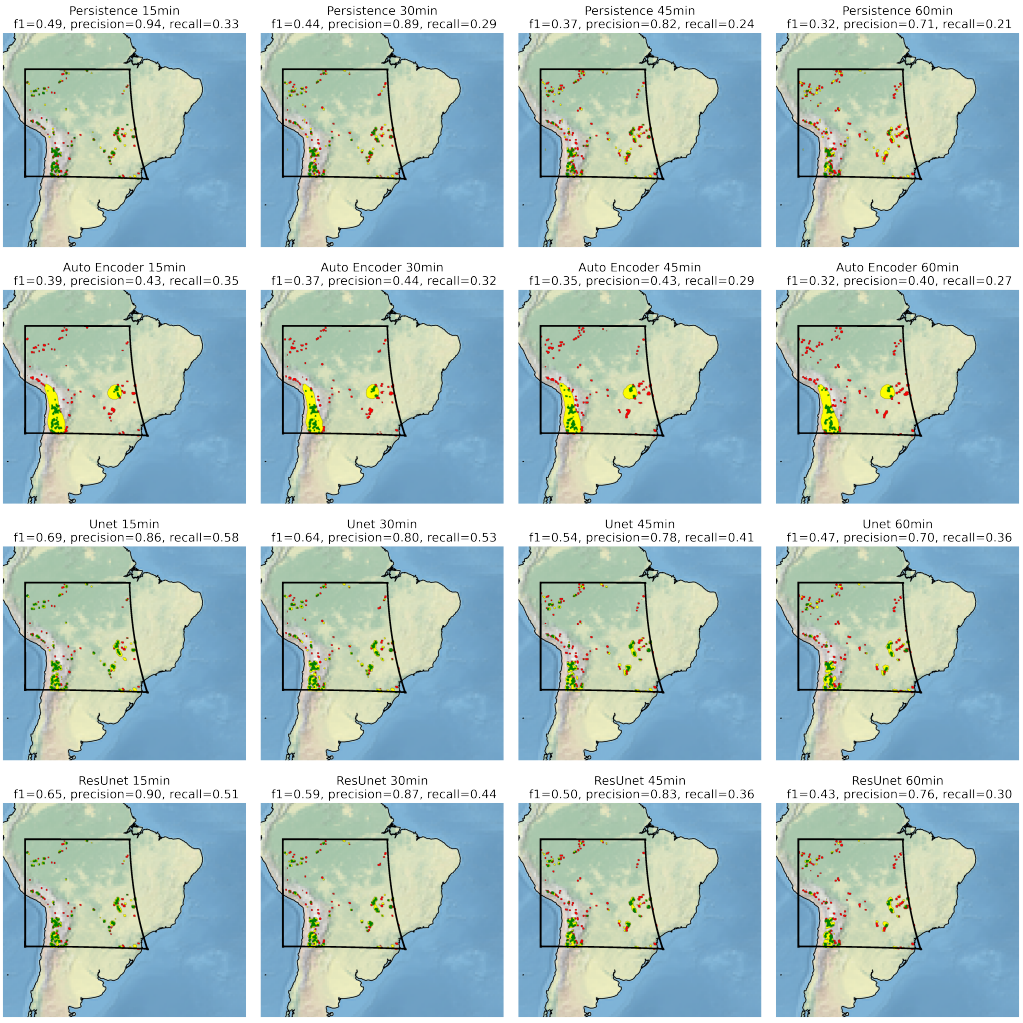

Figure 6. Sample prediction comparison. From top to bottom, (1) persistence, (2) Auto Encoder, (3) U-Net, and (4) ResUNet. Green dots represent the correctly predicted lightning, i.e., true positive. Yellow dots represent regions where model falsely predicted a lightning occurrence, i.e., false positive. Finally, red dots show the occurred lightnings missed by prediction, i.e., false negative.

Furthermore, it can be seen that for a lead time of 15 min, persistence works quite well and outperforms other models but falls behind for further lead times. On the other hand, the proposed residual U-Net and U-Net models suffer less degradation and maintain better performance as the lead time increases, overcoming the persistence for larger lead times. These models outperform persistence and autoencoders by about 20% depending on the lead time.

As mentioned earlier, the models consistently predict smaller lead times with higher f1 scores. Although we did not present any analysis of prediction performance for periods longer than one hour, we expect that the performance will decrease for these longer lead times as well due to the model limitations, absence of parameters involved in lightning initiation, and of course, the chaotic characteristics of the atmosphere. In the second experiment, the models were also trained on the same dataset but the objective was set to only predict one of the four lead times, 15, 30, 45, or 60 min. Each model has the same structure of layers except the last convolutional layer, which produces one output channel corresponding to the selected lead time as opposed to four output channels for all four lead times simultaneously. In addition to evaluating the model prediction capacity, it is also possible to verify whether predicting all the frames simultaneously is beneficial for learning the thundercloud and lightning activity movement pattern. The results are shown in Table 2 for four different lead times corresponding to "ResUNet-small". By comparing the corre-

sponding values of the two tables, the same performance can be observed, contradicting the limited network capacity hypothesis. Therefore, this degradation in performance can be attributed to the complex atmospheric behavior and hopefully can be addressed, at least to some extent, by deploying more complex models with fundamentally different structures and/or using more inclusive datasets.

