



Article

Deep Learning Model for Global Spatio-Temporal Image Prediction

Dušan P. Nikezić , Uzahir R. Ramadani , Dušan S. Radivojević, Ivan M. Lazović and Nikola S. Mirkov 

Vinča Institute of Nuclear Sciences, National Institute of the Republic of Serbia, University of Belgrade, 11000 Belgrade, Serbia

* Correspondence: dusan@vin.bg.ac.rs

Abstract: Mathematical methods are the basis of most models that describe the natural phenomena around us. However, the well-known conventional mathematical models for atmospheric modeling have some limitations. Machine learning with Big Data is also based on mathematics but offers a new approach for modeling. There are two methodologies to develop deep learning models for spatio-temporal image prediction. On these bases, two models were built—ConvLSTM and CNN-LSTM—with two types of predictions, i.e., sequence-to-sequence and sequence-to-one, in order to forecast Aerosol Optical Thickness sequences. The input dataset for training was NASA satellite imagery MODAL2_E_AER_OD from Terra/MODIS satellites, which presents global Aerosol Optical Thickness with an 8 day temporal resolution from 2000 to the present. The obtained results show that the ConvLSTM sequence-to-one model had the lowest RMSE error and the highest Cosine Similarity value. The advantages of the developed DL models are that they can be executed in milliseconds on a PC, can be used for global-scale Earth observations, and can serve as tracers to study how the Earth's atmosphere moves. The developed models can be used as transfer learning for similar image time-series forecasting models.

Keywords: deep learning model; spatio-temporal image prediction; aerosol; climate change

MSC: 62M30; 62M45; 62P12



Citation: Nikezić, D.P.; Ramadani, U.R.; Radivojević, D.S.; Lazović, I.M.; Mirkov, N.S. Deep Learning Model for Global Spatio-Temporal Image Prediction. *Mathematics* **2022**, *10*, 3392. <https://doi.org/10.3390/math10183392>

Academic Editors: Fei Qi, Chao Liu and Yiping Wang

Received: 15 August 2022

Accepted: 14 September 2022

Published: 19 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The need for atmospheric spatio-temporal forecasts has led to the development of many well-known mathematical models, such as the Gaussian dispersion model, which is suitable for fast but less accurate predictions, because it does not require a lot of computational time [1]. Euler–Lagrange mathematical model is more accurate but also more demanding in terms of calculations. The Euler model is used to give spatial prediction as an explicit method for the numerical integration of ordinary differential equations. The Lagrangian method is used for forecasting a trajectory path over time with implicit time-dependence through a set of generalized coordinates. In [2], a spatio-temporal image pattern prediction method based on the Navier–Stokes equation and non-dimensional advection–anisotropic-diffusion equation is presented for an image sequence.

The aforementioned models are based on well-known equations, but since the modeling and prediction of complex and nonlinear function approximations do not give exact solutions (ground truth), such models have errors. As an alternative to classical models, Machine Learning (ML) and Artificial neural networks (ANNs) with Big Data offer the new possibility to capture spatial, temporal and spatio-temporal correlations more rapidly with fewer errors, revealing new patterns of atmospheric non-linearity.

Deep learning (DL), as part of ML, is based on mathematics, e.g., numerical optimization problems with gradient descent. Automatic differentiation is used for backpropagation in the training of neural networks (NNs) to estimate the correct polynomial for regression

predictive modeling with minimum error by hyperparameter optimization. Mathematics explains how algorithms work, e.g., statistical, probability and linear algebraic tools. The implementation of DL demands a lot of data, and size does matter. Among other things, deep neural networks (DNN) are very useful and successful with time-stepped data [3].

Satellite remote sensing offers imagery datasets which are sorted by date, usually sequentially (time-series). This approach has proved to be very useful in climate and environmental analyses. In this study, the Aerosol Optical Thickness (AOT) dataset from NASA Earth Observations (NEO) was used [4]. AOT is the measurement of the quantity and distribution of aerosols that have a serious impact on environment, climate and human health.

Spatio-temporal image predictions in DL can be done using Convolutional Neural Networks (CNNs), followed by Long Short-Term Memory (LSTM), where the CNN (Convolution2D) captures the spatial features and LSTM captures the correlations over time. Another method involves a new class, ConvLSTM2D in Keras, which captures spatial and temporal components at the same time.

The aim of the presented study is to determine which methodology shows better results in predictions of image time series. Few studies have attempted this, especially for sequences of images (2D input and 2D output). The first approach is to develop the model with Conv2D layers followed by LSTM layers; the problem is to determine how to transform a 2D array to 1D as input to the LSTM layer, and then to return to 2D as an image). The second approach is to develop a model with perspective ConvLSTM2D layers in order to replace the old model design (CNN followed by LSTM). For this reason, in this study, one model is named CNN-LSTM and the other ConvLSTM.

Furthermore, prediction problems with times series can be divided into three main types [5]: one-to-sequence, sequence-to-one and sequence-to-sequence. For the developed models, the MODAL2_E_AER_OD dataset was used for sequence-to-sequence and sequence-to-one predictions. Thus, four models for global spatio-temporal image forecasting were developed for the sake of comparison: ConvLSTM sequence-to-sequence, ConvLSTM sequence-to-one, CNN-LSTM sequence-to-sequence, and CNN-LSTM sequence-to-one.

In our literature review, we did not find spatio-temporal image forecasts with similar models. The goal of our study was to implement ML with satellite remote sensing in order to more precisely measure aerosols as one of the main uncertainties in climate modeling.

2. Data and Methodology

2.1. Literature Review and Related Work

Regularization is an efficient way to improve the generalization ability of deep CNNs, because it makes it possible to train more complex models while maintaining lower overfitting [6]. Regularization techniques such as Learning rate, EarlyStopping and ReduceLROnPlateau with Dropout and BatchNormalization layers were implemented in this study in order to reduce the overfitting problem. Another useful technique is the data augmentation method at the training stage in DL, which can significantly improve the accuracy of the deep CNN [7].

Recurrent neural networks (RNNs) have been widely adopted in research areas concerned with sequential data, such as text, audio, video. LSTM with gate functions in the cell structure has become the focus of DL [8]. Furthermore, in [9], LSTM was employed to capture the temporal features from a sequence of ultrasound images.

DL architectures have been developed to deal with raw structured data and to enable the rapid analysis of structured input data (sequences, images, videos) to predict complex outputs with unprecedented accuracy [10]. A systematic review on traditional NNs and the most advanced DL methods in environmental remote sensing applications is presented in [11]. According to the authors of that study, the combination of physical and the DL models is a promising direction.

In [12], satellite-based aerosol retrievals are used to provide global spatially distributed estimates of atmospheric aerosol parameters that are commonly needed in applications

such as estimations of atmospherically corrected satellite data products, climate modeling and air quality monitoring. A one-dimensional CNN (which acts as a spatial feature extractor) was integrated with a one-directional fully gated recurrent unit (GRU) for more timely and accurate air quality forecasting, with the ultimate objective of contributing to better public health protection and air pollution prevention [13].

