


RESEARCH ARTICLE

A deep-learning model to predict thunderstorms within 400 km² South Texas domains

Hamid Kamangir^{1,2}  | Waylon Collins³ | Philippe Tissot² | Scott A. King¹

¹Department of Computing Sciences, Texas A&M University Corpus Christi, Corpus Christi, TX

²Conrad Blucher Institute for Surveying and Science, Texas A&M University Corpus Christi, Corpus Christi, TX

³National Weather Service, Corpus Christi, TX

Correspondence

Hamid Kamangir, Department of Computing Sciences, Texas A&M University-Corpus Christi, TX, USA.
Email: hkamangir@islander.tamucc.edu

Abstract

A deep-learning neural network (DLNN) model was developed to predict thunderstorm occurrence within 400 km² South Texas domains for up to 15 hr (± 2 hr accuracy) in advance. The input features were chosen primarily from numerical weather prediction model output parameters/variables; cloud-to-ground lightning served as the target. The deep-learning technique used was the stacked denoising autoencoder (SDAE) in order to create a higher order representation of the features. Logistic regression was then applied to the SDAE output to train the predictive model. An iterative technique was used to determine the optimal SDAE architecture. The performance of the optimized DLNN classifiers exceeded that of the corresponding shallow neural network models, a classifier *via* a combination of principal component analysis and logistic regression, and operational weather forecasters, based on the same data set.

KEYWORDS

deep learning, numerical weather prediction, stacked denoising autoencoder, thunderstorm prediction

1 | INTRODUCTION

The objective of this research is to use the machine learning (ML) technique known as deep learning (Goodfellow *et al.*, 2016) to predict thunderstorm occurrence (± 2 hr and 400 km² accuracy) up to 15 hr in advance. In particular, a deep-learning neural network (DLNN) was used to post-process a deterministic numerical weather prediction (NWP) model output to train and optimize a binary classifier. The

model performed in a superior manner to a corresponding shallow neural network model developed to predict thunderstorms over a South Texas domain (Figure 1) within the United States (Collins and Tissot, 2015, 2016). The NWP models historically have been used to predict the future state of the atmosphere (Bjerknes, 1904; Kalnay, 2003) and are essential with respect to predictions beyond 3 hr (Wilson *et al.*, 1998).

However, the NWP model structure itself limits the predictability of the future atmospheric state. The NWP model configuration components, such as discretization, truncation and parameterization, introduce errors that grow during model integration (Kalnay, 2003). Bifurcation, wherein nearly identical modelled atmospheric thermodynamic profiles can result in divergent solutions (Elmore *et al.*, 2002), further limits predictability. The chaotic nature of the

[Corrections added on 20 May 2020, after first online publication: The article title has been updated and the word “deionizing” has been changed to “denoising” throughout the text.]

[Corrections added on 20 May 2020, after first online publication: In Abstract, the text “of south Texas (USA)” in the first sentence has been changed to “South Texas”.]

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. Meteorological Applications published by John Wiley & Sons Ltd on behalf of the Royal Meteorological Society.

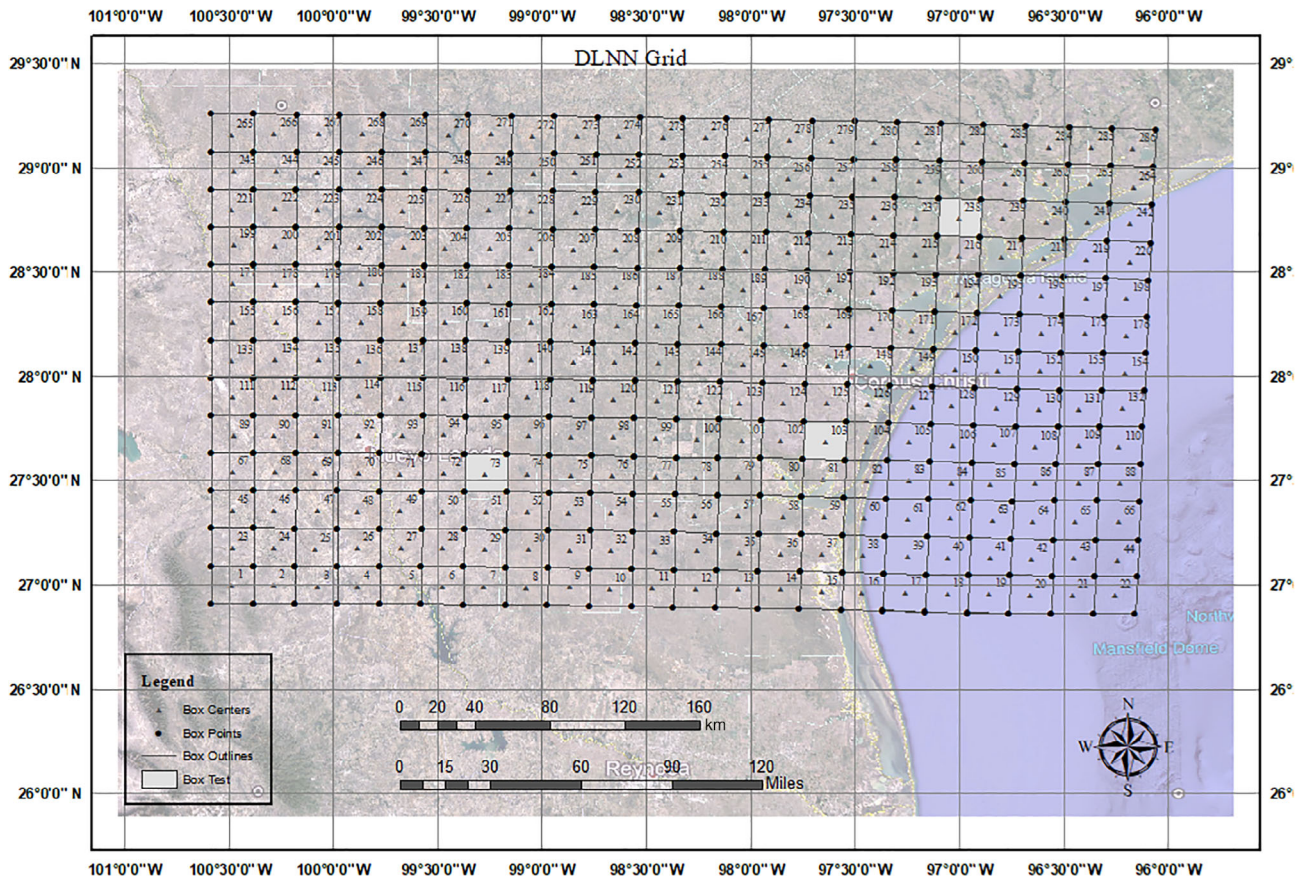


FIGURE 1 Domain grid defined by 286 boxes of area 20×20 km. The deep-learning neural network (DLNN) model used in the present study is trained for all boxes and tested on boxes 73, 103 and 238 (grey filled boxes). Source: Adapted from Collins and Tissot (2015)

atmosphere, with extreme sensitive dependence on the initial conditions (Lorenz, 1963, 1969), also limits the predictability of the future atmospheric state. Further, the fact that thunderstorms develop at micro- α and meso- γ scales (0.5–5 km; Orlandi, 1975) renders the prediction of such more difficult since the intrinsic predictability of the atmosphere is proportional to the spatial scale (Lorenz, 1969).

A well-known strategy used to account for the uncertainty in deterministic NWP output due to model initial conditions and configuration errors is to use an ensemble of NWP runs (Leith, 1974) wherein each ensemble member is a separate NWP run distinguished by a change to the model configuration and/or magnitude of the initial parameters of the atmospheric variables. The idea is to model the sensitivity of the prediction due to the initial conditions and configuration errors. Thus, the forecaster can access the level of prediction uncertainty by assuming a positive correlation between uncertainty and the divergence (or spread) of the ensemble members. With respect to thunderstorm forecasting, prediction probabilities can be developed by post-processing the ensemble. For example, the parameter Calibrated Probability of Thunderstorm is generated by post-processing the ensemble of 21 NWP model runs from the Short-Range Ensemble Forecast (SREF) system developed

by the US National Weather Service (Storm Prediction Center, 2019). Kain *et al.* (2013) and Bouttier and Marchal (2020) demonstrated the utility of this state-of-the-art ensemble approach to thunderstorm prediction.

A different strategy/paradigm is sought in order to account for chaos in atmospheric prediction and errors in NWP output. Rather than post-process a set of unique NWP ensemble runs to predict the future atmospheric state and/or atmospheric phenomena, the NWP output is post-processed from single deterministic runs by training and optimizing a model using the ML, a strategy similar to that used by Collins and Tissot (2015), wherein an NWP model output served as features to train and optimize shallow neural network models to predict thunderstorms. Unlike Collins and Tissot (Collins and Tissot, 2015, 2016), the DLNN is used as the ML technique instead of shallow neural networks. Pathak *et al.* (2018) demonstrated the utility of predicting the future state of chaotic dynamic systems by combining a dynamic model that describes the system with the ML. Further, given that an ensemble of the NWP runs is obviously computationally expensive relative to single deterministic runs, a prediction of thunderstorm occurrence by post-processing secondary output parameters from a single deterministic NWP model integration with an

accuracy/skill comparable with that of a model ensemble system would clearly possess high utility. A motivation for using deep learning to predict thunderstorms is the apparent performance enhancement of deep-learning models relative to the shallow variety with respect to weather and air quality predictions (Hossain *et al.*, 2015; Hernández *et al.*, 2016; Li *et al.*, 2016; Tao *et al.*, 2016). These deep-learning models are based on representation learning (Goodfellow *et al.*, 2016), whereby unsupervised learning occurs across two or more hidden layers as a pre-training step in order to generate an increasingly higher order representation of the input features, with the final hidden layer (with the highest order representation) serving as input into a supervised learning algorithm.