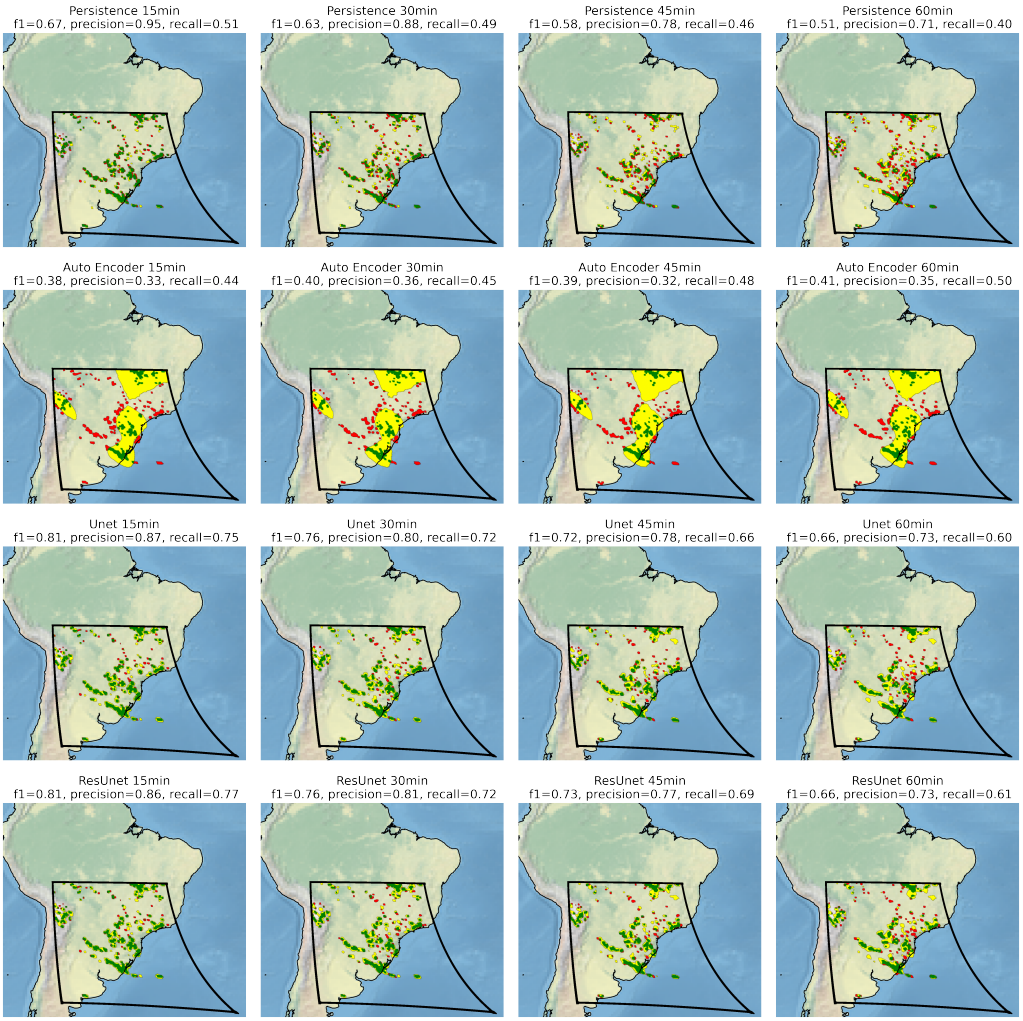

Figure 7. Sample prediction comparison. From top to bottom, (1) persistence, (2) Auto Encoder, (3) U-Net, and (4) ResUNet. Green dots represent the correctly predicted lightning, i.e., true positives. Yellow dots represent regions where model falsely predicted a lightning occurrence, i.e., false positives. Finally, red dots show the occurred lightnings missed by prediction, i.e., false negatives.

This second experiment can be interpreted from another angle. As we observe no performance gain by using a separate predictor for each lead time, we can use the architecture in the first experiment with a four-channel output, one for each lead time. By sharing the base encoder/decoder and using a four-channel output only in the last layer, the memory usage and computation cost can be reduced by a factor of four. in other words, the second experiment costs four-folded compared to the first one since it runs almost the same network for each lead time individually.

By examining the results of the two experiments, an efficient lightning predictor using a mixture of persistence for lead times of 15 min or less, and ResUnet for larger lead times could be proposed.