The motivation for developing the ConvLSTM and CNN-LSTM models lies in the lack of literature. Only a few studies, such as [14,15], have provided comparative analyses of the ConvLSTM and CNN-LSTM methodologies. Furthermore, spatio-temporal image predictions are very rare, especially with comparing these two methodologies when used for the same purpose. Usually, the base comparative model is LSTM, but both studies [14,15] and the experiences of the authors of the present study show that both methodologies yield better results than LSTM. As such, the aim of this study is to undertake a credible comparison of the ConvLSTM and CNN-LSTM methodologies for sequence image prediction.

2.2. Pre-Training Process

Satellite remote sensing for climate modeling can help scientists to understand how aerosols affect the Earth's environment. Scientists use satellites to map where there are large amounts of aerosol on a given day or for a couple of days [4]. Scientists use measurements from the Moderate Resolution Imaging Spectroradiometer (MODIS), aboard NASA's Terra and Aqua satellites, to map the quantities of aerosols in the air over most of the globe [4]. Because aerosols reflect visible and near-infrared light back to space, scientists can use satellites to make maps of where there are high concentrations [4].

Satellite image time series are sequences of satellite images that record a given area at consecutive moments [5]. Since the input dataset comprises continuous snapshots of AOT as a sequence every 8 days, the aims of this study were to develop a model which is capable of learning patterns and relationships and to extract features from time series of satellite images in order to predict AOT for the next 8 days.

The advantage of DL over linear mathematical models and some statistical models that is able to capture nonlinearity based on ANNs; however, to achieve this, the DL model needs a lot of data to learn patterns and identify the relationships between them. Thus, DL modeling is possible because of Big Data. Many open-source datasets are available, e.g., satellite imagery from NEO. Typically, snapshots are taken daily, every 8 days, or monthly. Near-infrared and visible light emitted from the Aqua and Terra satellites is reflected back to space due to the presence of aerosols, making it possible to map global aerosol concentrations. These satellites monitor AOT in order to determine the quantity of aerosols that prevent the transmission of light through scattering or absorption, thereby affecting climate change.

In the present study, satellite-retrieved AOT was used as a dataset MODAL2_E_AER_OD that provides snapshots from 2000 to the present [4]. The dataset, as input data for the training of DL models, covers the period from 18 February 2000 to 14 September 2021, with a temporal resolution of 8 days, providing a total of 993 snapshots in PNG format with 3600×1800 resolution. Measurements of AOT concentration in the snapshots are based on the color scale in a linear range, where yellow indicates $AOT < 0.1$ (clear sky) and brown indicates $AOT = 1$ (very hazy).

Image pre-processing is done by resizing to 288×144 pixels, converting to JPG, normalizing by dividing by 255 and saving the database as a NumPy array. A new function to shift the frames for predictions in the next step was built, whereby x represents frames from 0 to $(n - 1)$ and y frames from 1 to n ; n is the number of input and output time-series sequences. During training, two types of predictions were used: first, a sequence of 10 frames was used for predictions when $n = 10$, and second, this sequence was shifted by one frame, i.e., (1–11) for predictions of the following frame (11) [5].

The train/test split is a technique for training and evaluating the performance of a ML model. The input dataset (MODAL2_E_AER_OD) must be split into training and testing subsets in order to prevent overfitting and poor predictions with new data. The training

subset is a set of data used to fit the model and to learn from it. The test subset is not used to train the model, but rather, to compare predicted values with expected ones in order to evaluate the fit of ML model.

A train/test split is not appropriate when the dataset is small. However, a dataset of almost 1000 images was suitable for the present task. For the used database size, 70% of the dataset (MODAL2_E_AER_OD) was used for training and 30% was used for testing, as well as 80%/20%, or 90%/10%. The reason for such splits lies in the fact that training subset must have enough data to learn and fit the model. Further, the snapshots in the training and testing subsets were selected sequentially, i.e., sorted by data. This was done in this way because the model needs to learn from sequences in time series and predict the next one sequence. After performing all three split types, the evaluation metric showed that a split of 80%/20% yielded had the best result.

For research purposes, two DL methodologies were implemented to develop two models, ConvLSTM and CNN-LSTM, with sequence-to-sequence and sequence-to-one prediction types. Subsequently, an intercomparison was carried out. The input for the CNN was a 4D tensor (comprising samples, rows/height, cols/width, channels). The output was a feature map that extracts spatial information from images with shape, i.e., samples, feature map heights, feature map widths and feature map channels. The number of samples (frames) was 10 in the first case (sequence-to-sequence) and 1 in the second (sequence-to-one).

LSTM is a type of Recurrent Neural Networks which is capable of learning the relationships between time-series elements, i.e., sequential data. ConvLSTM2D is also a recurrent layer with internal matrix multiplications that are exchanged with convolution operations; thus, the input and output dimensions are the same [16]. The input to LSTM layers is a 3D tensor with shape (samples, time steps, features), and as such, in contrast to the ConvLSTM2D layer, it is not able to capture sequential images. The input to the ConvLSTM2D layers is a set of images over time as a 5D tensor comprising samples, time steps, rows, cols and channels.

The output to LSTM layers is a 3D tensor comprising samples, time steps and features when the `return_sequences` attribute is set as true. It is also possible that the output of the LSTM layers is a 2D tensor with samples and features when the `return_sequences` attribute is set as false. The output of the ConvLSTM2D layers is a 5D tensor with samples, time steps, rows, cols and filters if the `return_sequences` attribute is set as true. Otherwise, if that attribute is set as False, the output is a 4D tensor with samples, rows, cols and filters.

3. Deep Learning Models

The ConvLSTM and CNN-LSTM models for global spatio-temporal image prediction were developed based on two methodologies in Keras, an open-source software library that provides a Python interface for ANN. Satellite remote sensing images usually comprise datasets of time series images. Spatio-temporal image prediction with the DL model can be done using the two aforementioned methodologies. In order to create a DL model in Keras, five steps are required: define, compile, fit and evaluate the model, and make predictions. For both methodologies, and training and testing subsets were the same. DL model optimization requires a lot of trial and error during the training process, because for regression, there is no analytical solution, just a numerical solution.

In order to achieve better model performance, a manual search and testing of the construction and learning parameters (like hyperparameters) were done according to the author's previous experience. Hyperparameters are parameters whose values control the learning process and determine the values of the model parameters. In the ConvLSTM methodology, the optimal number of convolutional layers, the number of filters in those layers and the optimal activation functions were tested. Different values for the dropout layer intensity were also used. In the CNN-LSTM methodology, in addition to the steps mentioned above, the number of LSTM units was also examined. Different training optimizers were tested for all models, e.g., learning rates, features to monitor learning, epoch patience and `min_delta`, with the `ReduceLROnPlateau` and `EarlyStopping` methods.

In the interests of comparing the developed models as objectively as possible, the same parameters were defined for the ReduceLROnPlateau and EarlyStopping regulatory methods, and the learning rates were the same at the beginning of the process. Training the ConvLSTM model under the stated conditions took significantly less time than it did for the CNN-LSTM model. For these reasons, the ConvLSTM model regularly tended toward overfitting, and as such, the learning rate was reduced from 0.001 to 0.0005.