Thanks to the increasing availability of large data sets and affordable computational power, deep-learning algorithms can now model complex nonlinear relationships in the earth sciences (Hernández *et al.*, 2016; Scher, 2018; Gagne II *et al.*, 2019; Lagerquist *et al.*, 2019; Kamangir *et al.*, 2018; Pashaei *et al.*, 2020; Reichstein *et al.*, 2019). The introduction of DLNNs in 2006 (Hinton *et al.*, 2006) has led to large changes in artificial intelligence (AI) and ML in many areas of research. Deep-learning algorithms aim at learning high-level feature representations to solve complex problems, while establishing relationships between problem predictors and predictands. The DLNN has shown the ability to generate higher performance models as compared with the traditional ML. The specific representation learning-based algorithm used in the present study is the stacked denoising autoencoder (SDAE), which involves an unsupervised greedy layer-wise pre-training process followed by the training of a predictive model (Goodfellow *et al.*, 2016).

The paper is organized as follows. Section 2 describes the DLNN model development. Section 3 details the model's implementation, including the presentation of the results. The conclusions are given in Section 4.

2 | METHODOLOGY

This section describes the DLNN model domain, the chosen features (and associated rationale), the target, a detailed explanation of the SDAE method and a description of the reference methods used to compare it with the DLNN model developed in the present study.

2.1 | Model domain

The DLNN model domain is represented as a grid of 13×22 equidistant points, with a grid spacing of 20 km, resulting in 286 box regions of area 400 km^2 (Figure 1). The domain includes the southern portion of the US state of Texas and a

section of the Gulf of Mexico. The latitude–longitude pair of the northeast, southeast, southwest and northwest corners of the domain are 29.17955° N – 96.05539° W , 26.86122° N – 96.15182° W , 26.91000° N – $100.58000^\circ \text{ W}$ and 29.25613° N – $100.58000^\circ \text{ W}$, respectively. This is the same domain as used by Collins and Tissot (2015).

The DLNN models were trained over the entire 286 box domain, while the results were analysed for three of the boxes (see the highlighted boxes in Figure 1). The three boxes were selected to account for the diversity of thunderstorm-formation mechanisms and the climatic frequency of thunderstorm occurrence in order to assess the robustness of the DLNN models developed. Boxes 103 and 238 were representative of conditions within the West Gulf Coastal Plain, with Box 103 selected to assess more specifically model performance along the shoreline of the Gulf of Mexico, and Box 238 to exemplify the portion of the study area with the highest lightning frequency. Box 73 was indicative of the conditions within the Rio Grande Plains, a drier portion of the study area with a low lightning frequency.

2.1.1 | Features

The features chosen originated from the North American Mesoscale Forecast System (NAM) (NWS/EMC, 2019) developed by the National Weather Service, National Centers for Environmental Prediction, Environmental Modeling Center (NWS/NCEP/EMC). The NAM is a placeholder for the operational mesoscale model run on the North American domain. The training and testing sets were derived from the 2004–2012 period of the NAM, which includes the hydrostatic Eta (Rogers *et al.*, 1996) (March 1, 2004–June 19, 2006), Weather Research and Forecasting Non-hydrostatic Mesoscale Model (WRF-NMM) (Janjic *et al.*, 2001) (June 29, 2006–September 30, 2011), and National Oceanic and Atmospheric Administration (NOAA) Environmental Modeling System Non-Hydrostatic Multiscale Model on the Arakawa B-Grid (NEMS-NMMB) (October 1, 2011–December 31, 2012) model output.

All 35 of the specific NAM-based features chosen for the study originated from all NAM features chosen by Collins and Tissot (2015) and were based on meteorological expertise combined with an extensive literature search regarding the myriad of atmospheric moisture, instability and lift factors critical to thunderstorm development. Collins and Tissot (2016) used the same NAM-based features. Several studies add credence to the choice of features used in the present study. The variable selection scheme used by Simon *et al.* (2018) and the variable importance method used by Mecikalski *et al.* (2015) collectively identified convective available potential energy (CAPE), convective inhibition (CIN), convective

TABLE 1 Deep-learning neural network (DLNN) model predictors (features)

Abbreviation	Description (units)	Justification as a thunderstorm predictor
PWAT (1)	Total predictable water (mm)	Atmospheric moisture proxy
MR_{850} (1)	Mixing ratio at 850 hPa ($\text{g}\cdot\text{kg}^{-1}$)	Lower level moisture necessary for convective cell to reach a horizontal scale of ≥ 4 km in order to overcome dissipative effects (Khairoutdinov and Randall, 2006)
RH_{850} (1)	Relative humidity at 850 hPa (%)	When combined with CAPE, it is a predictor of subsequent thunderstorm location independent of synoptic pattern (Ducrocq <i>et al.</i> , 1998)
CAPE (1)	Surface-based convective available potential energy ($\text{J}\cdot\text{kg}^{-1}$)	Instability proxy; the quantity $(2\text{CAPE})^{0.5}$ is the theoretical limit of the thunderstorm updraft velocity (Trier, 2003)
CIN (1)	Convective inhibition ($\text{J}\cdot\text{kg}^{-1}$)	Surface-based convective updraft magnitude must exceed $(\text{CIN})^{0.5}$ for parcels to reach a level of free convection (Trier, 2003)
LI (1)	Lifted index (K)	Atmospheric instability proxy; utility in thunderstorm prediction (Haklander and Van Delden, 2003)
$U_{\text{LEVEL}}, V_{\text{LEVEL}}$ (1)	U, V wind components at surface, 850 hPa [LEVEL = surface, 850 hPa] ($\text{m}\cdot\text{s}^{-1}$)	Strong wind can modulate or preclude surface heterogeneity-induced mesoscale circulations (Dalu <i>et al.</i> , 1996; Wang <i>et al.</i> , 1996)
VV_{LEVEL} (1)	Vertical velocity at 925, 700 and 500 hPa [LEVEL = 925, 700 and 500 hPa] ($\text{Pa}\cdot\text{s}^{-1}$)	Account for mesoscale and synoptic-scale thunderstorm triggering mechanisms (sea breezes, fronts, upper level disturbances) that are resolved by the NAM
$\text{DROPOFF}_{\text{PROXY}}$ (1)	Potential temperature drop-off proxy (K)	Atmospheric instability proxy; highly sensitive to CI (Crook, 1996)
LCL (1)	Lifted condensation level (m)	Proxy for cloud base height; positive correlation between cloud base height and CAPE to convective updraft conversion efficiency (Williams <i>et al.</i> , 2005)
T_{LCL} (1)	Temperature at the LCL (K)	$T_{\text{LCL}} \geq -10^\circ\text{C}$ is essential for the presence of supercooled water in convective cloud essential for lightning via a graupel-ice crystal collisional mechanism (Saunders, 1993)
CP (1)	Convective precipitation ($\text{kg}\cdot\text{m}^{-2}$)	Byproduct of the Betts–Miller–Janjic convective parameterization scheme (Janjic, 1994), when triggered; proxy for when the NAM anticipates existence of subgrid-scale convection
VSHEARS8 (1)	Vertical wind shear: 10 m to 800 hPa layer (10^3 s^{-1})	Combination of horizontal vorticity (associated with ambient 0–2 km vertical shear) and density current (e.g. gust front)-generated horizontal vorticity (associated with a 0–2 km vertical shear of the opposite sign than that of ambient shear) can trigger new convection (Rotunno <i>et al.</i> , 1988)
VSHEAR86 (1)	Vertical wind shear: 800–600 hPa layer (10^3 s^{-1})	Convective updraft must exceed the vertical shear immediately above the boundary layer for successful thunderstorm development (Colquhoun, 1987; Crook, 1996)
$U_{\text{LEVEL}}, V_{\text{LEVEL}}$ (2)	U, V wind at the surface, 900, 800, 700, 600 and 500 hPa levels [LEVEL = surface, 900, 800, 700, 600 and 500] ($\text{m}\cdot\text{s}^{-1}$)	Thunderstorm profile modification owing to veering of the wind (warming) or backing of the wind (cooling); backing (veering) of the wind in the lowest 300 hPa can suppress (enhance) convective development (Findell and Eltahir, 2003)
HI_{LOW} (2)	Humidity index ($^\circ\text{C}$)	Both a constraint on afternoon convection and an atmospheric control on the interaction between soil moisture and convection (Findell and Eltahir, 2003)
$\text{CTP}_{\text{PROXY}}$ (2)	Proxy for convective triggering potential (dimensionless)	Both a constraint on afternoon convection and an atmospheric control on the interaction between soil moisture and convection (Findell and Eltahir, 2003)

TABLE 1 (Continued)

Abbreviation	Description (units)	Justification as a thunderstorm predictor
VSHEARS7 (2)	Vertical wind shear: Surface to 700 hPa layer (10^3 s^{-1})	Strong vertical shear in the lowest 300 hPa can suppress convective development (Findell and Eltahir, 2003)
VSHEAR75 (2)	Vertical wind shear: 700–500 hPa layer (10^3 s^{-1})	Convective updraft must exceed vertical shear immediately above the boundary layer for successful thunderstorm development (Colquhoun, 1987; Crook, 1996)
JD (3)	Julian day (day)	Periodic function providing information to the DLNN regarding thunderstorm occurrence as a function of season
Location (3)	Latitude and longitude	Providing information to the DLNN regarding thunderstorm occurrence as a function of location

Note: Numbers in parentheses to the right of each feature denote the following: 1: North American Mesoscale Forecast System (NAM) predictor variable, as described in Section 2.1.1; 2: NAM initialization variable; and 3: variable other than the NAM variable.

Source: Adapted from Collins and Tisot (2015).

