Comparing Deep Learning Models for Lightning Strike Prediction in a Changing Climate

An Empirical Study

NAME REDACTED

Blekinge Institute of Technology

2024-10-04

Outline

Purpose and objectives

Academic Evaluate and compare the performance of multiple deep learning models for lightning strike prediction across different timeframes.

Practical Construction of a practically applicable model for lightning strike prediction.

Importance

Many practical applications:

- **Early warning systems:** Evacuations can be initiated sooner, airline travel will be safer, etc.
- **Efficient resource allocation:** Standby personal, planning vehicle routes, etc.
- **Assess environmental impact:** Detection of lightning-induced wildfires, stormwater runoff, soil erosion.

The Problem

Current methods largely relies on numerical models and simulations.

- Requires a large and complex network of sensors.
- Requires massive computational resources on a centralized system.
- Requires cooperation on a global scale, meaning large communication overhead.

Deep learning might be a better way.

Research Questions

Too achieve the objectives, the following research questions were proposed:

1. ***How does the proposed models perform across different time frames?***
2. *Which of the proposed models are most effective for predicting lightning strikes, and how do they compare to each other?*
3. *Which are the optimal features and input data for predicting lightning strikes?*
4. *What are the optimal hyperparameters and model configurations for predicting lightning strikes?*
5. *What are the challenges and limitations of using deep learning for lightning prediction, and how can these limitations be mitigated?*

Data source

All data is curated by the *Swedish Meteorological and Hydrological Institute* (SMHI).
Why?

- As SMHI is an official governmental agency, the data is likely legitimate and unbiased.
- As the data is open and publicly available, it is likely free and reproducable.
- Low-level error handling can be outsourced, further increasing its viability.

Data Overview

In total, two datasets were utilized:

1. The *Lightning Archive* (LIGHT dataset), a collection of all lightning strikes occured in sweden, used as labels.
2. The *Meteorological Analysis Model data* (MESAN dataset), a collection of meteorological data, used as features.

LIGHT Overview

The LIGHT dataset is a collection of lightning strikes that occured in Sweden since January 2, 2012.

Has been subjected to initial cleaning procedures and error handling, such as

- Interpolation
- Chi-Square filter

Dynamism

Even though the underlying data is static, the measured data is dynamic. Data is available from before 2012, but before this it was collected on different sensors and hardware.

LIGHT Visualization

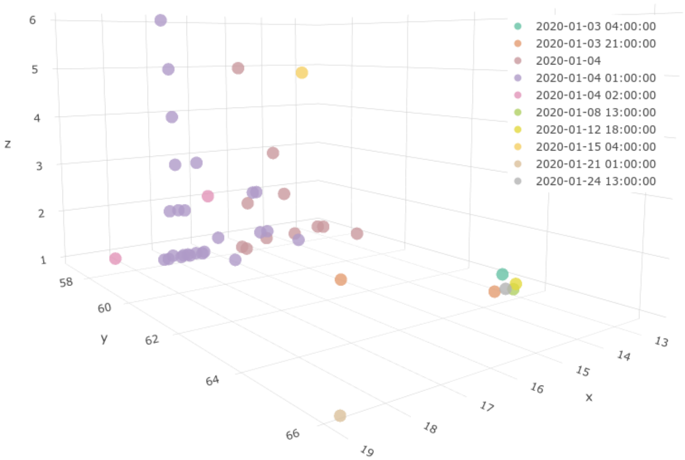


Figure: Visualization of the LIGHT dataset during January 2020.

LIGHT Analysis

The graph illustrates that lightning strikes tend to occur both in close temporal and spacial proximity.

Careful when analyzing labels to not include any "hints" about the testing data.

LIGHT Preprocessing

1. **Filtering:** The data was geographically restricted due to limited computational resources.
   - Filter applied to the vicinity of Linköping $\pm 300km$.
   - Resulting in a total area of $90000km^2$.
   - Might enhance the model accuracy as the topology remains largely the same.
2. **Balancing:** An even ratio of positive/negative instances is preferred.
   1. Negative instances were randomly generated.
   2. Discarded if they didn't adhere to previous filters.
   3. Discarded if they were already present in the positive dataset.
   4. Otherwise added to the dataset.
   5. Repeated until the positive/negative instance ratio was 50/50.
3. **Binning:** The data was clustered to maximize variability.
   - Useful because many positive instances are very similar.
   - Achieved by flooring every timestamp to the nearest hour and coordinates to a specific number of decimals.
   - Duplicate instances were deleted, favoring positive instances to maximize data size. Afterwards the classes were rebalanced.
   - In this study 0 decimals were used, resulting in a bin size of $555.6km$, covering the entire geographic area.
   - A higher decimal count would likely be beneficial.