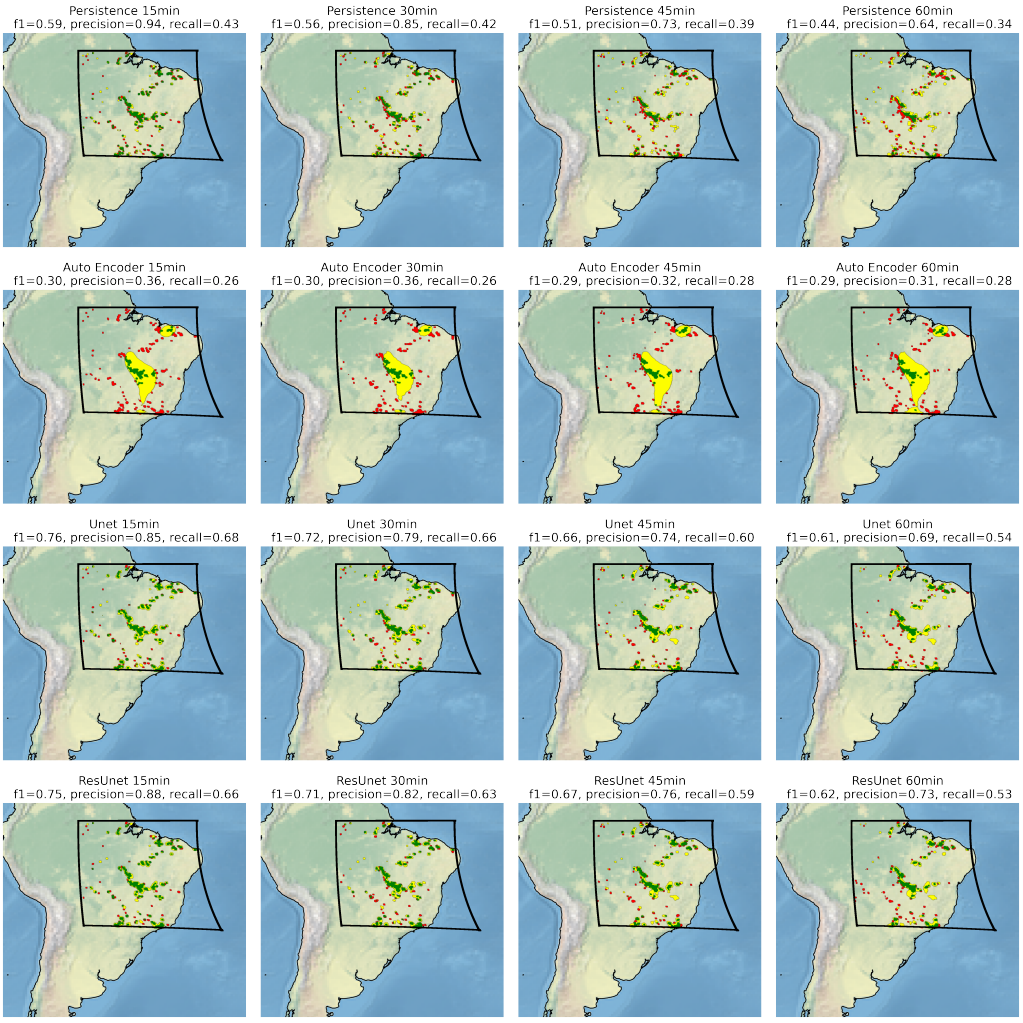

Figure 8. Sample prediction comparison. From top to bottom, (1) persistence, (2) Auto Encoder, (3) U-Net, and (4) ResUNet. Green dots represent the correctly predicted lightning, i.e., true positives. Yellow dots represent regions where model falsely predicted a lightning occurrence, i.e., false positives. Finally, red dots show the occurred lightnings missed by prediction, i.e., false negatives.

Due to the inherent structure of our proposed method, a direct comparison of its performance is not possible with the ones available in the literature. These include the fact that we only utilize lightning data for prediction, we train our model on significantly less amount of data, and the fact that there is a different temporal and spatial granularity for each model. However, by considering the work of Tippett et al. [23], Leinonen et al. [27], and Song et al. [36], we can see that our metrics, namely precision (POD) and recall (1-FAR) lies in the same range as the ones presented by these papers. This is an advantage point for our model as it uses less data for training.

Table 2. Single output models' comparison.

Model Name	Parameters	Loss	Precision	Recall	f1-15	f1-30	f1-45	f1-60
persistence	0	1.33	0.57	0.58	0.74	0.62	0.51	0.42
ResUnet 15 min	8M	0.68	0.65	0.79	0.72	-	-	-
ResUnet 30 min	8M	0.77	0.58	0.73	-	0.65	-	-
ResUnet 45 min	8M	0.86	0.53	0.67	-	-	0.59	-
ResUnet 60 min	8M	0.93	0.46	0.66	-	-	-	0.54

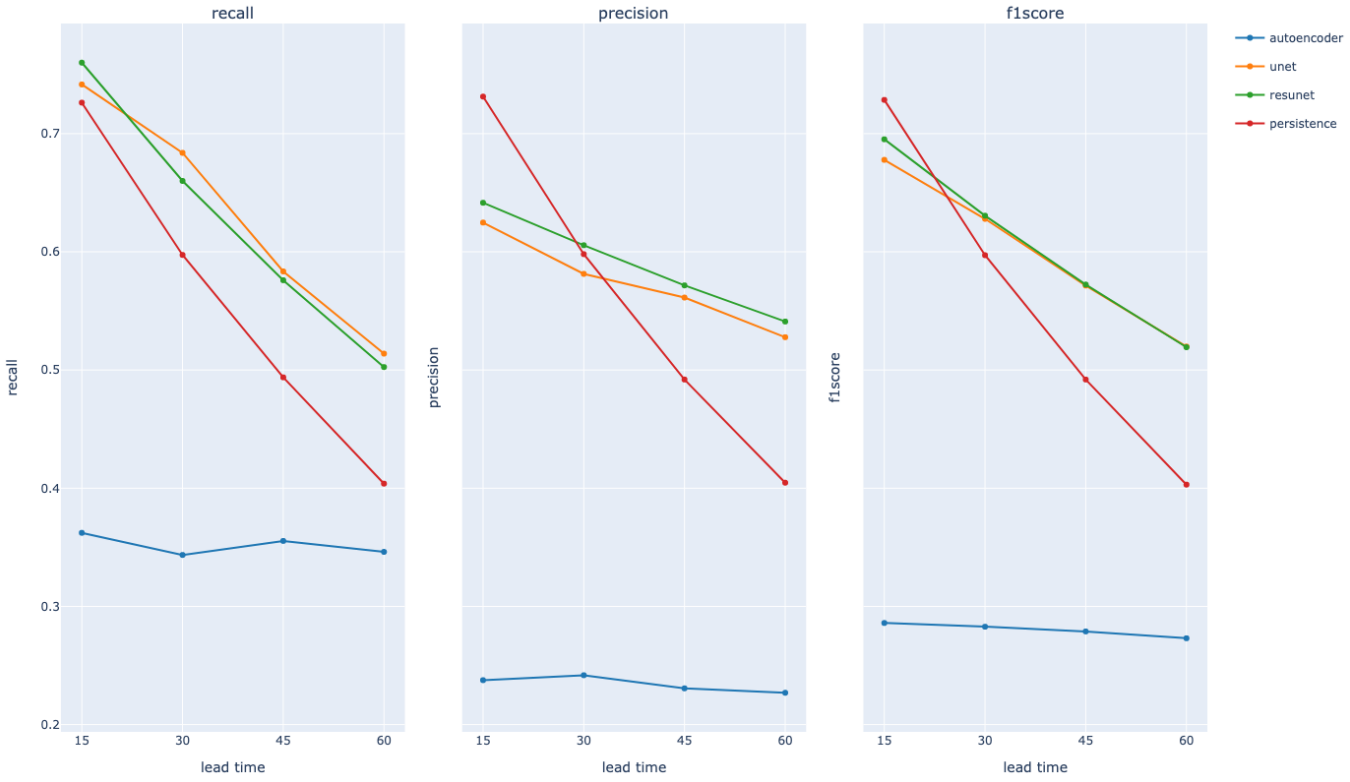

Figure 9. Recall, Precision, and f1score vs. lead time for different models (small version).