CI or CIN, Convective Inhibition.

precipitation, vertical velocity (at both the 500 and 700 mb pressure levels), and lifted condensation level (LCL) height as relevant thunderstorm predictors, all of which are included as features in the present study. Table 1 depicts all the variables used as input features to the SDAE, as well as a brief discussion of the rationale for their selection. The study combines the 35 NAM-based features with Julian day, latitude and longitude (for a total of 38 features). Owing to the additional three features, the DLNN models were trained to predict thunderstorms as a function of location and season.

2.1.2 | Target

Cloud-to-ground (CG) lightning was used as the proxy for thunderstorm occurrence and obtained from the terrestrial-based National Lightning Data Network (NLDN) (Orville, 2008). A target vector was created that contained the number of CG lightning strikes *per* date/hr/box for the study duration (2004–2012). The target was defined as:

$$t_{d,h} = \begin{cases} 0, L = 0 \\ 1, L \neq 0 \end{cases}$$

where d and h represent the date and hour (UTC), respectively; and L is the quantity of CG lightning strikes *per* hr within a given 400 km^2 box region.

2.2 | Stacked denoising autoencoder (SDAE)

While most ML models work with raw input features, their performance is influenced by the number of features. Sometimes the performance is degraded by increasing the

number of features. This problem is known as the curse of dimensionality (Charte *et al.*, 2018). One solution to this problem is to engineer manually a set of features based on expertise. This technique can be time-consuming and error prone. Automated feature selection methods continue to be developed to reduce the dimension of input space, selecting the best subset of features (Dash and Liu, 1997). However, presently these techniques typically consider the importance of each feature independently before selecting or eliminating them. Feature extraction or construction (Liu and Motoda, 1998) is a good alternative to reduce the dimension of input feature space. Several feature extraction techniques all have the goal of finding a better representation of input features by extracting combinations of the original features. Methods include linear combinations, known as linear dimensionality-reduction techniques such as principal component analysis (PCA) (Jolliffe, 2011) or linear discriminant analysis (Fisher, 1938), and nonlinear combinations or nonlinear dimensionality reduction techniques such as kernel PCA and autoencoders (AEs).

2.2.1 | Autoencoder (AE)

An AE network is a specific type of feed-forward neural network with a symmetric structure that attempts to reconstruct the output to resemble the input as closely as possible. The basic structure of the AE is illustrated in Figure 2, including the encoder function (function f) responsible for mapping the input (x) onto the encoding y and producing the reconstructed features r using the decoder function g (Vincent *et al.*, 2010; Baldi, 2012). Both x and r must have the same dimension. Internally, the AE hidden layer is known as the bottleneck. The dimension of the encoding layer y is selected based on the desired properties of the AE. It can be less than the input dimension known as an undercomplete AE, or higher than the input

dimension known as an overcomplete AE. The simplest AE consists of just one hidden layer, and is defined by two weight matrices and two bias vectors (Charte *et al.*, 2018):

$$y = f(x) = S_1(W_1x + b_1) \quad (1)$$

$$r = g(x) = S_2(W_2x + b_2) \quad (2)$$

where W_1 and W_2 are weight matrices; b is a bias; x is the input data; and S_1 and S_2 denote the activation functions. These activation functions are nonlinear to model the potential non-linearity of the relationships between input and encoded features. Several linear and nonlinear activation functions are in use for ML models (Karlik and Olgac, 2011). Rectified linear units (ReLU) is one popular activation function, but it tends to degrade AE's performance since it always outputs zero for negative encoded data (Karlik and Olgac, 2011). The most frequently used nonlinear activation functions are sigmoid functions, including the logistic function.

Traditionally, AEs have been used as dimensionality-reduction methods or feature-extraction techniques (Vincent *et al.*, 2010; Baldi, 2012). An undercomplete AE can be used to reduce the dimension of the input features by constraining the dimension of the bottleneck (layer y in Figure 2) to have a smaller dimension than input features (layer x in Figure 2). Undercomplete representations force the AEs to learn the most latent features of the input data while minimizing a loss function:

$$\Gamma(w, b; S) = \sum_{x \in S} L(x, g(f(x))) \quad (3)$$

where L is a loss function to minimize the difference between the original input (x) from the set of input S and its reconstruction $g(f(x))$ with (w) the weights and (b) the biases of the neural network AE. The loss function is typically based on the mean squared error (Wang and Bovik, 2009) (L_{MSE} , Equation 4) or cross-entropy (CE) (Bengio *et al.*, 2007) (L_{CE} ; Equation 5). Back-propagation is using to

update weights and biases to minimize the reconstruction error (Hinton and Salakhutdinov, 2006):

$$L_{MSE}(r, x) = \|r - x\|_2^2 \quad (4)$$

$$L_{CE}(r, x) = - \sum_{k=1}^d x_k \log(r_k) + (1 - x_k) \log(1 - r_k) \quad (5)$$

where r is the reconstructed output; x is the input; d is a set of samples; and k is the number of iterations.

If a linear activation function and the MSE loss function are selected, an undercomplete AE will extract variables in the same way variables are mapped by the PCA (Jolliffe, 2011). If, for an undercomplete AE, nonlinear activation functions are selected for the encoder and decoder, the model can nonlinearly extract the most salient features of the data set. Such AE models are also referenced to as auto-associative neural networks (Kramer, 1991). These methods can have overfitting issues and may require different validation methods than the common use of testing sets. For in-depth discussions about the challenges and methods for the validation of nonlinear PCA, see Hsieh (2007) and Scholz (2012). The following sections expand on the concept of auto-associative neural networks to the use of deep learning, including the use of methods to avoid overfitting such as weight decay and addition of noise to the model inputs.

2.2.2 | Regularized AEs

Like many other ML models, AEs are prone to overfitting of the training data set, resulting in poor out-of-sample performance (Srivastava *et al.*, 2014). To avoid the overfitting issue, in addition to limiting the model's capacity by choosing an undercomplete structure, some regularization terms can be added to the loss function to encourage the model to learn other properties of the input data. Regularization can be achieved by adding a penalization

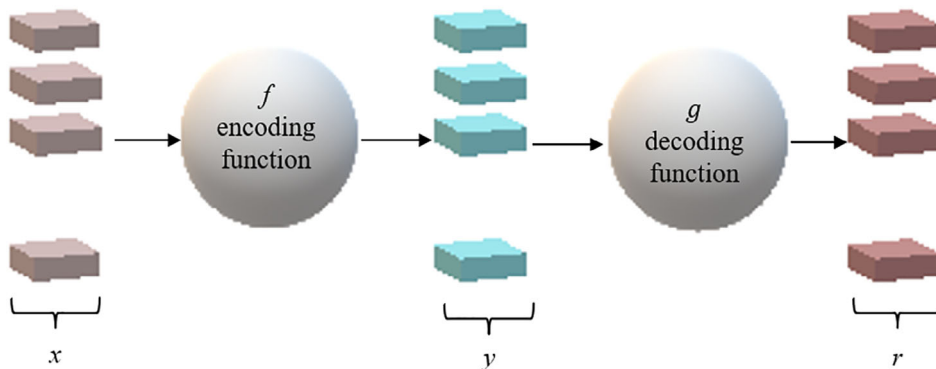


FIGURE 2 Basic structure of an autoencoder (AE), which includes an input x that is mapped onto the encoding y via an encoder, represented as a function f . The encoding is in turn mapped to the reconstruction r by means of a decoder, represented as function g

weight decay or introducing sparsity in the representation resulting in a sparse AE (Charte *et al.*, 2018). For a sparse AE, a set percentage of the units in the hidden layer will be disabled owing to the low values for activation of the encoding layer (Lee *et al.*, 2008; Xu *et al.*, 2015). Weight decay enhances the generalization of training by amplifying smaller weights that produce good reconstruction (Hinton and Salakhutdinov, 2006). In Equation 6, the weight decay term λ is added to the loss function to quantify the magnitude of the decay and to limit the weights' growth. The resulting loss function is expressed as:

$$\Gamma(w, b; S) = \sum_{x \in S} L(x, g(f(x))) + \lambda \sum_i w_i^2 \quad (6)$$

where w_i are all the weights in W ; and S is set of input features. Another strategy to force the AEs to learn the best latent features is denoising AEs (DAEs) (Vincent *et al.*, 2010). A DAE learns to generate robust features from the input by reconstruction from potentially noisy instances. The structure for DAEs is the same as for AEs as well as the parameters, but for DAEs stochastic corruption is added to the input data during the training of the model. Based on the concept behind the denoising technique, the obtained representations and latent features are more robust and informative and in turn more useful for reconstruction.

2.2.3 | Stacked autoencoder (SAE)

As with any neural network, there is the flexibility to construct the AE with several hidden layers or nodes. An SAE is an AE with more than one hidden layer. An SAE takes advantage of all the benefits of any deep network with higher expressive power and computes features based on the greedy layer-wise training method (Hinton and Salakhutdinov, 2006; Bengio *et al.*, 2007). In this training method, the first layer of neurons, which ingests the raw input, is trained to obtain weight and biases that allow for a good reconstruction of the input layer. The output of the first layer is then used by the second layer to obtain its own set of weights and biases with the target to reproduce the output of the first layer. The process is repeated for potential additional layers using the output of each layer as input for the next and computing sets of weights and biases to reproduce the output that previous layer (Bengio *et al.*, 2007). Based on this strategy, the parameters of each layer are trained individually. Unlike supervised learning that tends to train models directly by gradient descent starting from randomly initialized parameters, the greedy layer-wise-based model is using unsupervised learning to pre-train each layer and leads to progressively higher level representations based on the lower level representation output of the previous layer (Bengio *et al.*, 2007).