MESAN Overview

Contains the meteorological data used as input or features to the models.

Released on hourly intervals.

Encompassing $11,679,839 km^2$, including Scandinavia, the Baltic States and the Northern part of Europe.

Available from December 2014. Before that, other hardware was used.

Big data, exceeding a terrabyte in size.

MESAN Parameters

The following parameters are included in the MESAN dataset:

- Pressure
- Temperature
- Wet bulb temperature
- Maximum temperature
- Minimum temperature
- Visibility
- Wind gust
- U-component of wind
- V-component of wind
- Relative humidity
- Total cloud cover
- Low cloud cover
- Medium cloud cover
- High cloud cover
- Fraction of significant clouds
- Cloud base of significant clouds above ground
- Cloud base of significant clouds above sea
- Cloud Top of significant clouds
- Frozen part of total precipitation
- Type of precipitation
- Sort of precipitation
- 12 hour precipitation
- 24 hour precipitation
- 1 hour precipitation
- 3 hour precipitation
- 12 hour snow
- 24 hour snow
- 1 hour snow
- 3 hour snow

MESAN Format

Stored in *Gridded Binary*, or GRIB files.

In order to minimize the distance disparity between grid points, grids and parameters in the files released by SMHI are rotated according to a Lambert projection.

However, the wind vectors can be left unrotated as the rotation remains the same for all files, and because it is their relationship to the rest of the data that matters.

MESAN Exploratory analysis

To get an overview of the data, an exploratory analysis was conducted:

- The dataset is missing a lot of values, with the "minimum/maximum temperate" being non-existent.
- Compound parameters such as "24-hour precipitation" are not continuous and misses values except for on the specific intervals.

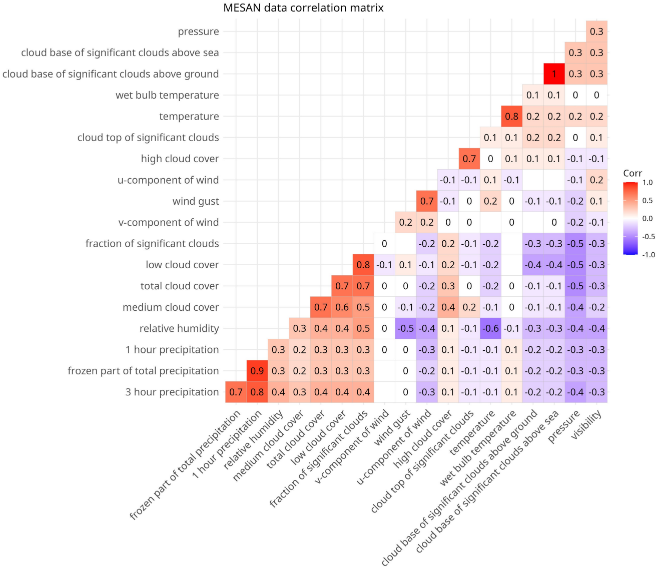
MESAN Correlation analysis

To decrease data processing and limit model confusion, the input data should be independent.

A *covariance matrix* was used to find similar parameters:

1. A Shapiro-Wilk test with $\alpha = 0.05$ was used on all parameters to check for normality.
2. All parameters were normalized.
3. The Spearman's rank correlation coefficient was used to build the matrix.

Covariance Matrix (pre)

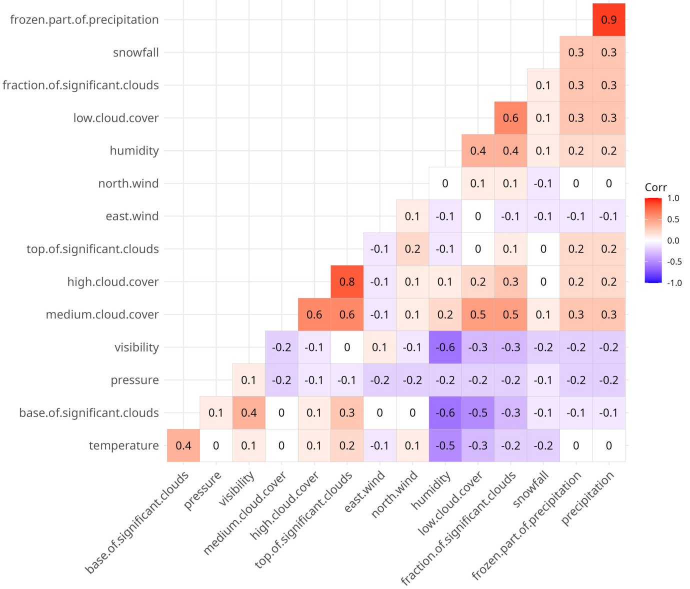


MESAN Correlation analysis

Based on the correlation matrix, the following parameters were removed in favor of more independent features:

- Wet-bulb temperature
- Cloud base of significant clouds above sea
- Total cloud cover
- Wind gust

Covariance Matrix (post)



MESAN PC Analysis

A *Principal Component Analysis*, or PCM, was used to determine the parameters with highest importance.