4. Conclusions

In this work, we presented an autoregressive approach based on deep neural networks to nowcast lightning for lead times of up to one hour in 15-minutes increments. The model uses solely lightning data as an input parameter and has similar performance in terms of precision (POD) and recall (1-FAR) compared to existing models in the literature. However, the advantage of our model is that it uses a significantly lower amount of data for its training. We have seen that by using U-Net and ResUNet, we can improve the lightning prediction for lead times greater than 15 min with respect to the persistence model. More precisely, we can see an improvement of 0.02 to 0.12 in the f1 score of different channels, see Table 1. It was also found that the persistence model's performance was comparable to and even outperforming U-Net and ResUNet for small lead times, namely 15 min. Furthermore, we observed that the prediction capability saturates as we increase the model's size. This encourages us to use smaller models to reduce memory usage and computation costs. Moreover, we showed that the computation cost can be reduced by predicting all lead times simultaneously without suffering from performance loss. Therefore, as an improved baseline for future lightning activity prediction, we propose a combination of ResUNet-small for lead times greater than 15 min and the persistence model otherwise.

To summarize and conclude, in this work, we have used the recent lightning activity data to predict the future lightning activity by treating the problem as an autoregressive model. Training a machine learning-based model requires huge amounts of data and, for this paper, we have used only one year of data. In our future work, we are planning to extend the study using more data spanned over several years. Moreover, other network architectures should be considered, including recurrent models, diffusion models, and transformer-based models like ViT, to deal with the problem of lightning nowcasting. So far, we have analyzed the satellite lightning data over the American continent, specifically the northern part of South America. However, our model is portable to other regions of the Earth. In our future work, we will collect data from satellites over Europe/Africa and East Asia, and we will investigate the model's performance for these regions as well.

Author Contributions: Conceptualization, E.M., A.M., F.R. and M.R.; methodology, E.M. and A.M.; software, E.M. and A.M.; validation, E.M. and A.M.; formal analysis, E.M. and A.M.; investigation, E.M.; resources, C.T., F.R. and M.R.; data curation, E.M. and C.T.; writing—original draft preparation, E.M.; writing—review and editing, E.M., A.M., C.T., M.R. and F.R.; visualization, E.M. and A.M.; supervision, C.T., F.R. and M.R.; project administration, C.T., F.R. and M.R.; funding acquisition, F.R. and M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by JiangSu Electric Power Co., Ltd. Suzhou Branch, Suzhou, China and by Swiss National Science Foundation, Grant 200020_204235.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this research is provided by NASA and NOAA and it is publicly available from different sources, including AWS S3 bucket at https://noaa-goes16.s3.amazonaws.com/index.html, accessed on 1 September 2021. The codes for efficiently downloading the dataset can be found in https://github.com/ehsan27770/lightning_nowcasting_data_retrieval, accessed on 16 March 2023 and are accessible via https://doi.org/10.5281/zenodo.7773771 [54], accessed on 27 March 2023. The codes for data preprocessing and dataset creation are documented at https://github.com/ehsan27770/lightning_nowcasting_data_preparation, accessed on 16 March 2023 and are accessible via https://doi.org/10.5281/zenodo.7773764 [55], accessed on 27 March 2023. The main codes of this paper and the models used in evaluation can be found in https://github.com/ehsan27770/lightning_nowcasting_model_training, accessed on 16 March 2023 and are accessible via https://doi.org/10.5281/zenodo.7773758 [56], accessed on 27 March 2023. No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: Chong Tong is an employee of State Grid Jiangsu Electric Power Co. Ltd Suzhou Branch, Suzhou, China. The paper reflects the views of the scientists and not the company.

Abbreviations

The following abbreviations are used in this manuscript:

WRF	Weather research and forecasting
CAPE	Convective available potential energy
LSTM	Long short-term memory
ConvLSTM	Convolutional LSTM
ResUNet	Residual U-Net
MRMS	multi-radar multi-sensor
NOAA	National Oceanic and Atmospheric Administration
GLM	Geostationary Lightning Mapper
ABI	Advanced Baseline Imager
GOES	Geostationary Operational Environmental Satellites
TP	True positive
FP	False positive
FN	False negative
POD	Probability of detection
FAR	False alarm ratio