The main module of the ConvLSTM methodology is the ConvLSTM2D layer in Keras, which is like an LSTM layer except that the input and recurrent transformations are convolutional [17]. The number of implemented ConvLSTM2D layers depends on the type of problem that is being solved; in our case, after testing several options, the optimal number was found to be 3. The last layer for the ConvLSTM sequence-to-one prediction was the fourth ConvLSTM2D layer, since the Conv2D layer only works with one image, i.e., output is only one image, while the last layer for ConvLSTM sequence-to-sequence predictions is a Conv3D layer [18], since output is sequence of 10 images. Thus, the ConvLSTM model consisted of 3 x ConvLSTM2D layers and the final Conv3D layer as output to produce a tensor containing rows, cols and channel. For further model optimization, the ConvLSTM2D layer was followed by Dropout (randomly selected neurons were ignored during training) and BatchNormalization, which standardizes the inputs to a layer for each mini-batch and reduces the number of training epochs. EarlyStopping was implemented in the fit() method. EarlyStopping is a type of regularization which is used to avoid overfitting, resulting in a reduced learning rate when a metric has stopped improving. The learning rate controls how much to update the weights at the end of each batch.

A grid search for hyperparameters was attempted, but better results were achieved with manual tune testing. The optimal solution for the ConvLSTM developed model was achieved with the following settings: batch size = 1, epochs = 20, activation functions 'sigmoid' and 'relu', and optimizer 'adam' with learning_rate = 0.0005. The activation function controls the non-linearity of individual neurons and when to activate them. The rectifier activation function (ReLU) is commonly used in DL due to its good gradient propagation and efficient computation [19]. The batch size is the number of training samples used by the training dataset to update a model's parameters. An epoch consists of one or more batches and represents the number of complete passes through the entire training dataset. Figure 1 shows the results of the plot_model function in Keras and depicts the ConvLSTM developed model for sequence-to-sequence predictions.

The second developed model for global spatio-temporal image prediction applied the CNN-LSTM methodology, with an input layer that received a sequence of 10 pre-processed images from the training subset and an output comprising 10 sequences, shifted by one sequence. Since the Conv2D layer can only work with one image, a TimeDistributed layer was added to the next four layers, which made it possible to work with a series of input data. The first layer was Conv2D with 33 filters for better model performance. BatchNormalization was used in the second and fourth layers to prevent the so-called explosion of the loss function and to achieve normalization during the learning process by preventing excessive growth of certain weights in relation to the others in each layer in the neural network. The third layer was Conv2D with three filters; this was used to return the format to three RGB channels. The fifth reshape layer transformed the 2D image format to 1D in order to prepare for the next LSTM memory layer. Dropout was used in the sixth and eight layers to prevent overfitting (good processing of known data, but poor processing of new data). The ninth layer determined the final number of pixels of the output predicted image, which was transformed into the appropriate format in the tenth layer. The eleventh layer was convolutional (Conv3D), i.e., a standard image processing layer adapted to the relevant output data type.

Figures 1 and 2 present plots for sequence-to-sequence prediction. Plots for the second type of prediction, i.e., sequence-to-one prediction, are almost the same as for the first type, with the differences being that the last layers are adapted to the output of a single image and the option not to return_sequences is used at the last recurrent layer.

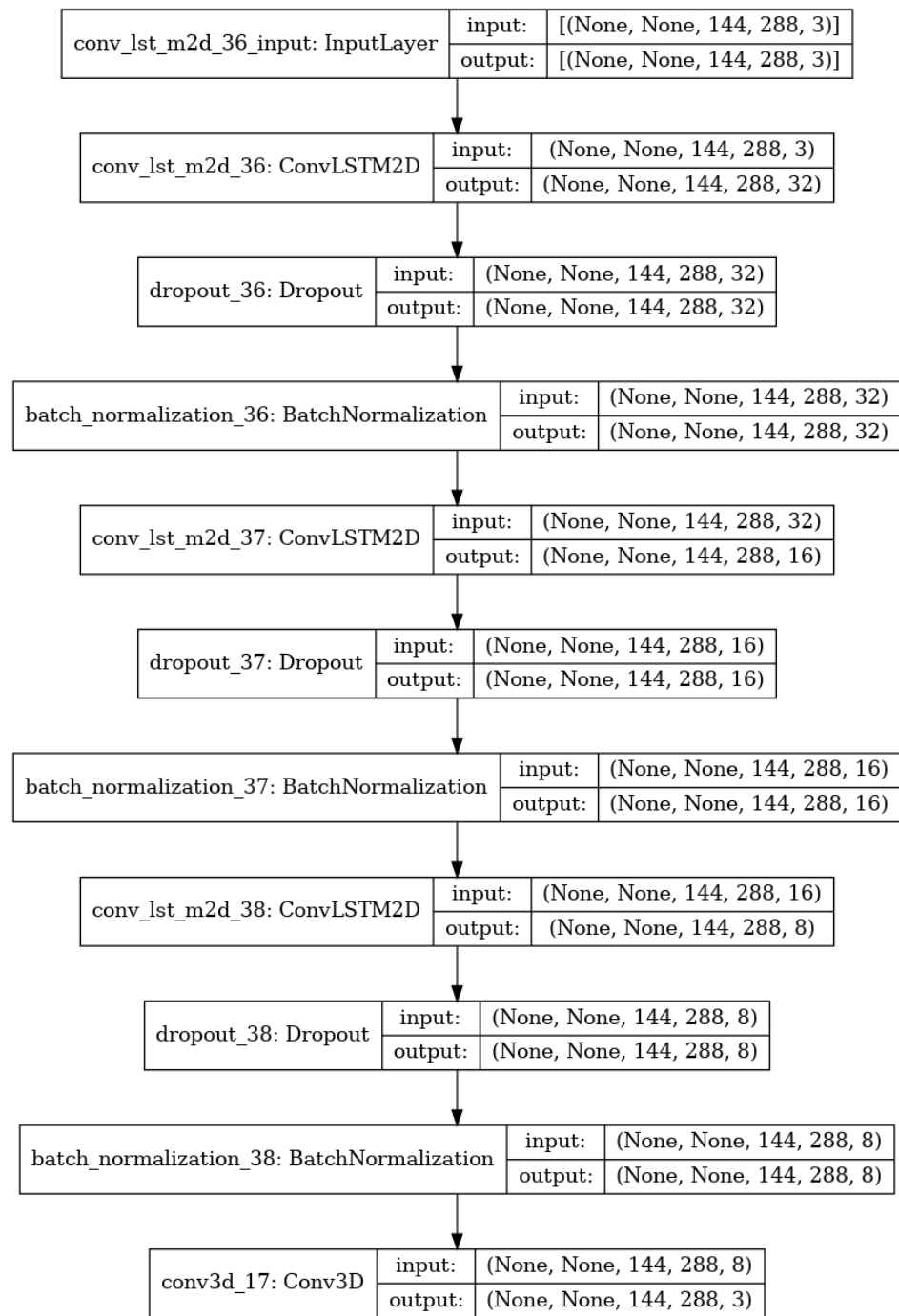


Figure 1. Plot of ConvLSTM developed model for sequence-to-sequence predictions.

The best result for the CNN-LSTM model was achieved using the following settings: batch size = 1, epochs = 20, activation function ‘relu’ (good for vanishing gradient problem) and optimizer ‘adam’ (learning_rate = 0.001). Figure 2 shows the result of the plot_model function in Keras and illustrates the developed CNN-LSTM model for sequence-to-sequence predictions.