It can be used to minimize the number of input parameters while retaining the maximum information amount.

The PCA process yields a set of vectors, ranked in order of their impact on the overall variance of the dataset.

MESAN PCA Table

Parameter	PC1	PC2	PC3
pressure	-0.21	-0.02	-0.02
temperature	-0.15	0.37	-0.10
visibility	-0.28	0.15	0.08
east.wind	-0.04	-0.08	0.18
north.wind	0.09	0.12	0.17
humidity	0.32	-0.28	0.07
low.cloud.cover	0.39	-0.15	0.21
medium.cloud.cover	0.37	0.25	0.22
high.cloud.cover	0.28	0.43	0.17
fraction.of.significant.clouds	0.37	-0.06	0.20
base.of.significant.clouds	-0.21	0.43	-0.05
top.of.significant.clouds	0.17	0.51	0.11
frozen.part.of.precipitation	0.30	0.08	-0.55
precipitation	0.25	0.09	-0.56
snowfall	0.13	-0.04	-0.35

MESAN PCA Results

- The data is relatively independent as 12/15 PCs are required for 95% comprehensiveness.
- PC1 is dispersely influenced, but primarily by cloud related parameters ("low/medium cloud cover" and "fraction of significant clouds").
- PC2 is similarly comprised of cloud related parameters ("base/top of significant clouds" and "high cloud cover").
- PC3 is influenced by precipitation related parameters ("precipitation" and "frozen part of precipitation").
- The least important parameters seem to include "snowfall", "temperature" and wind-related parameters.

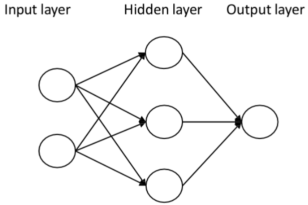
MESAN Preprocessing

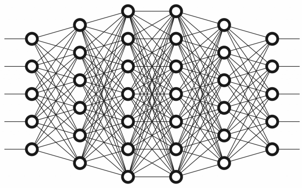
1. **Extraction:** The data had to be extracted from the GRIB files.
    1. For each lightning strike a trailing sequence of 73 weather observations were extracted.
    2. The coordinates for each lightning strike was rotated.
    3. A mean of all metrics gathered within a 3 km radius were calculated.
    4. If no metrics were found, the radius was incrementally expanded up to 6 km.
    5. If no value was found, it was labelled as missing.
2. **Imputation:** After extraction the dataset contained multiple missing values.
    - Missing values were derived from the closest known value, either forwards or backwards in time.
    - Group boundaries were kept to ensure no imputation from other lightning strikes occured.

DNNs

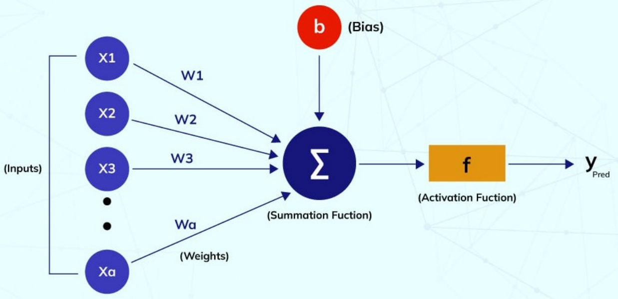
Dense Neural Networks, or DNNs, are the most basic type of DL model.

- Has an input layer, an output layer, and at least one hidden layer in between.
- The input layer varies depending on the input parameters.
- The output layer varies depending on the purpose and goal of the model.





Perceptron/Node



As the goal is binary classification, an output layer with a single node is used.

This node uses the *Sigmoid* activation function, making the model output a fraction between 0 and 1; the probability of the input being negative or positive instance.

SRNNs

Simple Recurrent Neural Networks: a modified variation of DNNs with the ability to take multiple instances into account simultaneously.

Useful for timeseries data where the changes between the values over time is as important as the values themselves.

Suffers from the "vanishing gradient" problem, were the importance of older instances diminishes resulting in neglected observations.

LSTMs

Long Short Term Memory, or LSTMs: a modified variation of SRNNs that attempts to solve the vanishing gradient issue.

Adds three "gates" in each node, where each gate controls what information is going to be forgotten, remembered or passed on.

GRUs

Gated Recurrent Units, or GRUs, use another gating mechanism to mitigate the vanishing gradient problem.