Appendix A. Models' Architecture

In this section, the architecture of Residual Unet, used in this paper, is described. Moreover, the architecture of autoencoder and U-Net are also briefly described as they are simplified versions of ResUnet. The schematic architecture of ResUNet-small is shown in Figure A1. (The network structure is generated with the VisualKeras library [57]). ResUNet consists of many building blocks which are described as follows. The first layer that acts on the input image is a convolution layer with a kernel size of 3×3, increasing the number of channels in the output of the layer. The output is passed through a batch normalization layer helping to keep the values in the same range for numerical stability, and also ensuring proper and efficient backpropagation of the loss. Then, a similar

convolution layer, batch normalization layer, and convolution layer are applied with the only difference that convolution layers maintain the same number of output channels. In order to improve the training efficiency and backpropagation of the gradient, there are short skip connections (residual connections) from the output of the first convolution layer to the output of the last convolution layer, where the output and skip connections are added together. These short skip connections utilize 1×1 convolutions to match the number of channels from the input to the output. Then, a batch normalization layer followed by a max-pooling layer reduces the spatial dimensions of the output. This block, i.e., convolution layers, batch normalization layers, and max-pooling layer, forms an encoder block and is cascaded several times to generate a latent space representation of the image, which has more channels compared to the initial image and has less spatial resolution. These consecutive layers increase the number of channels from 8 to 64, 128, 256, 512, and 1024 sequentially while reducing the spatial resolution by a factor of 2 from one block to the next. The intention is to summarize the image in a compressed latent variable to avoid overfitting. After summarizing the input image into the latent space, a series of up-scaling blocks are applied sequentially to the original image to restore the spatial resolution. Each upscaling block consists of a 2D upscaling layer to counter the effect of max-pooling applied in the encoder part. In order to learn more efficiently and keep the initial structure of the data, a series of long skip connections are utilized that connect the output of each encoder block to the input of each decoder block. The long skip connections bring the output of the encoder blocks into the input of the decoder where they are concatenated with the output of the upscaling layer. The concatenated output is followed by three sets of batch normalization and convolution layers where they reduce the number of output channels to keep the density of information consistent. There are again short skip connections from the output of the first batch normalization layer to the output of the last convolution layer. This combination of upscaling, concatenation, batch normalization, and convolution layers along with the short skip connection form the decoder block. These decoder blocks sequentially change the number of channels in reversed order compared to the encoder, namely 1024, 512, 256, 128, and 64. Finally, at the output of the last convolution layer (output with 64 channels), another convolution layer with a kernel size of 1×1 is applied which will perform the lightning map generation. The U-Net and autoencoder models have a similar structure and number of layers. The only difference for U-Net is the absence of short skip connections between the input and the output of each encoder or decoder block. For the autoencoder, both short skip connections and long skip connections are absent. Therefore, there is no direct connection from input to output to facilitate the flow of gradients, and that is the reason why the autoencoder is not trained well on the data.

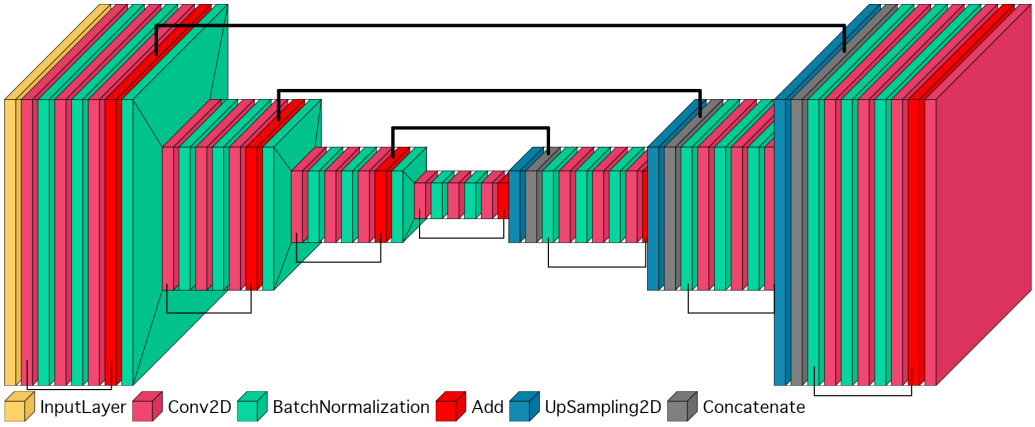

Figure A1. ResUNet. For visualization purposes, this is a smaller version (less layers) compared to the actual models used in this work.

Appendix B. Optimal Threshold

As the output of the presented models are probability of lightning (between zero and one after applying a sigmoid function), a thresholding process is needed to carry out the generation of final prediction, i.e., assuming values above the threshold to represent a lightning occurrence. All of the previous figures and results are generated by thresholding the output of the model with a value of "0.1". This value has been selected using the following approach. The threshold was varied from zero to one and the corresponding precision, recall, and f1 score were recorded. Figure A2 reflects the change in f1 score for each individual time frame with respect to the threshold value. In Figure A3, the precision and recall corresponding to the same threshold are drawn against each other. As in can be seen from both figures, the choice of "0.1" for the threshold results in the highest value for the f1 score.

Furthermore, in these curves one can observe that the persistence model curve is higher than the rest in the 15 min lead time, but drops below them for the rest of the lead times. This is the same behavior mentioned in Tables 1 and 2.