In DL, to create a model which is capable of learning, training is required to capture unknown relationships and patterns among data. After finishing the training process, the ConvLSTM and CNN-LSTM models were used to predict the first image from the test dataset, and subsequently, that image was compared with the original first image from the same dataset. Given that the input dataset consisted of global AOT snapshots with an

8 day temporal resolution, it made sense to forecast global AOT for the next eight days (which is the usual duration of the cycle of of aerosol transportation in the atmosphere).

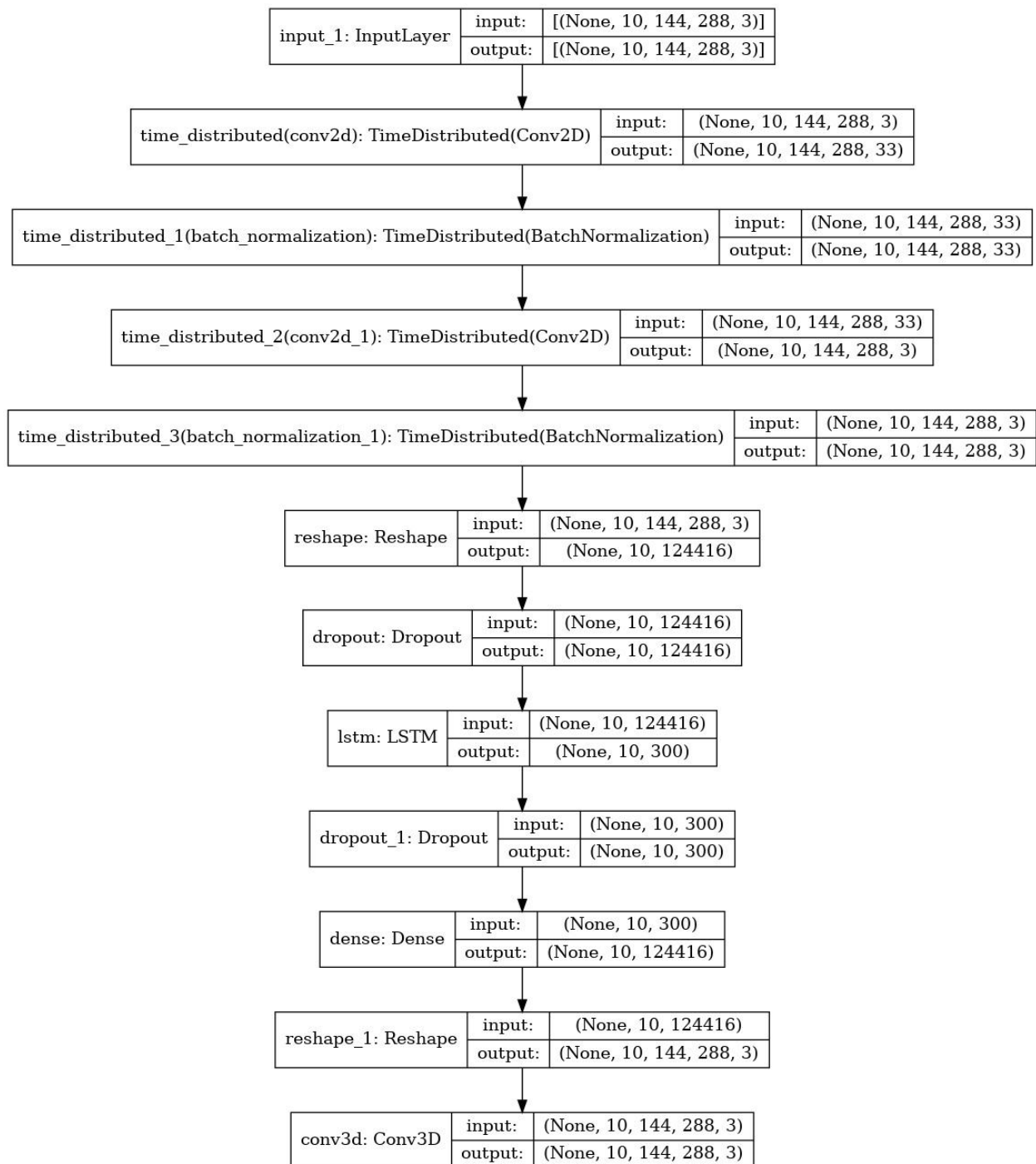


Figure 2. Plot of CNN-LSTM developed model for sequence-to-sequence predictions.

4. Results and Discussion

The ML model learns through an error function, i.e., loss function, whereby the weights can be updated to reduce the loss on the next evaluation as a part of the optimization. Ideally, the prediction will be the ground truth. MSE is the most commonly used loss function for regression, and as such, was selected as the loss function in both of the developed DL models.

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

where n is the number of data points, y_i is the vector of the observed values of the variable being predicted, and \hat{y}_i is the predicted values. MSE is always positive, regardless of the sign of the predicted and actual values. The MSE range is from zero to infinity, where zero means that there is no error (predicted value = actual value).

Our evaluation of the ML model was based on several evaluation metrics showing the amount of deviation from the actual values. Root Mean Squared Error (RMSE) is a metric for evaluating regression models as the square root of MSE. RMSE was used as a metric for both models.

In addition, for better DL model evaluation and comparisons with other studies, cosine similarity (CS) was used. CS is a measure of similarity between two paired sets of numbers. This metric is interesting for image recommendation systems based on comparisons of two paired images [20]. The results of such comparisons are given in the form of values between -1 and 1 , whereby values closer to 1 are obtained for the most similar strings. This metric is interesting for use in regression models for training because, together with the loss function, it can be used to estimate the moment when the model starts to overfit, similar to the accuracy metric for classification.

$$CS = \cos(\alpha) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (2)$$

where x_i and y_i are components of the X and Y vectors, respectively (X represents a 1D vector of the image from the training dataset and Y represents a 1D vector of the predicted image of the developed model) and α is the angle between vector X and vector Y. If a two-dimensional image is converted into a one-dimensional array, a vector with the same number of elements as there are pixels in the image will be obtained. Parameter α represents the angle between the vectors obtained in the described manner, i.e., from the images of the training or testing subset and from the images predicted by the model. The more similar the pixel values of the images, the more similar the sequences and the smaller the angle between the two vectors when the cosine is closer to 1 .

Before the DL models were trained, we examined the validation set in the database in order to better understand the results obtained by modeling. First, the average difference of randomly selected images in a given set was calculated with 1000 repetitions. This resulted in an RMSE value of 0.4613 with a standard deviation (STD) of 0.042 and a CS value of 0.7030 with STD of 0.050 . STD is a statistic that measures the dispersion of a dataset relative to its mean; it is calculated as the square root of the variance:

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (3)$$

where $\{x_1, x_2, \dots, x_n\}$ are the observed values of the sample items, \bar{x} is the mean value of these observations, and n is the size of the sample.

This test shows the average difference between images, regardless of where they are in the dataset. As the images were arranged in a time-spaced sequence, the next test was to determine the average difference between each two adjacent images, which would represent the average change at a given time step. This resulted in an RMSE of 0.4219 with STD of 0.0147 and a CS of 0.7608 with STD of 0.0304 .