Hinton and Salakhutdinov (2006) pointed out that:

Gradient descent can be used for fine-tuning the weights in such AE networks, but this works well only if the initial weights are close to a good solution. They describe an effective way of initializing the weights that allows deep AE networks to learn low-dimensional codes (greedy-layer wise approach) that work much better than principal components analysis as a tool to reduce the dimensionality of data.

Regarding randomly initializing the weights, optimizing the weights in nonlinear AEs is difficult because with large initial weights, the AE cannot find local minima, and with small initial weights, there is potential for the gradient vanishing problem. The greedy-layer-wise technique or pre-training approach was the solution to solve the training of the AE proposed by Hinton and Salakhutdinov (2006). There is a global fine-tuning stage to replace stochastic by deterministic, real-valued probabilities by using the backpropagation through the whole AE to update the weights for optimal reconstruction. Each layer in an AE extracts higher order correlation between features in the two layers. For a wide variety of data, the AE can reveal a low-dimensional nonlinear structure of the original data.

2.3 | Comparative methods

Three methods were applied to the same data set used to develop the present study's deep-learning model for the purpose of performance comparisons: (1) shallow neural network; (2) a PCA-based dimension reduction followed by logistic regression; and (3) operational predictions from the National Weather Service. The data set was identical to that used by Collins and Tissot (2016), except for the addition of latitude and longitude as features.

2.3.1 | Shallow neural network

Collins and Tissot (2016) developed a shallow artificial neural network model to predict thunderstorm occurrence within three 400 km² box regions, 9, 12 and 15 hr (± 2 hr) in advance, by post-processing primarily the NWP model output. This study was an adjustment of Collins and Tissot (2015) to increase the available training and testing data by two orders of magnitude. [Correction added on 20 May 2020, after first online publication: The first part of the preceding sentence has been updated for clarity.] The feedforward multilevel perceptron (MLP) topology with one hidden layer was chosen; a neural network with one hidden layer can approximate any

continuous function if there is a sufficient number of hidden layer neurons (Hornik *et al.*, 1989; Hagan *et al.*, 1997). One output neuron was chosen. The transfer functions were log-sigmoid and linear in the hidden and output layers, respectively. The training algorithm used was the second-order scaled conjugate gradient method (Møller, 1993). Binary classifiers were generated by thresholding the MLP continuous output. An iterative method was used to determine the optimum number of hidden layer neurons. For each number of hidden layer neurons (Y) tested from the set $Y = [1-10, 12, 15, 17, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 125, 150, 200]$, 50 iterations of the following were performed: the training set was used to create a receiver operating characteristic (ROC) curve (Bradley, 1997) and the chosen threshold corresponded to that point on the ROC curve where the Peirce skill score (PSS) was maximized, as suggested by Manzato (2007). The threshold was then used to generate binary classifiers which were then evaluated on the testing data set. After 50 iterations, the mean PSS was calculated. The topology corresponding to the maximum mean PSS was considered optimal. This procedure was performed for each prediction period (9, 12 and 15 hr) and box region (73, 103 and 238). Three sets of shallow networks were developed. One version was based on the use of all features. The remaining sets were based on features that survived the filtering-based and nonlinear correlation-based feature selection (CFS) (Hall and Smith, 1999) and the minimum redundancy maximum relevance (mRMR) (Ding and Peng, 2005) feature selection methods. The most skilful binary classifiers from Collins and Tissot (2016) for each prediction period *per* box were compared with the DLNN in the present study.

2.3.2 | Principal component analysis (PCA)

PCA is one of the best-known linear transformation techniques and is broadly used in the environmental sciences and other fields (Jolliffe, 2011). It is mainly used to reduce the dimension of feature space while preserving as much variability as possible (Wold *et al.*, 1987). It transforms the original variables (d -dimensional space) into a smaller dimensional subspace (k) by finding the direction of maximum variance in high-dimensional data (non-correlated variables). The new set of variables based on their variability is ordered so that the first few retained variables capture the most variability in the original variables (Jolliffe *et al.*, 2016).

2.3.3 | Operational forecasts

The National Digital Forecast Database (NDFD) (Glahn and Ruth, 2003) contains an archive of high-resolution

(5 km) grid of weather forecasts generated by operational forecasters within the US National Weather Service. Computer software was used to extract probabilistic thunderstorm forecasts and to convert to binary forecasts (a forecast of thunderstorms regardless of probability was classified as a thunderstorm forecast) which could be compared with the DLNN model.

3 | EXPERIMENT AND RESULTS

3.1 | DLNN model set-up

The DLNN models were developed to predict thunderstorm occurrence within three 400 km² box domains (73, 103 and 238 in Figure 1) for three prediction hours (9, 12 and 15 ± 2 hr). The data for the periods 2004–2006 and 2009–2013 were used to train the model. The data for 2007–2008 were used for testing. For each prediction hour (9, 12 and 15 hr), the DLNN models were trained over all 286 of the 400 km² continuous domains. This approach increases the number of thunderstorm cases ($t_{d,h} = 1$; see Section 2.1.2) and total instances sufficient to justify the use of deep learning. For the 9 hr prediction, there were 663,519 instances in the training sample, 22,139 (about 3.34%) of which were positive target data ($t_{d,h} = 1$). The corresponding values for 12 hr prediction were 646,073 instances with 16,904 (about 2.62%) positive target. For 15 hr, there were 659,802 instances in the training sample, 12,682 (about 1.92%), of which were positive target data ($t_{d,h} = 1$).

The development of the DLNN model began with the determination of the SDAE architecture. As discussed in Section 2.1.1, the number of predictor variables (X) (the input layer) for the SDAE model was 38. The SDAEs with different under- and overcomplete architectures were tested. The output of the SDAE was fed into a logistic classifier consisting of two neurons and resulted in a binary classification, zero for non-lightning and one for lightning. The SDAE was trained based on stochastic gradient descent (SGD) and experiments were repeated 50 times to assess the variability of the process. The trained model was finally evaluated based on the independent test data set. This experiment varied the number of neurons in the hidden layers from one to 100 to determine the optimum number, while also varying the number of hidden layers from two to three. Table 2 depicts the range of SDAE dimension and hyperparameters, including the number of layers, number of neurons for each layer, optimization technique, and so on. After iterating over the range of hidden layers (two to three) and number of hidden layer neurons (one to 100), it was determined that the optimum architecture was undercomplete with two hidden layers

TABLE 2 Range of stacked denoising autoencoder (SDAE) dimension and hyperparameters tested

Model	Hyperparameter	Values
SDAE	Hidden layers	2–3
	Neurons	1–100
	Loss function	Mean squared error (MSE) – cross-entropy (CE)
	Activation function	Sigmoid-Tanh
	Optimization	Stochastic gradient descent (SGD)
	Noise mask	15%; 0; 0
	Regularization parameter λ	0.1–0.001
	Learning rate	0.0001
	Training epochs	500

(30 neurons in the first hidden layer and three neurons in the bottleneck layer) (Figure 3).

To determine the optimum number of neurons, a 95% confidence interval based on the PSS metric and the standard error estimated based on the 50 iterations were used (Collins and Tissot, 2016). Figure 4a depicts the results for the bottleneck layer, which is the most important as it determines the number of latent features and is used as input data for the supervised classifier. The optimum number of hidden layer neurons was selected on the maximum PSS, while also avoiding an overlap in the standard errors with a solution for a smaller number of hidden neurons. A bottleneck layer with three neurons provided the maximum PSS with no overlap in standard error with solutions with one or two hidden neurons. Based on the same strategy, 30 neurons were selected for the first layer. Conceptually, for the purpose of binary classification or prediction, fine-tuning using backpropagation can be applied by tuning the parameters for all layers, and it is common to discard the “decoding” layers of an SAE and link the last hidden layer (bottleneck) to the classifier (Vincent *et al.*, 2010; Sainath *et al.*, 2012; Gehring *et al.*, 2013). The gradients from the classifier classification or prediction error are then back-propagated into the encoding layers (Vincent *et al.*, 2010). As a classifier for fine-tuning the whole network, logistic regression (logistic function) was applied on top of the network. To avoid overfitting in the model, 15% noise was added only for the first layer to force the SDAE model to understand the latent features better. Also, a regularization term (weight decay) was added to decrease further the likelihood of overfitting based on Formula 6. By trial and error, λ was set to 0.001. Using CE resulted in better performance while using a sigmoid activation function, hence the final loss function for the model could be expressed as:

$$\Gamma(w, b; S) = - \sum_{k=1}^d x_k \log(r_k) + (1-x_k) \log(1-r_k) + \lambda \sum_i w_i^2 \quad (7)$$

The ROC curve (Bradley, 1997) is a technique for visualizing the skill of binary classifiers. To determine the model that optimizes performance, an ROC curve was created by adjusting the decision threshold at an iteration of SDAE output range and calculating the probability of detection (POD) and false-alarm rate (F) at each iteration. Based on the best performance for the PSS metric, the threshold is chosen. Figure 4b depicts the ROC curve associated with the development of the SDAE model for the 12 hr thunderstorm prediction in box 238.

3.1.1 | Feature reduction comparison and model performance assessment

Figure 5 compares the importance of the first 10 PCA variables sorted by their ranking. The first 10 variables explain approximately 90% of the variability, but the first three PCA components already provide > 70% of the variability of the original data. Inputs consisting of both the first three and the first 10 PCA components were tested to predict lightning using a logistic regression classifier. The results were very close to each other, indicating that using only a simpler input with three PCA components gives a good representation of the performance of this method. Using three PCA components also allows one to compare more directly with the SDAE methods which also uses three latent features.