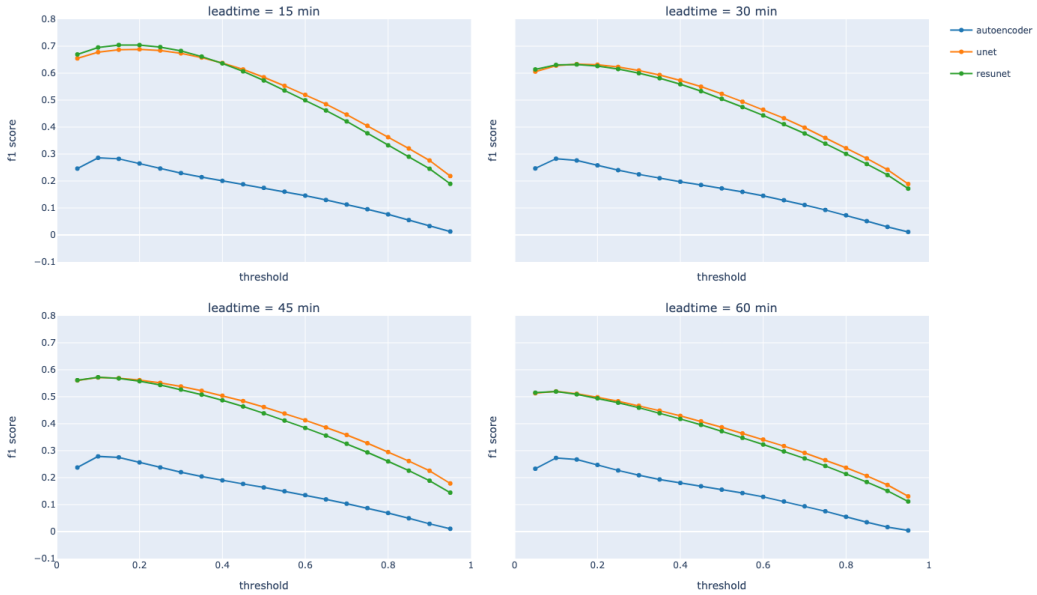

Figure A2. f1 score vs. detection threshold.

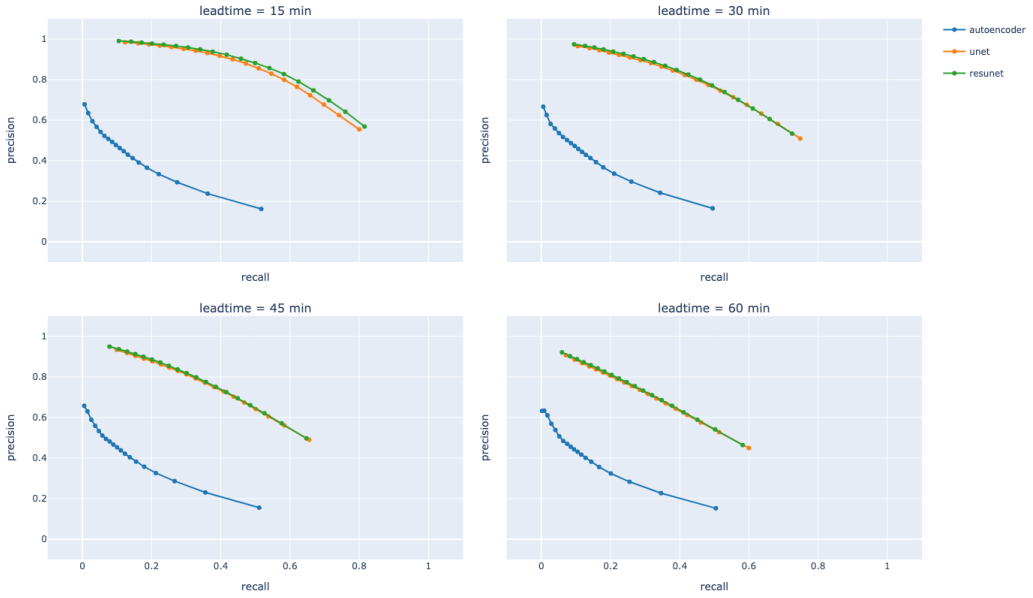

Figure A3. Precision vs. recall curve made by changing the detection threshold.