DL model training was performed with the same input data for all the models, whilst for the sequence-to-one type of prediction, the last image was extracted from the standard output sequence, as previously explained. For the developed models, the EarlyStopping and ReduceLROnPlateau regulation methods were defined with the same parameters in order to train the models under the same conditions, with the difference being that for models with ConvLSTM2D layers, the learning rate was reduced from 0.001 to 0.0005 due to the risk of fast learning resulting in to overfitting. The monitor for the aforementioned methods was CS for the validation dataset with a patience setting of five epochs and `min_delta = 0.0001` for EarlyStopping and a patience setting of two epochs with an order

of magnitude correction that was the same as that used for ReduceLROnPlateau. The dynamics of the learning process are depicted in Figure 3.

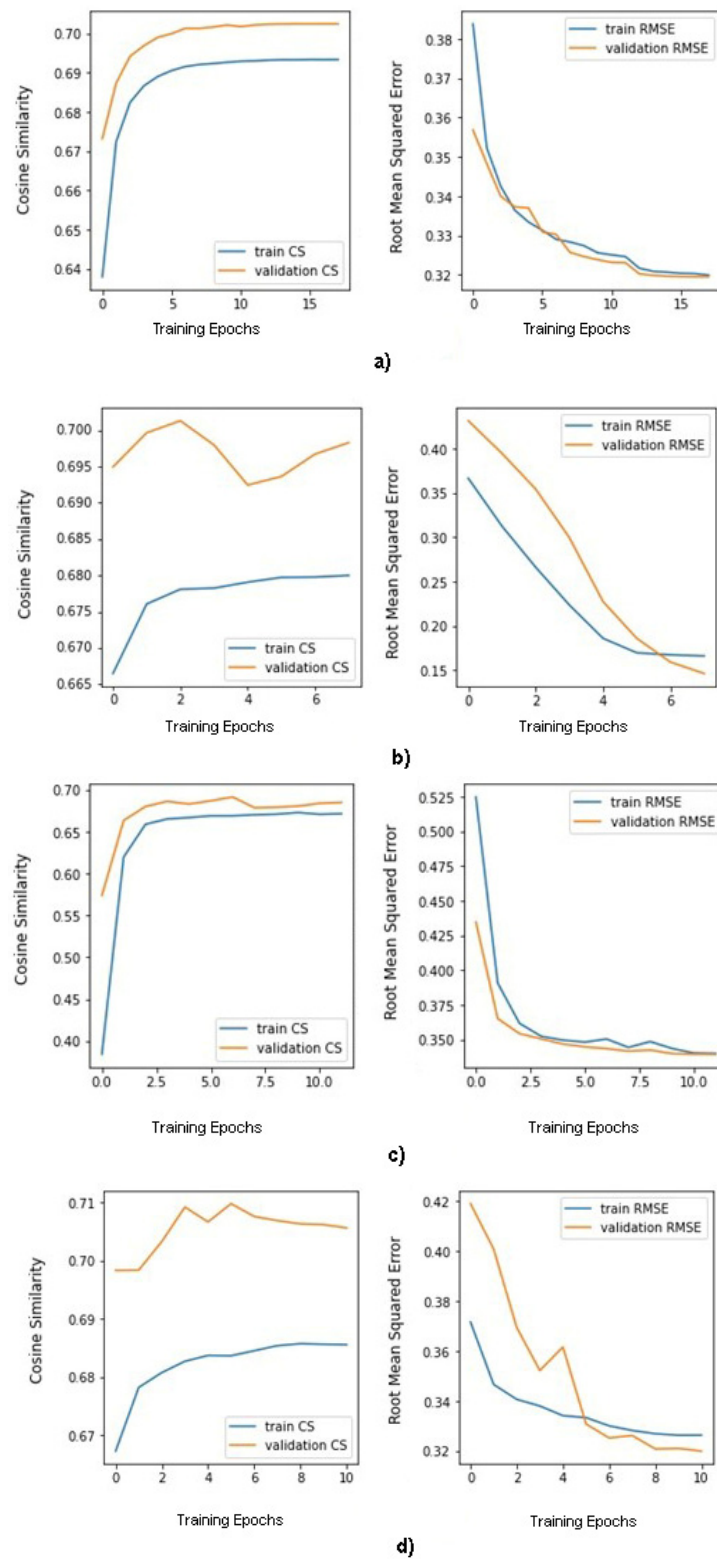


Figure 3. The values of the applied metrics during the learning process, displayed graphically: (a) CNN-LSTM sequence-to-sequence; (b) ConvLSTM sequence-to-sequence; (c) CNN-LSTM sequence-to-one; (d) ConvLSTM sequence-to-one.

Figure 3 reveals that the CNN-LSTM models showed steady and more uniform learning flows across epochs compared to the ConvLSTM models. This behavior depends on the construction of the model and the applied regulation techniques. However, it does not affect the results of various metrics when predicting the model, because in our case, the final characteristics of the model depended only on the weights that had been updated in the last epoch. The statistical characteristics of the final, trained models are shown in Table 1.

Table 1. Comparison of the prediction errors of the four developed DL models using the validation dataset.

Developed DL Model	Prediction Error (RMSE) for the First Set of 10 Frames from the Validation Dataset										Prediction Statistics			
	1	2	3	4	5	6	7	8	9	10	RMSE	STD	CS	STD
CNN-LSTM seq.-to-seq.	0.316	0.306	0.311	0.322	0.306	0.326	0.315	0.316	0.316	0.344	0.3280	0.0100	0.8422	0.0096
ConvLSTM seq.-to-seq.	0.122	0.103	0.099	0.097	0.098	0.102	0.101	0.104	0.105	0.342	0.3400	0.0064	0.8258	0.0119
CNN-LSTM seq.-to-one	/	/	/	/	/	/	/	/	/	0.359	0.3393	0.0096	0.8259	0.0150
ConvLSTM seq.-to-one	/	/	/	/	/	/	/	/	/	0.328	0.3199	0.0085	0.8470	0.0099

Figure 3 shows that the regulation methods responded in a timely manner due to fact that the developed models were slowed down several times and stopped at the appropriate time during training. In order check the performance and compare the developed models in an objective way, a validation dataset was used, i.e., a dataset with which the models had not been previously trained. To calculate the error and STD for the sequence-to-sequence types, only the last image from the prediction sequence was used, such that the results could be compared with those from the sequence-to-one types. A comparison of the prediction errors for the four developed models using the validation dataset is shown as Table 1.

The left side of Table 1 shows the prediction errors (RMSE) for the first set of 10 frames from the validation dataset. The first nine images of the sequence-to-sequence prediction had smaller errors than the tenth image; this could be attributed to data leakage. With this in mind, in further statistical testing, the result of only the tenth image from the prediction sequence was taken. The tenth image in the output sequence never appeared in the model input. For sequence-to-one type predictions, only the tenth image was predicted.

From the statistics shown in Table 1, it can be concluded that the all developed models have similar characteristics, although the ConvLSTM model with the sequence-to-one prediction showed the smallest RMSE and the best CS fit. The sequence-to-sequence prediction showed better stability in both cases compared to the sequence-to-one predictions. The statistics of all models showed lower RMSE, higher CS, and better stability for both metrics than the average random and time step difference of the validation set testing results.