As applied here, the SDAE is learning low-dimensional representations of the data through dimensionality reduction controlled by the number of hidden neurons in the bottleneck layer. The information from the bottleneck is then used as an input into the classifier, resulting in improved performance by focusing the model on the most relevant information in the input features (Hinton and Salakhutdinov, 2006). The SDAE can detect repetitive and redundant structures and consolidate them into lower dimensionality latent features, resulting in more distinguishable and informative features. To understand better how dimension reduction may lead to better performance, the SDAE output for three latent features is compared with the results of dimension reduction using the linear PCA technique. The comparisons are presented in Figure 6. The point cloud illustrated in Figure 6a–d shows the result for the PCA model; Figure 6e–h presents the result for the SDAE. Lightning cases are displayed in yellow;

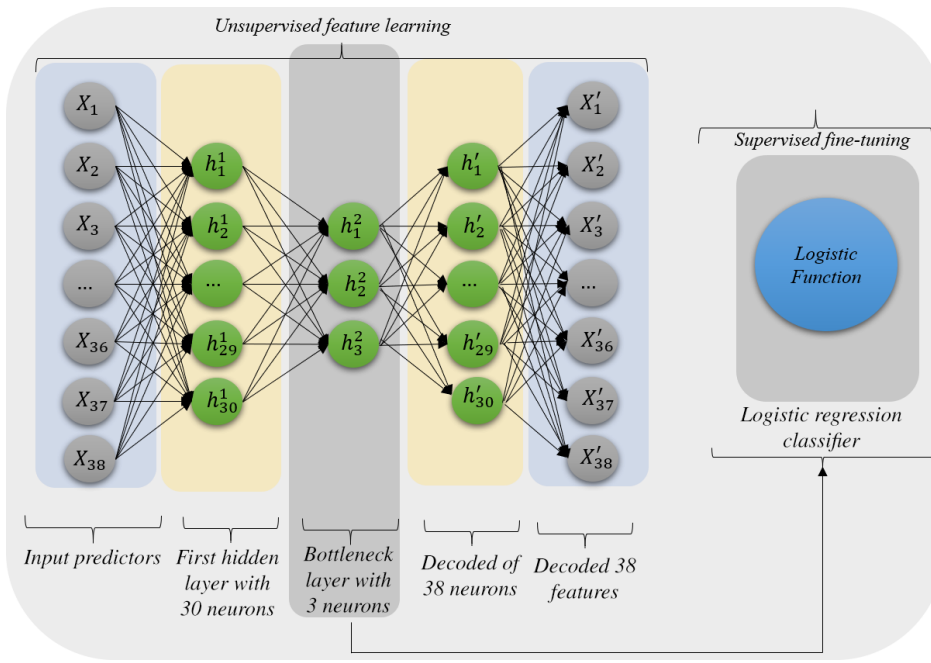


FIGURE 3 Stacked denoising autoencoder (SDAE) architecture applied for thunderstorm prediction

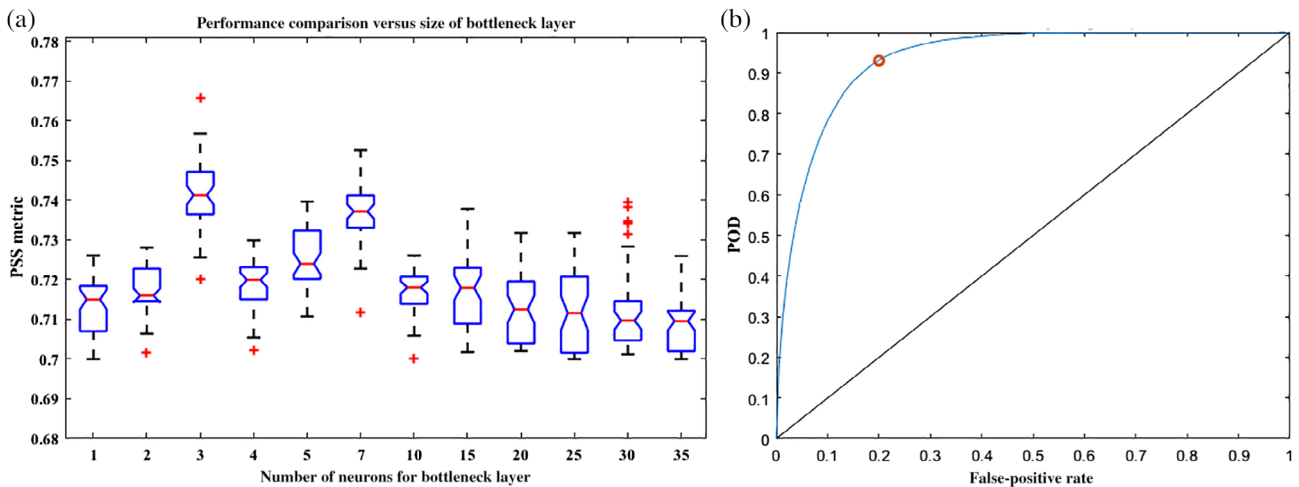


FIGURE 4 (a) Determination of the optimum number of hidden layer neurons for box 238 for a 12 hr prediction stacked denoising autoencoder (SDAE) model. The box plot shows the median and interquartile ranges estimated based on the 50 iterations. The graph is for the bottleneck layer. The number of hidden layer neurons (three) is selected based on the maximum Peirce skill score (PSS), while ensuring that there is not overlap with solutions including less hidden neurons. (b) Receiver operating characteristic (ROC) curve generated over the training set to select the logistic classifier threshold based on a maximizing PSS (0.73)

non-lightning cases are displayed in blue. For the 3D analysis of the SDAE point cloud (Figure 6e), the lightning cases are located along the outermost layer of the feature space forming a wedge around the non-lightning cases. Similarly, for the related 2D analysis, the lightning cases are mapped in the corner and edges (Figure 6f–h). Such segmentation is more distinguishable than that illustrated for PCA in Figure 6b–d (Wang *et al.*, 2016). For both the 3D and 2D cases of the PCA dimension reduction, the lightning cases

are restricted to a portion of the feature space, but they are surrounded and mixed in with non-lightning cases. Another advantage of the SDAE dimension reduction method is the better use of the feature space as the PCA point cloud does not fill the same cube as thoroughly (Figure 6c versus g). The comparisons between the 2D and 3D SDAE dimension reduction also illustrate the need for at least three latent features as the lightning cases are well clustered in a wedge-like shape area.

FIGURE 5 Feature importance of principal component analysis (PCA) model for the training data set

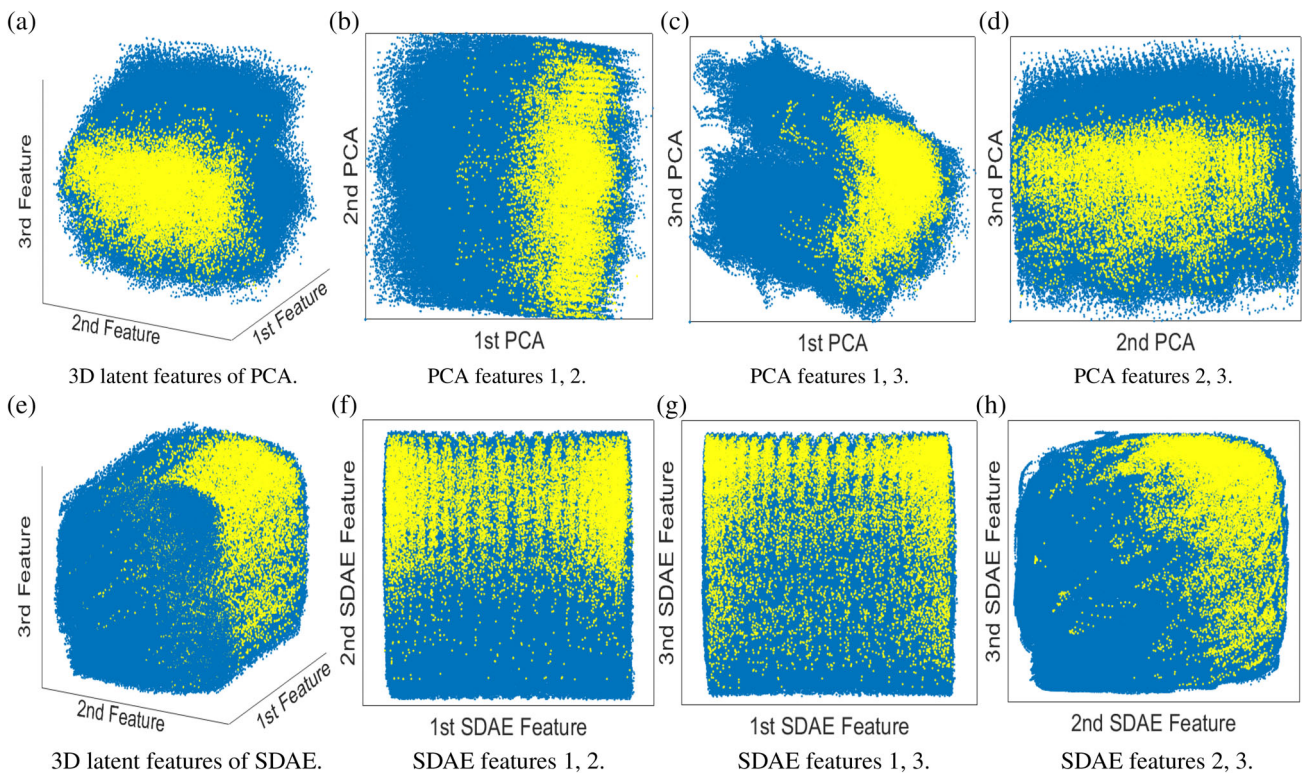
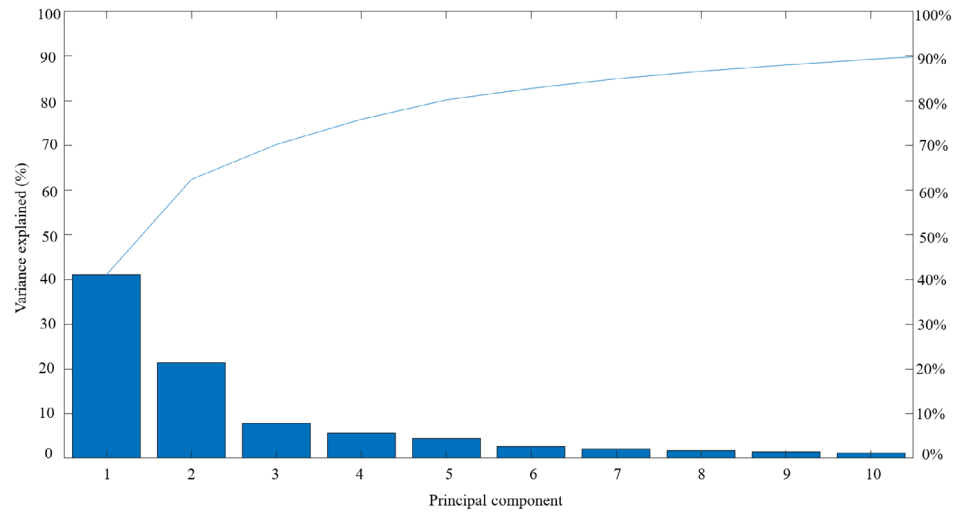