References

1. Cooper, M.A.; Holle, R.L. (Eds.) Current Global Estimates of Lightning Fatalities and Injuries. In *Reducing Lightning Injuries Worldwide*; Springer Natural Hazards; Springer International Publishing: Cham, Switzerland, 2019; pp. 65–73. [CrossRef]
2. Cerveny, R.S.; Bessemoulin, P.; Burt, C.C.; Cooper, M.A.; Cunjie, Z.; Dewan, A.; Finch, J.; Holle, R.L.; Kalkstein, L.; Kruger, A.; et al. WMO Assessment of Weather and Climate Mortality Extremes: Lightning, Tropical Cyclones, Tornadoes, and Hail. *Weather Clim. Soc.* **2017**, *9*, 487–497. [CrossRef]
3. Watson, A.I.; Holle, R.L.; López, R.E.; Ortiz, R.; Nicholson, J.R. Surface Wind Convergence as a Short-Term Predictor of Cloud-to-Ground Lightning at Kennedy Space Center. *Weather Forecast.* **1991**, *6*, 49–64. [CrossRef]
4. Watson, A.I.; López, R.E.; Holle, R.L.; Daugherty, J.R. The Relationship of Lightning to Surface Convergence at Kennedy Space Center: A Preliminary Study. *Weather Forecast.* **1987**, *2*, 140–157. [CrossRef]
5. Yokoyama, S.; Honjo, N.; Yasuda, Y.; Member, K. Causes of wind turbine blade damages due to lightning and future research target to get better protection measures. In Proceedings of the 2014 International Conference on Lightning Protection (ICLP), Shanghai, China, 11–18 October 2014; pp. 823–830. [CrossRef]
6. Braam, H.; Ramakers, S.G.M.; Rademakers, L.; Wessels, H.; Prins, R.K.N.J.; Lok, R.; Leunis, L. *Lightning Damage of OWECS Part 3: "Case Studies"*; ECN, Energy Research Centre of the Netherlands: Petten, The Netherlands, 2002. Available online: https://publications.ecn.nl/ECN-C--02-054 (accessed on 21 March 2023).
7. Curran, E.; Holle, R.; López, R. Lightning Casualties and Damages in the United States from 1959 to 1994. *J. Clim.* **2000**, *13*, 3448–3464. [CrossRef]
8. Mansouri, E.; Mostajabi, A.; Schulz, W.; Diendorfer, G.; Rubinstein, M.; Rachidi, F. On the Use of Benford's Law to Assess the Quality of the Data Provided by Lightning Locating Systems. *Atmosphere* **2022**, *13*, 552. [CrossRef]
9. MacGorman, D.R.; Straka, J.M.; Ziegler, C.L. A Lightning Parameterization for Numerical Cloud Models. *J. Appl. Meteorol.* **2001**, *40*, 459–478. [CrossRef]
10. Mansell, E.R.; MacGorman, D.R.; Ziegler, C.L.; Straka, J.M. Simulated three-dimensional branched lightning in a numerical thunderstorm model. *J. Geophys. Res. Atmos.* **2002**, *107*, ACL 2-1–ACL 2-12. [CrossRef]
11. Mansell, E.R.; MacGorman, D.R.; Ziegler, C.L.; Straka, J.M. Charge structure and lightning sensitivity in a simulated multicell thunderstorm. *J. Geophys. Res. Atmos.* **2005**, *110*, D12101. [CrossRef]
12. Helsdon, J.H.; Wojcik, W.A.; Farley, R.D. An examination of thunderstorm-charging mechanisms using a two-dimensional storm electrification model. *J. Geophys. Res. Atmos.* **2001**, *106*, 1165–1192. [CrossRef]
13. Fierro, A.O.; Mansell, E.R.; Ziegler, C.L.; MacGorman, D.R. Explicit electrification and lightning forecast implemented within the WRF-ARW model. In Proceedings of the XV International Conference on Atmospheric Electricity, Norman, OK, USA, 15–20 June 2014; p. 11.
14. Fierro, A.O.; Mansell, E.R.; MacGorman, D.R.; Ziegler, C.L. The Implementation of an Explicit Charging and Discharge Lightning Scheme within the WRF-ARW Model: Benchmark Simulations of a Continental Squall Line, a Tropical Cyclone, and a Winter Storm. *Mon. Weather Rev.* **2013**, *141*, 2390–2415. [CrossRef]
15. Guichard, F.; Couvreux, F. A short review of numerical cloud-resolving models. *Tellus A Dyn. Meteorol. Oceanogr.* **2017**, *69*, 1373578. [CrossRef]
16. Field, P.R.; Roberts, M.J.; Wilkinson, J.M. Simulated Lightning in a Convection Permitting Global Model. *J. Geophys. Res. Atmos.* **2018**, *123*, 9370–9377. [CrossRef]
17. Dowdy, A.J. Seasonal forecasting of lightning and thunderstorm activity in tropical and temperate regions of the world. *Sci. Rep.* **2016**, *6*, 20874. [CrossRef]
18. Romps, D.M.; Charn, A.B.; Holzworth, R.H.; Lawrence, W.E.; Molinari, J.; Vollaro, D. CAPE Times P Explains Lightning Over Land But Not the Land-Ocean Contrast. *Geophys. Res. Lett.* **2018**, *45*, 12623–12630. [CrossRef]
19. Bates, B.C.; Dowdy, A.J.; Chandler, R.E. Lightning Prediction for Australia Using Multivariate Analyses of Large-Scale Atmospheric Variables. *J. Appl. Meteorol. Climatol.* **2018**, *57*, 525–534. [CrossRef]
20. Lopez, P. A Lightning Parameterization for the ECMWF Integrated Forecasting System. *Mon. Weather Rev.* **2016**, *144*, 3057–3075. [CrossRef]
21. Price, C.; Rind, D. A simple lightning parameterization for calculating global lightning distributions. *J. Geophys. Res. Atmos.* **1992**, *97*, 9919–9933. [CrossRef]
22. Lynn, B.H.; Yair, Y.; Price, C.; Kelman, G.; Clark, A.J. Predicting Cloud-to-Ground and Intracloud Lightning in Weather Forecast Models. *Weather Forecast.* **2012**, *27*, 1470–1488. [CrossRef]
23. Tippett, M.K.; Koshak, W.J. A Baseline for the Predictability of U.S. Cloud-to-Ground Lightning. *Geophys. Res. Lett.* **2018**, *45*, 10719–10728. [CrossRef]
24. Golding, B.W. Nimrod: A system for generating automated very short range forecasts. *Meteorol. Appl.* **1998**, *5*, 1–16. [CrossRef]
25. Rojas-Campos, A.; Langguth, M.; Wittenbrink, M.; Pipa, G. Deep learning models for generation of precipitation maps based on numerical weather prediction. *Geosci. Model Dev.* **2023**, *16*, 1467–1480. [CrossRef]
26. Serifi, A.; Günther, T.; Ban, N. Spatio-Temporal Downscaling of Climate Data Using Convolutional and Error-Predicting Neural Networks. *Front. Clim.* **2021**, *3*, 656479. [CrossRef]
27. Leinonen, J.; Hamann, U.; Germann, U. Seamless Lightning Nowcasting with Recurrent-Convolutional Deep Learning. *Artif. Intell. Earth Syst.* **2022**, *1*, e220043. [CrossRef]