During testing of the prediction speed of the trained models, hardware ablation was performed, as shown in Table 2. Testing was done on a cloud server separately using the server GPU and CPU, described as hardware 1 and 2, respectively in Table 2. Hardware 3 comprised a regular business laptop. The testing was done with a test subset, and the results are presented as mean values.

The results in Table 2 show that the fastest execution of a single prediction was achieved with the CNN-LSTM sequence-to-sequence model running on the server hardware and the sequence-to-one prediction running on the laptop. It is interesting that even though ConvLSTM models make slower prediction, they require as little as half of the space of CNN-LSTM models. The impact of changing the hardware environment on the prediction errors was checked. It was found that there were no deviations in the results up four decimal places for RMSE and CS. As such, it makes no sense to display the results here.

Table 2. Inference speed of one prediction for all developed models on different hardware.

Model	Model Size [GB]	Hardware Number	Hardware Processor	Hardware RAM [GB]	Inference Period [ms]
CNN-LSTM seq.-to-seq.	2.102	1	GPU NVIDIA® TESLA® P100	13+16	29.77
		2	4 × CPU Intel(R) Xeon(R) CPU @ 2.20 GHz	16	105.18
		3	Intel(R) Core(TM) i3-6006U CPU @ 2.00 GHz 2.00 GHz	12	351.54
ConvLSTM seq.-to-seq.	0.968	1	GPU NVIDIA® TESLA® P100	13+16	87.99
		2	4 × CPU Intel(R) Xeon(R) CPU @ 2.20 GHz	16	709.32
		3	Intel(R) Core(TM) i3-6006U CPU @ 2.00 GHz 2.00 GHz	12	1916.66
CNN-LSTM seq.-to-one	2.108	1	GPU NVIDIA® TESLA® P100	13+16	30.04
		2	4 × CPU Intel(R) Xeon(R) CPU @ 2.20 GHz	16	325.58
		3	Intel(R) Core(TM) i3-6006U CPU @ 2.00 GHz 2.00 GHz	12	246.46
ConvLSTM seq.-to-one	0.962	1	GPU NVIDIA® TESLA® P100	13+16	47.89
		2	4 × CPU Intel(R) Xeon(R) CPU @ 2.20 GHz	16	685.97
		3	Intel(R) Core(TM) i3-6006U CPU @ 2.00 GHz 2.00 GHz	12	1790.71

By comparing the obtained results with the results from [5], where RMSE = 0.81 for the CNN architecture and RMSE = 0.68 for the LSTM architecture, the developed models showed better performance with the dataset from the same MODIS satellite. Another comparison was made with similar studies (DNN and hourly AOT); their RMSE was 0.112, which indicates better accuracy, as well as overall superior results compared to those obtained using an SVM, random forest and physical model [21].

The predicted data in the form of images are displayed in Figure 4. More precisely, Figure 4 shows the predicted images of all of the developed models for the same first image of the validation set where the original image is the central one under e). Upon comparing the predicted images visually, it can easily be seen that the ConvLSTM sequence-to-one prediction model yielded the best results.

The color linear scale is the AOT concentration value; this could be converted into exact numbers. Dark brown pixels show high aerosol concentrations, yellowish brown pixels are for lower concentrations, and light-yellow pixels indicate little or no aerosols. Black represents cases in which the sensor could not make a measurement. An $AOT \leq 0.1$ means a clear sky with relatively few aerosols and maximum visibility, whilst a value near 1 indicates high aerosol loads.

As shown in the color scale, very high AOT concentrations (≈ 1) were observed in the south of the Sahara Desert and in the Arabian Desert, which was attributed to sand.

As a practical example of an application of the ConvLSTM models, a forecast of global AOT for the next eight days was performed. The predicted image was processed and filtered in order to locate spots with the highest aerosol concentrations on a global map; see Figure 5.

In order to make predictions using images with higher resolution, more powerful hardware will be needed. In addition, the developed models could be trained and used in specific regions rather than globally, i.e., spatial downscaling [19]. The accuracy of AOT depends on factors such as clouds and snow, i.e., bright surfaces which may lead to missing data. Additionally, the CNN requires full images for training, feature extraction and predictions [19]. However, since aerosol measurements can be used as tracers to study how the Earth's atmosphere moves, the developed DL models can give global insights, i.e., regarding global atmospheric circulation. Remote sensing of AOT will be very useful for the validation of climate models [22].