FIGURE 6 Comparison of latent features generated by the stacked denoising autoencoder (SDAE) and principal component analysis (PCA) models. For all cases, blue dots represent non-lightning cases, and yellow dots represent lightning cases in feature space: (a) displays cases in the first three principal components; and (e) cases in the three features identified by the SDAE. (b–d) Cases projected onto a two-dimensional PCA space onto the first two components (b), onto the first and third components (c), and onto the second and third components (d), respectively. Similarly, (f–h) project into the SDAE feature space

3.2 | DLNN model evaluation

To evaluate the performance of the DLNN model, the model was applied to the independent data set (2007–2008). Based on the confusion matrix for binary classes, eight different metrics were calculated. The formulation of the confusion matrix and performance metrics

are shown in Tables 3 and 4, respectively. The metrics include the PSS, critical success index (CSI), Heidke skill score (HSS), odds ratio skill score (ORSS), and so on. See Hogan *et al.* (2010) and Wilks (2011) for more detailed information about the utility of these metrics. Tables 5–7 depict the performance results of the DLNN, shallow neural network and PCA-based classifiers, and the

TABLE 3 Confusion matrix for calculating scalar performance metrics

Forecast	Yes	No	Total
Yes	a (hit)	b (false alarm)	$a + b$
No	c (miss)	d (correct rejection)	$c + d$
Total	$a + c$	$b + d$	$a + b + c + d = n$

TABLE 4 Evaluation metrics

Performance metric	Symbol	Equation
Probability of detection [0, 1]	POD	$a/(a + c)$
False-alarm rate [0, 1]	F	$b/(b + d)$
False-alarm ratio [0, 1]	FAR	$b/(a + b)$
Critical success index [0, 1]	CSI	$a/(a + b + c)$
Peirce skill score [-1, 1]	PSS	$(ad - bc)/(b + d)(a + c)$
Heidke skill score [-1, 1]	HSS	$2(ad - bc)/[(a + c)(c + d) + (a + b)(b + d)]$
Odds ratio skill score [-1, 1]	ORSS	$(ad - bc)/(ad + bc)$
Clayton skill score [-1, 1]	CSS	$(ad - bc)/(a + b)(c + d)$

corresponding performance of the operational forecasters for boxes 73, 103 and 238.

With respect to the 9 hr prediction (Table 5), there was a significant improvement in the value of the selected performance metrics for the DLNN models over the CT2016 model (Collins and Tissot, 2016). For box 73, the CSI metric increased from 0.19 (CT2016) to 0.55 (DLNN), and there was a large improvement of the DLNN models over the CT2016 models for the HSS (0.25–0.54), the CSS (0.19–0.58), and for the ORSS, POD and PSS. The F and FAR metrics were substantially improved; F decreased from 0.22 (CT2016) to 0.10 (DLNN) and FAR decreased from 0.81 to 0.44. For box 103, there was an improvement of the DLNN models over the CT2016 models for the metrics CSI (0.13–0.53), PSS (0.62–0.75), HSS (0.15–0.52) and CSS (0.12–0.51). Also, F and FAR decreased from 0.31 (CT2016) to 0.08 (DLNN) and from 0.87 to 0.43, respectively. For box 238, the superior performance of the DLNN over CT2016 was similar to the performance improvements at the other locations. In general, the DLNN models outperformed the PCA-based models and the operational weather forecasters.

Table 6 summarizes performance comparisons for 12 hr predictions. For box 103, there were approximate 89%, 88% and 91% improvements of the DLNN models over the CT2016 models for CSI, HSS and CSS metrics, respectively. Also, for the DLNN model, the number of false predictions and the false prediction alarm rate decreased significantly. For box 238, there were approximately 87%, 84% and 89% improvements of the DLNN models over the CT2016 models for the CSI, HSS and CSS metrics, respectively, and the F metric decreased from 0.28 (CT2016) to 0.07 (DLNN), and the FAR decreased from 0.93 to 0.37. DLNN improvements over the PCA-based models and operational forecasters continue.

Table 7 summarizes the 15 hr prediction performance. The performance improvement of the DLNN over the CT2016 models continues; the F metric for boxes 73, 103 and 238 decreased by 68%, 66% and 65%, respectively. Also, the FAR decreased by 56%, 60% and 62% for boxes 73, 103 and 238, respectively. For the CSI, HSS and CSS there were 87%, 81% and 87% improvement for box 73, respectively. The improvement of the CSI, HSS and CSS continues for boxes 103 and 238. Continued improvement of the DLNN models over that of the PCA-based variety and the operational forecasters was noted.

As mentioned above, and depicted in Tables 5–7, the DLNN classifiers provided greater performance than the PCA-based and CT2016 shallow neural network models (with respect to all skill-based performance metrics). These results demonstrate that latent features generated by the SDAE are more informative and distinguishable than those provided by the PCA linear dimensionality reduction technique and the nonlinear CFS and mRMR feature selection methods used by CT2016 (and mentioned in Section 2.3.1). These results show that the SDAE (by using nonlinear activation functions and greedy-layer-wise learning) can model the complexity and nonlinearity of original predictors in order to achieve better performance.

Finally, the performance of the DL model was compared with a recent state-of-the-art ensemble prediction system. Bouttier and Marchal (2020) assessed the performance of four ensemble systems (four different sets of ensembles) derived from three separate NWP models, and three ensemble blends (combinations of two ensemble systems), when predicting thunderstorm occurrence (convective initiation) in Western Europe. They defined an ensemble thunderstorm prediction as the probability that a thunderstorm activity diagnostic variable exceeded a specific threshold. The thresholds chosen were the values that maximized the area under the ROC curve (AUC). They defined a thunderstorm observation (the target) as a lightning strike (CG or intracloud) or a maximum radar reflectivity > 35 dBZ. The AUC for the ensembles and ensemble blends for predictions of

TABLE 5 Performance results of the deep-learning neural network (DLN) (stacked denoising autoencoder, SDAE) and principal component analysis (PCA)-based classifiers developed in the present study, the shallow neural network of Collins and Tissot (2016) (CT2016), and the corresponding performance of the operational forecasters (National Digital Forecast Database, NDFD) for 9 hr prediction

	POD	F	FAR	CSI	PSS	HSS	ORSS	CSS
<i>Box 73</i>								
SDAE	0.91	0.10	0.44	0.55	0.75	0.54	0.96	0.58
CT2016	0.94	0.22	0.81	0.19	0.71	0.25	0.96	0.19
NDFD	0.91	0.26	0.83	0.16	0.65	0.21	0.93	0.16
PCA	0.89	0.36	0.94	0.05	0.50	0.06	0.86	0.05
<i>Box 103</i>								
SDAE	0.93	0.08	0.43	0.53	0.75	0.52	0.96	0.51
CT2016	0.93	0.31	0.87	0.13	0.62	0.15	0.93	0.12
NDFD	1.00	0.31	0.85	0.15	0.69	0.19	1.00	0.15
PCA	0.85	0.40	0.96	0.03	0.45	0.04	0.79	0.03
<i>Box 238</i>								
SDAE	0.89	0.09	0.39	0.55	0.73	0.57	0.96	0.53
CT2016	0.94	0.30	0.80	0.20	0.63	0.24	0.94	0.19
NDFD	0.94	0.35	0.81	0.19	0.59	0.21	0.93	0.18
PCA	0.83	0.37	0.94	0.05	0.45	0.06	0.78	0.05

TABLE 6 Performance results of the deep-learning neural network (DLNN) (stacked denoising autoencoder, SDAE) and principal component analysis (PCA)-based classifiers developed in the present study, the shallow neural network of Collins and Tissot (2016) (CT2016), and the corresponding performance of the operational forecasters (National Digital Forecast Database, NDFD) for 12 hr prediction

	POD	F	FAR	CSI	PSS	HSS	ORSS	CSS
<i>Box 73</i>								
SDAE	0.82	0.07	0.36	0.55	0.74	0.66	0.96	0.60
CT2016	0.86	0.29	0.90	0.10	0.57	0.12	0.88	0.09
NDFD	0.91	0.23	0.86	0.14	0.68	0.19	0.94	0.14
PCA	0.89	0.42	0.94	0.05	0.48	0.05	0.84	0.05
<i>Box 103</i>								
SDAE	0.80	0.06	0.34	0.49	0.76	0.62	0.97	0.60
CT2016	0.80	0.23	0.95	0.05	0.58	0.07	0.87	0.05
NDFD	0.80	0.24	0.94	0.06	0.56	0.08	0.86	0.06
PCA	0.85	0.36	0.96	0.03	0.48	0.04	0.82	0.04
<i>Box 238</i>								
SDAE	0.81	0.07	0.37	0.55	0.74	0.59	0.96	0.59
CT2016	0.81	0.28	0.93	0.07	0.53	0.09	0.83	0.06
NDFD	0.67	0.28	0.92	0.07	0.39	0.08	0.67	0.06
PCA	0.83	0.33	0.93	0.06	0.49	0.07	0.81	0.05