28. Geng, Y.a.; Li, Q.; Lin, T.; Jiang, L.; Xu, L.; Zheng, D.; Yao, W.; Lyu, W.; Zhang, Y. LightNet: A Dual Spatiotemporal Encoder Network Model for Lightning Prediction. In *KDD '19, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 2439–2447. [CrossRef]
29. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.k.; Woo, W.C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*; Curran Associates, Inc.: Red Hook, NY, USA, 2015.
30. Lin, T.; Li, Q.; Geng, Y.A.; Jiang, L.; Xu, L.; Zheng, D.; Yao, W.; Lyu, W.; Zhang, Y. Attention-Based Dual-Source Spatiotemporal Neural Network for Lightning Forecast. *IEEE Access* **2019**, *7*, 158296–158307. [CrossRef]
31. Brodehl, S.; Müller, R.; Schömer, E.; Spichtinger, P.; Wand, M. End-to-End Prediction of Lightning Events from Geostationary Satellite Images. *Remote Sens.* **2022**, *14*, 3760. [CrossRef]
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016, Proceedings, Part IV 14;* Springer: Berlin/Heidelberg, Germany, 2016.
33. Diakogiannis, F.I.; Waldner, F.; Caccetta, P.; Wu, C. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J. Photogramm. Remote Sens.* **2020**, *162*, 94–114. [CrossRef]
34. Zhang, Z.; Liu, Q.; Wang, Y. Road Extraction by Deep Residual U-Net. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 749–753. [CrossRef]
35. Weigert, M.; Schmidt, U.; Boothe, T.; Müller, A.; Dibrov, A.; Jain, A.; Wilhelm, B.; Schmidt, D.; Broaddus, C.; Culley, S.; et al. Content-aware image restoration: Pushing the limits of fluorescence microscopy. *Nat. Methods* **2018**, *15*, 1090–1097. [CrossRef]
36. Song, G.; Li, S.; Xing, J. Lightning nowcasting with aerosol-informed machine learning and satellite-enriched dataset. *npj Clim. Atmos. Sci.* **2023**, *6*, 1–10. [CrossRef]
37. Soula, S. Lightning and Precipitation. In *Lightning: Principles, Instruments and Applications: Review of Modern Lightning Research*; Betz, H.D., Schumann, U., Laroche, P., Eds.; Springer: Dordrecht, The Netherlands, 2009; pp. 447–463. [CrossRef]
38. Agrawal, S.; Barrington, L.; Bromberg, C.; Burge, J.; Gazen, C.; Hickey, J. Machine Learning for Precipitation Nowcasting from Radar Images. *arXiv* **2019**, arXiv:1912.12132. https://doi.org/10.48550/arXiv.1912.12132.
39. Sønderby, C.K.; Espeholt, L.; Heek, J.; Dehghani, M.; Oliver, A.; Salimans, T.; Agrawal, S.; Hickey, J.; Kalchbrenner, N. MetNet: A Neural Weather Model for Precipitation Forecasting. *arXiv* **2020**, arXiv:2003.12140. https://doi.org/10.48550/arXiv.2003.12140.
40. Espeholt, L.; Agrawal, S.; Sønderby, C.; Kumar, M.; Heek, J.; Bromberg, C.; Gazen, C.; Carver, R.; Andrychowicz, M.; Hickey, J.; et al. Deep learning for twelve hour precipitation forecasts. *Nat. Commun.* **2022**, *13*, 5145. . [CrossRef]
41. Albrecht, R.I.; Goodman, S.J.; Buechler, D.E.; Blakeslee, R.J.; Christian, H.J. Where Are the Lightning Hotspots on Earth? *Bull. Am. Meteorol. Soc.* **2016**, *97*, 2051–2068. [CrossRef]
42. Neumann, C.J. *Frequency and Duration of Thunderstorms at Cape Kennedy, Part 1*; Technical Report NASA-CR-97843; Weather Bureau; Silver Spring: Montgomery, MD, USA, 1968.
43. Mecikalski, J.; Jewett, C.; Carey, L.; Zavodsky, B.; Stano, G.; Chronis, T. An Integrated 0-1 Hour First-Flash Lightning Nowcasting, Lightning Amount and Lightning Jump Warning Capability. In Proceedings of the Conference on the Meteorological Applications of Lightning Data, Phoenix, AZ, USA, 4–8 January 2015.
44. Mostajabi, A.; Finney, D.L.; Rubinstein, M.; Rachidi, F. Nowcasting lightning occurrence from commonly available meteorological parameters using machine learning techniques. *npj Clim. Atmos. Sci.* **2019**, *2*, 41. [CrossRef]
45. Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507. [CrossRef]
46. Schwenk, H.; Bengio, Y. Training Methods for Adaptive Boosting of Neural Networks. In *Advances in Neural Information Processing Systems 10 (NIPS 1997)*; Jordan, M., Kearns, M., Solla, S., Eds.; MIT Press: Cambridge, MA, USA, 1997; Volume 10.
47. Japkowicz, N.; Hanson, S.J.; Gluck, M.A. Nonlinear Autoassociation Is Not Equivalent to PCA. *Neural Comput.* **2000**, *12*, 531–545. [CrossRef]
48. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2013**, arXiv:1312.6114. https://doi.org/10.48550/arXiv.1312.6114.
49. Zhou, K.; Zheng, Y.; Dong, W.; Wang, T. A Deep Learning Network for Cloud-to-Ground Lightning Nowcasting with Multisource Data. *J. Atmos. Ocean. Technol.* **2020**, *37*, 927–942. [CrossRef]
50. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015, Proceedings, Part III 18*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
51. Shamir, R.R.; Duchin, Y.; Kim, J.; Sapiro, G.; Harel, N. Continuous Dice Coefficient: A Method for Evaluating Probabilistic Segmentations. *arXiv* **2019**, arXiv:1906.11031.
52. Schultz, M.G.; Betancourt, C.; Gong, B.; Kleinert, F.; Langguth, M.; Leufen, L.H.; Mozaffari, A.; Stadtler, S. Can deep learning beat numerical weather prediction? *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.* **2021**, *379*, 20200097. [CrossRef]
53. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980.
54. Mansouri, E. ehsan27770/Lightning_Nowcasting_Data_Retrieval. 2023. Available online: https://zenodo.org/records/7773771 (accessed on 27 March 2023).
55. Mansouri, E. ehsan27770/Lightning_Nowcasting_Data_Preparation. 2023. Available online: https://zenodo.org/records/7773764 (accessed on 27 March 2023).

56. Mansouri, E. ehsan27770/Lightning_Nowcasting_Model_Training. 2023. Available online: https://zenodo.org/records/7773758 (accessed on 27 March 2023).
57. Gavrikov, P. Visualkeras. GitHub Repository. 2020. Available online: https://github.com/paulgavrikov/visualkeras (accessed on 25 March 2023).