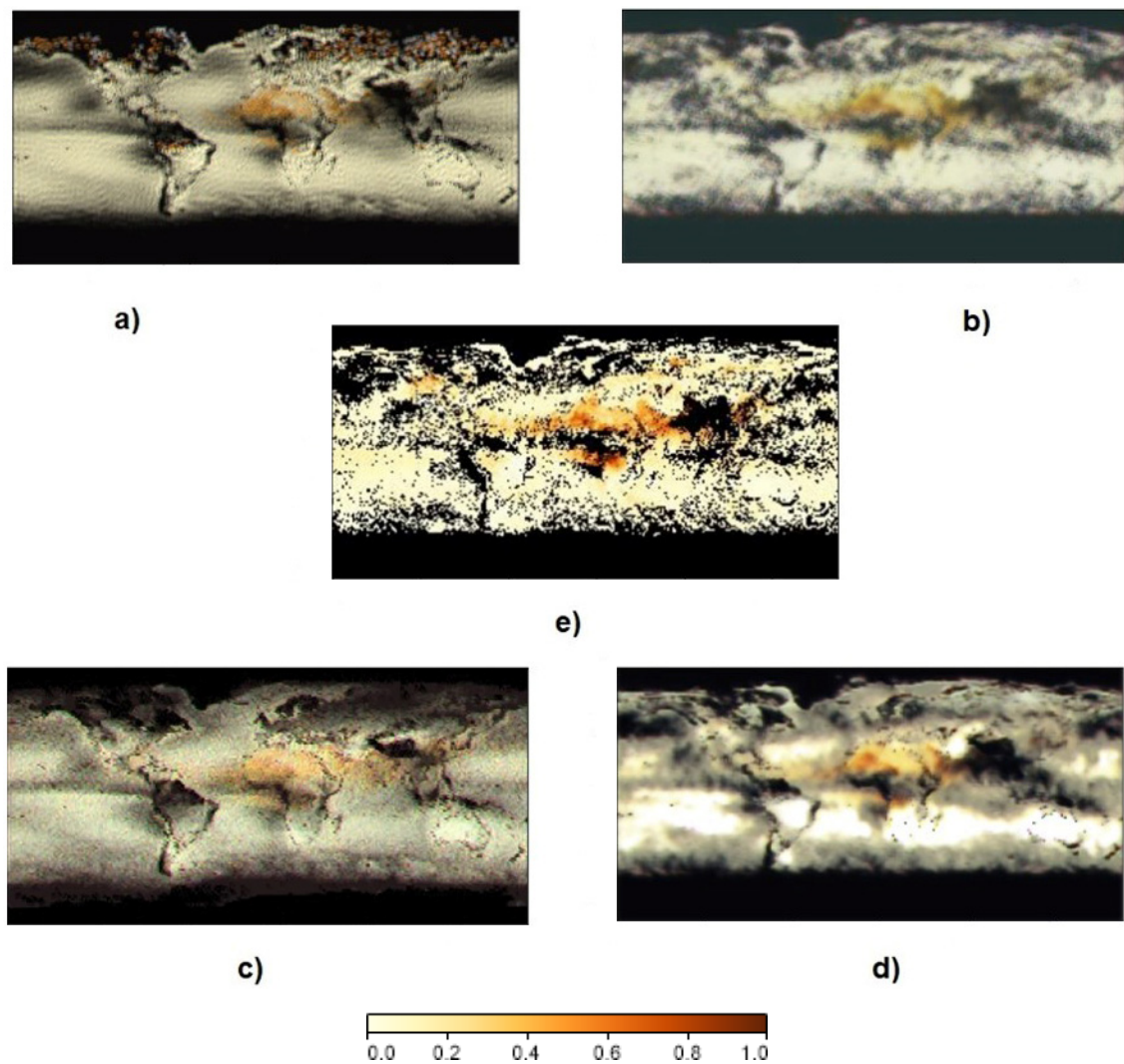


Figure 4. Prediction of the first image from the validation set using the following developed DL models: (a) CNN-LSTM sequence-to-sequence; (b) ConvLSTM sequence-to-sequence; (c) CNN-LSTM sequence-to-one; (d) ConvLSTM sequence-to-one; (e) original output of the first image; Color linear scale of AOT concentration values ranging from 0.0 to 1.0.

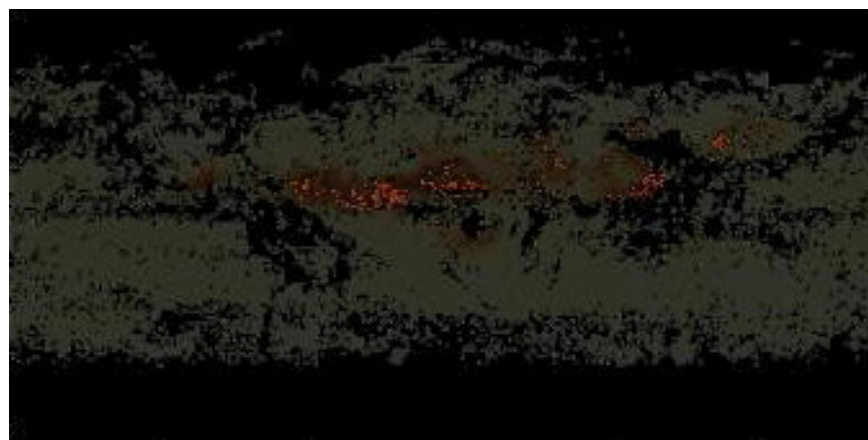


Figure 5. Predicted image on/for the eighth day with spots showing the highest AOT on a global map.

Particulate matter (PM), especially fine particulate matter, i.e., $PM_{2.5}$ (diameter $\leq 2.5 \mu m$), is one of the most dangerous forms of air pollution, with a negative influence on air quality

and human health. In [23], the authors describe the empirical relationship between $PM_{2.5}$ and AOT derived from the MODIS dataset. They found that a linear regression equation can be used for ground-level $PM_{2.5}$ estimations with the AOT results obtained from the developed models.

5. Conclusions and Future Research

The conventional mathematical models for atmospheric predictions, which describe physical phenomena based on the presence of chemicals, have not changed much lately. However, new possibilities appeared with ML and modeling using DL with Big Data to capture non-linear and unknown relationships and patterns. The basis of ML is mathematics, applying differential calculus, like the gradient descent algorithm, in ANNs. The authors developed two DL models based on two methodologies, i.e., ConvLSTM and CNN-LSTM, for two types of predictions, i.e., sequence-to-sequence and sequence-to-one, as a new approach for atmospheric forecasts in order to better understand and improve global climate change resulting from environmental pollution.

The interaction of aerosols with clouds is one of the largest sources of uncertainty in climate modeling. Satellite remote sensing for observations of the global atmosphere makes it possible to monitor the long-range transport of pollutants, albeit with some limitations. The advantage of satellite-based remote sensing is that it provides an aerial view. Determining the altitude of aerosol layers in the atmosphere requires cross verification with ground survey data. According to the authors of [24], it is difficult to detect the impact of AERONET when combined with MODIS in regional and global verification statistics due to the much greater spatial coverage of MODIS. In [25], the authors presented a new meteorological photo classification system based on a multichannel CNN and improved Frame Difference Method (FDM). The authors intend to carry out future research on the classification of clouds and AOT pixels, and then to predict AOT. Furthermore, a deep transfer learning method could be used for classifications. Transfer learning involves applying the knowledge gained while solving one problem to a different but related problem. In [26], the authors reported experimental results with above 90% accuracy. In that study on facial diagnoses, a CNN was deemed to be the most appropriate deep transfer learning method for cases with small datasets.

During our evaluation of the developed DL models, it was determined that all models met the initial criterion of predicting the next image of AOT over the next 8 day based on the last snapshot in the input database. The initial criterion was that the metrics of the predicted images should be better than those obtained by examining the input image database. In this way, the average differences between randomly selected images and between two adjacent images were examined, given that the images were arranged temporally in a sequence.

The obtained results showed small RMSE values, higher CS values and better stability for both metrics than the average random and time step difference of the validation set testing results. The authors believe that the optimal metric for comparing images is CS in combination with RMSE, since it gives much better insights into the moment when the model begins to overfit.

By comparing the developed models, it was determined that ConvLSTM sequence-to-one had the best prediction results according to the criteria of the smallest RMSE error and the best matching of the predicted image compared to the original image using the CS method. During the testing of the prediction speed and hardware ablation, it was determined that the CNN-LSTM models have advantages but require more space than the ConvLSTM models; the ConvLSTM models occupy less than 1 GB, making them suitable for use on weaker hardware architectures. The CNN-LSTM models showed lower inference periods and lower STD error dissipation with the aforementioned metrics. The inference period was in milliseconds; therefore, all of the developed models had prediction speeds of less than one or two seconds. Based on that, it can be concluded that all models predicted almost immediately, i.e., in real time. Considering that the snapshots were collected over

8 days, the difference in prediction speed was not so important, and as such, we recommend the use of the ConvLSTM model with sequence-to-one type predictions.

The developed DL models can be used on PCs provided that the model is trained with commands compile and fit and saved in the hdf5 format on the edge servers (which require significant computational resources). After that, with the load and predict commands, the model forecasts the next sequence of AOT in milliseconds.

With trajectory forecasts, the developed DL models can provide information about aerosol concentration trends, albeit with some limitations. Further steps should involve the implementation of meteorological parameters in order to improve the global AOT forecast accuracy. To increase the size of the input dataset, it should use daily global AOT instead of 8 days, with goal of improving the predictive performance of the models.

The developed DL models are original and innovative, and the obtained results can be taken as a contribution to the scientific community. Furthermore, the models could also be used for other satellite remote sensing datasets for transfer learning. The reader is invited to contact the authors for the relevant code.

Author Contributions: Conceptualization, D.P.N. and N.S.M.; methodology, D.P.N., D.S.R. and N.S.M.; formal analysis, U.R.R. and D.S.R.; investigation, U.R.R. and I.M.L.; writing—original draft preparation, I.M.L. and U.R.R.; writing—review and editing, D.P.N., U.R.R., D.S.R. and N.S.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia, No. 1002205.