≤ 15 hr ranged generally from 0.75 to 0.85. For the present study's DL model, the AUCs averaged over the three diagnostic locations were 0.86 for all three lead times of 9, 12 and 15 hr. The Bouttier and Marchal ensemble performance results cannot be directly compared with the DL model given the differences in spatial resolution and locations as well as the averaging period used to calculate

the mean AUCs, 92 days for Bouttier and Marchal as compared with two years (2007–2008) for the present model. The performance, *via* the AUC metric, of the DL model was similar to that of the ensemble approach of Bouttier and Marchal. Further, the present approach can be implemented operationally with only one deterministic NWP model run, while the ensemble-based models in

	POD	F	FAR	CSI	PSS	HSS	ORSS	CSS
<i>Box 73</i>								
SDAE	0.86	0.07	0.40	0.57	0.73	0.54	0.95	0.56
CT2016	0.92	0.25S	0.93	0.07	0.68	0.10	0.95	0.07
NDFD	0.86	0.24	0.93	0.07	0.61	0.01	0.89	0.07
PCA	0.82	0.34	0.94	0.05	0.47	0.07	0.79	0.05
<i>Box 103</i>								
SDAE	0.91	0.07	0.38	0.56	0.78	0.58	0.97	0.58
CT2016	0.83	0.21	0.96	0.04	0.62	0.05	0.90	0.03
NDFD	1.00	0.19	0.96	0.04	0.81	0.07	1.00	0.04
PCA	0.75	0.25	0.95	0.04	0.49	0.06	0.79	0.04
<i>Box 238</i>								
SDAE	0.84	0.08	0.36	0.59	0.72	0.57	0.95	0.60
CT2016	0.64	0.23	0.95	0.05	0.41	0.06	0.71	0.03
NDFD	0.92	0.23	0.92	0.08	0.69	0.11	0.95	0.07
PCA	0.70	0.20	0.91	0.08	0.50	0.11	0.80	0.07

TABLE 7 Performance results of the deep-learning neural network (DLNN) (stacked denoising autoencoder, SDAE) and principal component analysis (PCA)-based classifiers developed in the present study, the shallow neural network of Collins and Tissot (2016) (CT2016), and the corresponding performance of the operational forecasters (National Digital Forecast Database, NDFD) for 15 hr prediction

Bouttier and Marchal require three to 50 ensemble members (separate NWP model runs). In addition, the NWP model used in the present study has a grid spacing of 12 km, whereas the NWP models used to create three of the ensembles ranged from 1.3 to 10 km. Thus, the computational expense of the present approach is much smaller than that of the ensemble model variety. The similar performance results between the SDAE approach and the Bouttier and Marchal ensemble methods, combined with the substantial relative computational cost savings of the present approach, demonstrates tremendous utility.

4 | CONCLUSIONS

Deep-learning neural network (DLNN) models were developed to predict thunderstorms ≤ 15 hr in advance within 400 km² regions in a South Texas domain in the United States. The models were constructed *via* features originating from numerical weather prediction (NWP) model output variables/parameters that influence and preclude convective development and from location (latitude/longitude) and Julian day variables in order to train the models to predict thunderstorms as a function of location and season. Cloud-to-ground (CG) lightning served as the thunderstorm proxy and as the target. The particular deep-learning technique used was the stacked denoising autoencoder (SDAE), a type of representation learning, whereby unsupervised learning occurs across a multitude of hidden layers in order to create a higher order

representation of the original features as a pre-training step. The highest order representation of the output served as features used to train predictive models *via* logistic regression. The DLNN model's performance exceeded substantially that of corresponding shallow neural network models developed by Collins and Tissot (2016). Collins and Tissot developed shallow feed-forward multilayer perceptron (MLP) models using a second-order learning algorithm and an iterative process to determine the number of hidden layer neurons that optimize performance.

It can be speculated that the superior performance of the DLNN classifiers over the shallow neural network classifiers is due to the ability of the SDAE to identify the nonlinear combination of the initial features that optimizes the performance of the subsequent predictive model. The DLNN (SDAE/logistic regression) predictive models were also compared with predictive models developed using principal component analysis (PCA) as the pre-training step (PCA/logistic regression). The SDAE-based models performed superiorly to that of the PCA-based models. The low optimal dimensionality of the resulting latent features (three) associated with the SDAE allowed for visual comparison between the results of the SDAE nonlinear dimension reduction and latent variables generated by the linear PCA. This comparison illustrates the better clustering of the two categories (lightning and non-lightning) by the nonlinear method and provides an explanation for the superior performance of the SDAE over the PCA.

With respect to the skill-based performance metrics Heidke skill score (HSS) and Peirce skill score (PSS), the

performance of the DLNN models in the study generally exceeded substantially the corresponding performance of operational forecasters (the National Digital Forecast Database (NDFD) in Collins and Tissot, 2015, tabs 11–13, and reproduced in Tables 5–7 in the present study). This superior thunderstorm predictive performance of the DLNN models developed in the present study demonstrates the predictive power of representation learning (the SDAE combined with logistic regression) and suggests future improvement in operational thunderstorm forecasting (a small sample size notwithstanding). Such forecast improvements would benefit society greatly given the adverse socioeconomic impacts of thunderstorms to specific industries such as aviation (Wolfson and Clark, 2006; Ding and Rakas, 2015) and to human life generally (Holle, 2008; Holle and Lopez, 2003; NWS, 2015).

ORCID

Hamid Kamangir  <https://orcid.org/0000-0001-9718-7518>

REFERENCES

- Baldi, P., 2012. *Autoencoders, unsupervised learning, and deep architectures*. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning (pp. 37–49).
- Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H., 2007. *Greedy layer-wise training of deep networks*. In Advances in Neural Information Processing Systems (pp. 153–160).
- Bjerknes, V., 1904. Das Problem der Wettervorhersage, betrachtet vom Standpunkte der Mechanik und der Physik. *Meteor. Z.*, 21, pp.1–7.
- Bouttier, F. and Marchal, H. (2020) Probabilistic thunderstorm forecasting by blending multiple ensembles. *Tellus A: Dynamic Meteorology and Oceanography*, 72(1), 1–19.
- Bradley, A.P. (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7), 1145–1159.
- Collins, W.G. and Tissot, P. (2016) Thunderstorm predictions using artificial neural networks. In: Rosa, J.L.G. (Ed.) *Artificial Neural Networks—Models and Applications*. London: IntechOpen.
- Collins, W. and Tissot, P. (2015) An artificial neural network model to predict thunderstorms within 400 km² South Texas domains. *Meteorological Applications*, 22(3), 650–665.
- Colquhoun, J.R. (1987) A decision tree method of forecasting thunderstorms, severe thunderstorms and tornadoes. *Weather and Forecasting*, 2(4), 337–345.
- Charte, D., Charte, F., Garcia, S., del Jesus, M.J. and Herrera, F. (2018) A practical tutorial on autoencoders for nonlinear feature fusion: taxonomy, models, software and guidelines. *Information Fusion*, 44, 78–96.
- Crook, N.A. (1996) Sensitivity of moist convection forced by boundary layer processes to low-level thermodynamic fields. *Monthly Weather Review*, 124(8), 1767–1785.
- Dalu, G.A., Pielke, R.A., Baldi, M. and Zeng, X. (1996) Heat and momentum fluxes induced by thermal inhomogeneities with and without large-scale flow. *Journal of the Atmospheric Sciences*, 53(22), 3286–3302.
- Dash, M. and Liu, H. (1997) Feature selection for classification. *Intelligent Data Analysis*, 1(1–4), 131–156.
- Ding, W. and Rakas, J. (2015) Economic impact of a lightning strike-induced outage of air traffic control tower: case study of Baltimore–Washington International Airport. *Transportation Research Record*, 2501(1), 76–84.
- Ding, C. and Peng, H. (2005) Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(02), 185–205.
- Ducrocq, V., Tzanos, D. and S en esi, S. (1998) Diagnostic tools using a mesoscale NWP model for the early warning of convection. *Meteorological Applications*, 5(4), 329–349.
- Elmore, K.L., Stensrud, D.J. and Crawford, K.C. (2002) Explicit cloud-scale models for operational forecasts: A note of caution. *Weather and Forecasting*, 17(4), 873–884.
- Findell, K.L. and Eltahir, E.A. (2003) Atmospheric controls on soil moisture–boundary layer interactions. Part II: feedbacks within the continental United States. *Journal of Hydrometeorology*, 4 (3), 570–583.
- Fisher, R.A. (1938) The statistical utilization of multiple measurements. *Annals of Eugenics*, 8(4), 376–386.
- Gagne, D.J., II, Haupt, S.E., Nychka, D.W. and Thompson, G. (2019) Interpretable deep learning for spatial analysis of severe hailstorms. *Monthly Weather Review*, 147(8), 2827–2845.
- Gehring, J., Miao, Y., Metze, F. and Waibel, A., 2013. *Extracting deep bottleneck features using stacked auto-encoders*. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 3377–3381). IEEE.
- Glahn, H.R. and Ruth, D.P. (2003) The new digital forecast database of the National Weather Service. *Bulletin of the American Meteorological Society*, 84(2), 195–202.
- Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. Cambridge: MIT press.
- Hagan, M.T., Demuth, H.B. and Beale, M. (1997) *Neural Network Design*. Boston: PWS Publishing Co..
- Haklander, A.J. and Van Delden, A. (2003) Thunderstorm predictors and their forecast skill for The Netherlands. *Atmospheric Research*, 67, 273–299.
- Hall, M.A. and Smith, L.A., 1999. Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. In FLAIRS conference (Vol. 1999, pp. 235–239).
- Hern andez, E., Sanchez-Anguix, V., Julian, V., Palanca, J. and Duque, N., 2016. *Rainfall prediction: A deep learning approach*. In International Conference on Hybrid Artificial Intelligence Systems (pp. 151–162). Springer, Cham.
- Hinton, G.E., Osindero, S. and Teh, Y.W. (2006) A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 1527–1554.
- Hinton, G.E. and Salakhutdinov, R.R. (2006) Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504–507.
- Hogan, R.J., Ferro, C.A., Jolliffe, I.T. and Stephenson, D.B. (2010) Equitability revisited: Why the “equitable threat score” is not equitable. *Weather and Forecasting*, 25(2), 710–726.
- Holle, R.L., 2008. *Annual rates of lightning fatalities by country*. In 20th International lightning detection conference (Vol. 2425).
- Holle, R.L. and Lopez, R.E., 2003, September. *A comparison of current lightning death rates in the US with other locations and*