Data Availability Statement: The study did not report any data.

Acknowledgments: The research was funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia, No. 1002205. The authors gratefully acknowledge the NASA Earth Observations (NEO) for their effort in making the data available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nikezić, D.P.; Gršić, Z.J.; Dramlić, D.M.; Dramlić, S.D.; Lončar, B.B.; Dimović, S.D. Modeling air concentration of fly ash in Belgrade, emitted from thermal power plants TNTA and TNTB. *Process Saf. Environ. Prot.* **2017**, *106*, 274–283. [CrossRef]
2. Sakaino, H. Spatio-Temporal Image Pattern Prediction Method Based on a Physical Model With Time-Varying Optical Flow. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 3023–3036. [CrossRef]
3. Radivojevic, D.S.; Mirkov, N.S.; Maletic, S. Human activity recognition based on machine learning classification of smartwatch accelerometer dataset. *FME Trans.* **2021**, *49*, 225–232. [CrossRef]
4. Available online: https://neo.gsfc.nasa.gov/archive/rgb/MODAL2_E_AER_OD/ (accessed on 27 March 2022).
5. Moskolai, W.R.; Abdou, W.; Dipanda, A.; Kolyang. Application of Deep Learning Architectures for Satellite Image Time Series Prediction: A Review. *Remote Sens.* **2021**, *13*, 4822. [CrossRef]
6. Zheng, Q.; Yang, M.; Yang, J.; Zhang, Q.; Zhang, X. Improvement of Generalization Ability of Deep CNN via Implicit Regularization in Two-Stage Training Process. *IEEE Access* **2018**, *6*, 15844–15869. [CrossRef]
7. Zheng, Q.; Zhao, P.; Li, Y.; Wang, H.; Yang, Y. Spectrum interference-based two-level data augmentation method in deep learning for automatic modulation classification. *Neural Comput. Appl.* **2021**, *33*, 7723–7745. [CrossRef]
8. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef]
9. Zhang, Y.; Dai, X.; Tian, Z.; Lei, Y.; Chen, Y.; Patel, P.; Bradley, J.D.; Liu, T.; Yang, X. Liver Motion Tracking in Ultrasound images Using Attention Guided Mask R-CNN with Long-Short-Term-Memory Network. *Prog. Biomed. Opt. Imaging—Proc. SPIE* **2022**, *12038*, 120380O.
10. Riordon, J.; Sovilj, D.; Sanner, S.; Sinton, D.; Young, E.W.K. Deep Learning with Microfluidics for Biotechnology. *Trends Biotechnol.* **2019**, *37*, 310–324. [CrossRef]
11. Yuan, Q.; Shen, H.; Li, T.; Li, Z.; Li, S.; Jiang, Y.; Xu, H.; Tan, W.; Yang, Q.; Wang, J.; et al. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sens. Environ.* **2020**, *241*, 111716. [CrossRef]
12. Lipponen, A.; Reinval, J.; Väisänen, A.; Taskinen, H.; Lähivaara, T.; Sogacheva, L.; Kolmonen, P.; Lehtinen, K.; Arola, A.; Kolehmainen, V. Deep-learning-based post-process correction of the aerosol parameters in the high-resolution Sentinel-3 Level-2 Synergy product. *Atmos. Meas. Tech.* **2022**, *15*, 895–914. [CrossRef]
13. Sharma, E.; Deo, R.C.; Soar, J.; Prasad, R.; Parisi, A.V.; Raj, N. Novel hybrid deep learning model for satellite based PM10 forecasting in the most polluted Australian hotspots. *Atmos. Environ.* **2022**, *279*, 119111. [CrossRef]

14. Agga, A.; Abbou, A.; Labbadi, M.; El Houm, Y. Short-term self consumption PV plant power production forecasts based on hybrid CNN-LSTM, ConvLSTM models. *Renew. Energy* **2021**, *177*, 101–112. [[CrossRef](#)]
15. Daoud, N.; Eltahan, M.; Elhennawi, A. Aerosol optical depth forecast over global dust belt based on LSTM, CNN-LSTM, CONV-LSTM and FFT algorithms. In Proceedings of the EUROCON 2021—19th IEEE International Conference on Smart Technologies, Lviv, Ukraine, 6–8 July 2021; pp. 186–191.
16. Shi, X.; Chen, Z.; Wang, H.; Yeung, D. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 802–810.
17. Available online: https://keras.io/api/layers/recurrent_layers/conv_lstm2d/ (accessed on 9 September 2022).
18. Available online: https://keras.io/api/layers/convolution_layers/convolution3d/ (accessed on 9 September 2022).
19. Li, L.; Franklin, M.; Girguis, M.; Lurmann, F.; Wu, J.; Pavlovic, N.; Breton, C.; Gilliland, F.; Habre, R. Spatiotemporal imputation of MAIAC AOD using deep learning with downscaling. *Remote Sens. Environ.* **2020**, *237*, 111584. [[CrossRef](#)]
20. Sejal, D.; Ganeshsingh, T.; Venugopal, K.R.; Iyengar, S.S.; Patnaik, L.M. Image Recommendation Based on ANOVA Cosine Similarity. *Procedia Comput. Sci.* **2016**, *89*, 562–567. [[CrossRef](#)]
21. Yeom, J.-M.; Jeong, S.; Ha, J.S.; Lee, K.H.; Lee, C.S.; Park, S. Estimation of the Hourly Aerosol Optical Depth From GOCI Geostationary Satellite Data: Deep Neural Network, Machine Learning, and Physical Models. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4103612. [[CrossRef](#)]
22. Glantz, P.; Bourassa, A.; Herber, A.; Iversen, T.; Karlsson, J.; Kirkevåg, A.; Maturilli, M.; Seland, Ø.; Stebel, K.; Struthers, H.; et al. Remote sensing of aerosols in the Arctic for an evaluation of global climate model simulations. *J. Geophys. Res. Atmos.* **2014**, *119*, 8169–8188. [[CrossRef](#)]
23. Xu, Y.; Ho, H.C.; Wong, M.S.; Deng, C.; Shi, Y.; Chan, T.-C.; Knudby, A. Evaluation of machine learning techniques with multiple remote sensing datasets in estimating monthly concentrations of ground-level PM_{2.5}. *Environ. Pollut.* **2018**, *242*, 1417–1426. [[CrossRef](#)]
24. Rubin, J.I.; Reid, J.S.; Hansen, J.A.; Anderson, J.L.; Holben, B.N.; Xian, P.; Westphal, D.L.; Zhang, J. Assimilation of AERONET and MODIS AOT observations using variational and ensemble data assimilation methods and its impact on aerosol forecasting skill. *J. Geophys. Res.* **2017**, *122*, 4967–4992. [[CrossRef](#)]
25. Zhao, M.; Chang, C.H.; Xie, W.; Xie, Z.; Hu, J. Cloud Shape Classification System Based on Multi-Channel CNN and Improved FDM. *IEEE Access* **2020**, *8*, 44111–44124. [[CrossRef](#)]
26. Jin, B.; Cruz, L.; Gonçalves, N. Deep Facial Diagnosis: Deep Transfer Learning From Face Recognition to Facial Diagnosis. *IEEE Access* **2020**, *8*, 123649–123661. [[CrossRef](#)]