- times. In International Conference on Lightning and Static Electricity (pp. 16–18).
- Hornik, K., Stinchcombe, M. and White, H. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366.
- Hossain, M., Rekabdar, B., Louis, S.J. and Dascalu, S., 2015. *Forecasting the weather of Nevada: A deep learning approach*. In 2015 international joint conference on neural networks (IJCNN) (pp. 1–6). IEEE.
- Hsieh, W.W. (2007) Nonlinear principal component analysis of noisy data. *Neural Networks*, 20(4), 434–443.
- Janjic, Z.I., Gerrity, J.P., Jr. and Nickovic, S. (2001) An alternative approach to nonhydrostatic modeling. *Monthly Weather Review*, 129(5), 1164–1178.
- Jolliffe, I. (2011) *Principal component analysis*. Heidelberg: Springer, pp. 1094–1096.
- Jolliffe, I.T. and Cadima, J. (2016) Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), 20150202.
- Kain, J.S., Coniglio, M.C., Correia, J., Clark, A.J., Marsh, P.T., Ziegler, C.L., Lakshmanan, V., Miller, S.D., Jr., Dembek, S.R., Weiss, S.J. and Kong, F. (2013) A feasibility study for probabilistic convection initiation forecasts based on explicit numerical guidance. *Bulletin of the American Meteorological Society*, 94(8), 1213–1225.
- Kalnay, E. (2003) *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge: Cambridge University Press.
- Kamangir, H., Rahnemoonfar, M., Dobbs, D., Paden, J. and Fox, G., (2018) Deep hybrid wavelet network for ice boundary detection in radar imagery. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 3449–3452). IEEE.
- Karlik, B. and Olgac, A.V. (2011) Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4), 111–122.
- Khairoutdinov, M. and Randall, D. (2006) High-resolution simulation of shallow-to-deep convection transition over land. *Journal of the Atmospheric Sciences*, 63(12), 3421–3436.
- Kramer, M.A. (1991) Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2), 233–243.
- Lagerquist, R., McGovern, A. and Gagne, D.J., II. (2019) Deep learning for spatially explicit prediction of synoptic-scale fronts. *Weather and Forecasting*, 34(4), 1137–1160.
- Lee, H., Ekanadham, C. and Ng, A.Y., 2008. *Sparse deep belief net model for visual area V2*. In *Advances in Neural Information Processing Systems* (pp. 873–880).
- Leith, C.E. (1974) Theoretical skill of Monte Carlo forecasts. *Monthly Weather Review*, 102(6), 409–418.
- Li, X., Peng, L., Hu, Y., Shao, J. and Chi, T. (2016) Deep learning architecture for air quality predictions. *Environmental Science and Pollution Research*, 23(22), 22408–22417.
- Liu, H. and Motoda, H. (Eds.). (1998) *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Vol. 453. New York: Springer.
- Lorenz, E.N. (1963) Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2), 130–141.
- Lorenz, E.N. (1969) The predictability of a flow which possesses many scales of motion. *Tellus*, 21(3), 289–307.
- Manzato, A. (2007) A note on the maximum Peirce skill score. *Weather and Forecasting*, 22(5), 1148–1154.
- Mecikalski, J.R., Williams, J.K., Jewett, C.P., Ahijevych, D., LeRoy, A. and Walker, J.R. (2015) Probabilistic 0–1-h convective initiation Nowcasts that combine geostationary satellite observations and numerical weather prediction model data. *Journal of Applied Meteorology and Climatology*, 54, 1039–1059.
- Møller, M.F. (1993) A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4), 525–533.
- NWS. 2015. *Natural Hazards Statistics National Weather Service Office of Climate, Water, and Weather Services*. <http://www.nws.noaa.gov/om/hazstats.shtml>.
- NWS/EMC. 2019. *The North American Mesoscale Forecast System*. <https://www.emc.ncep.noaa.gov/index.php?branch=NAM>.
- Orlanski, I. (1975) A rational subdivision of scales for atmospheric processes. *Bulletin of the American Meteorological Society*, 56(5), 527–530.
- Orville, R.E. (2008) Development of the national lightning detection network. *Bulletin of the American Meteorological Society*, 89(2), 180–190.
- Pashaei, M., Kamangir, H., Starek, M.J. and Tissot, P. (2020). Review and evaluation of deep learning architectures for efficient land cover mapping with UAS hyper-spatial imagery: A case study over a Wetland. *Remote Sensing*, 12(6), 959.
- Pathak, J., Wikner, A., Fussell, R., Chandra, S., Hunt, B.R., Girvan, M. and Ott, E. (2018) Hybrid forecasting of chaotic processes: using machine learning in conjunction with a knowledge-based model. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(4), 041101.
- Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J. and Carvalhais, N. (2019) Deep learning and process understanding for data-driven earth system science. *Nature*, 566 (7743), 195.
- Rogers, E., Black, T.L., Deaven, D.G., DiMego, G.J., Zhao, Q., Baldwin, M., Junker, N.W. and Lin, Y. (1996) Changes to the operational “early” eta analysis/forecast system at the National Centers for environmental prediction. *Weather and Forecasting*, 11(3), 391–413.
- Rotunno, R., Klemp, J.B. and Weisman, M.L. (1988) A theory for strong, long-lived squall lines. *Journal of the Atmospheric Sciences*, 45(3), 463–485.
- Sainath, T.N., Kingsbury, B. and Ramabhadran, B., 2012, March. *Auto-encoder bottleneck features using deep belief networks*. In 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4153–4156). IEEE.
- Saunders, C.P.R. (1993) A review of thunderstorm electrification processes. *Journal of Applied Meteorology*, 32(4), 642–655.
- Scher, S. (2018) Toward data-driven weather and climate forecasting: approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, 45(22), 12–616.
- Simon, T., Fabsic, P., Mayr, G.J., Umlauf, N. and Zeileis, A. (2018) Probabilistic forecasting of thunderstorms in the eastern Alps. *Monthly Weather Review*, 146(9), 2999–3009.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: A simple way to prevent

- neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.
- Scholz, M. (2012) SValidation of nonlinear PCA. *Neural Processing Letters*, 36(1), 21–30.
- Storm Prediction center. 2019. *Short Range Ensemble Forecast (SREF) Products*. <http://www.spc.noaa.gov/exper/sref>.
- Tao, Y., Gao, X., Hsu, K., Sorooshian, S. and Ihler, A. (2016) A deep neural network modeling framework to reduce bias in satellite precipitation products. *Journal of Hydrometeorology*, 17(3), 931–945.
- Trier, S.B. (2003) *Convective storms. Convective Initiation*, 560–570. <https://doi.org/10.1016/B0-12-227090-8/00122-6>.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. and Manzagol, P. A. (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11, 3371–3408.
- Wang, Z. and Bovik, A.C. (2009) Mean squared error: love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1), 98–117.
- Wang, J., Bars, R.L. and Eltahir, E.A. (1996) A stochastic linear theory of mesoscale circulation induced by the thermal heterogeneity of the land surface. *Journal of the Atmospheric Sciences*, 53(22), 3349–3366.
- Wang, Y., Yao, H. and Zhao, S. (2016) Auto-encoder based dimensionality reduction. *Neurocomputing*, 184, 232–242.
- Williams, E., Mushtak, V., Rosenfeld, D., Goodman, S. and Boccippio, D. (2005) Thermodynamic conditions favorable to superlative thunderstorm updraft, mixed phase microphysics and lightning flash rate. *Atmospheric Research*, 76(1–4), 288–306.
- Wilks, D.S. (2011) *Statistical Methods in the Atmospheric Sciences: An Introduction, Electronic Version*, Vol. 100. Cambridge: Academic Press.
- Wilson, J.W., Crook, N.A., Mueller, C.K., Sun, J. and Dixon, M. (1998) Nowcasting thunderstorms: a status report. *Bulletin of the American Meteorological Society*, 79(10), 2079–2100.
- Wold, S., Esbensen, K. and Geladi, P. (1987) Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1–3), 37–52.
- Wolfson, M.M. and Clark, D.A. (2006) Advanced aviation weather forecasts. *Lincoln Laboratory Journal*, 16(1), 31.
- Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., Tang, J. and Madabhushi, A. (2015) Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. *IEEE Transactions on Medical Imaging*, 35(1), 119–130.

How to cite this article: Kamangir H, Collins W, Tissot P, King SA. A deep-learning model to predict thunderstorms within 400 km² South Texas domains. *Meteorol Appl.* 2020;27:e1905. <https://doi.org/10.1002/met.1905>