Compared to LSTMs they are newer and uses a more computationally efficient gating mechanism.

Hyperparameter Tuning

Every DL model has a set of internal and external hyperparameters.

- Internal hyperparameters are optimized by the model to fit the training data, such as weights and biases.
- External hyperparameters cannot be changed as they are part of the model itself, such as layer/node count.

You can not change the tires of a car while driving.

Common Tuning Methods

Some common methods used for finding the optimal hyperparameters:

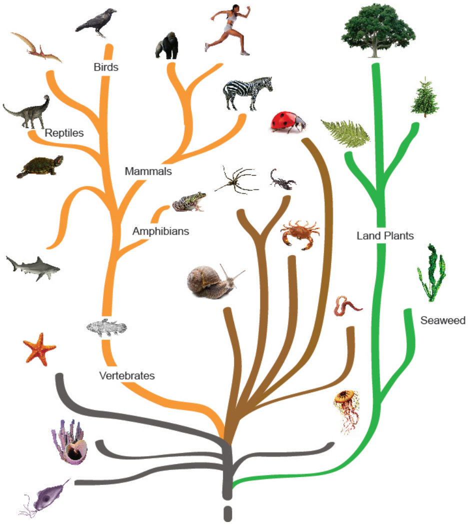
- **Random search:** Random hyperparameters are tested. After N iterations or seconds the best performing combination is used.
- **Grid search:** All possible hyperparameters combinations are tested. It is certain to give the optimal hyperparameters, at large computational costs.

Genetic Tuning

In this study, a genetic algorithm was used. These have shown to be highly promising for tuning.

1. A population of 100 "candidates" were generated. Each candidate is a combination of external hyperparameters.
2. A fitness value is calculated for every candidate, were the fitness is essentially its accuracy with a small penalty for training time.
3. The top 10% best performing candidates immediately survive to the next generation.
4. The rest of the population is replaced by sampling two sequences of candidates, with the probability of selection proportional to their fitness.
5. The two sequences are merged to create "offspring", inheriting its values from either parent and a 5% chance of random mutation.

Evolution Tree



Tuning Results

Due to hardware restrictions, only three generations were able to be simulated, making it somewhat better than a random search.

The following hyperparameters were produced:

- Batch size: 128
- Optimizer: adam
- Activation function for the recurrent layers: tanh
- Activation function for the dense layers: relu
- Dropout for the recurrent layers: 0.2
- Dropout for the dense layers: 0.2
- Number of recurrent layers: 2
- Number of dense layers: 0
- Number of units in the recurrent layers: 256
- Number of units in the dense layers: 64
- Normalization between recurrent layers: yes
- Normalization between dense layers: yes

Timeframes & lookback/lookahead

In order to assess models over timeframes, two evaluation factors *lookback* and *lookahead* are introduced.

Lookback is the number of time steps to take into account when making a prediction.

Lookahead is the of number of time steps within which to make a prediction.

Timeframe restructuring

Modifying the data based on lookback is simple, just use the *n* last observations.

Modifying based on lookahead is more complicated. Simply increasing the lookahead period still means the lightning strike always occur within the first hour.

```
_ _ x x x x | y
_ x x x x | _ y
x x x x | _ _ y
```

Timeframe restructuring

Furthermore, within the newly created lookahead sequence, another independent lightning strike might have occurred.

```
x x x x | _
x x x x | _ _ z
```

This will increase the number of positive classes, and at this point it is not feasable to repopulate negative instances.

To accomodate for this, class weights are used as a last resort.

Stratified k-fold cross validation

A strategy for testing, with the purpose of minimizing randomness and luck. In this study $k = 10$.

1. Split the data into 10 different partitions, where each partition contains an equal number of positive and negative instances.
2. Iterate through the training and evaluation phase k times, with each iteration using a different partition as test set and the remaining partitions as training set.
3. Calculate the evaluation metric for each iteration. Set the final evaluation metric as the mean.

Metrics

The following metrics were collected for each model:

- **Training Time:** The time it takes to train the model. Provides an indication of its computational requirements.
- **Accuracy:** A simple and widely used metric providing the frequency of correct predictions made by a model.
- **F1-Score:** F1-score is similar to accuracy, but also takes class balance into account.
- **Mean Absolute Error:** A metric that provides insight into the general confidence of the models.

Model Performance: DNN

- Very short training time of 9s.
- Good accuracy overall of 88-89%.
- Very concistent performance across lookahead values (mean diffence of 0.15%).
- Remember, lookback is constant for this model.

Model Performance: SRNN

- Highest training time of 130s.
- Highest overall accuracy, avaraging 89-90%.
- Most confident model overall.

Model Performance: LSTM

- Decent training time of 58s.
- Highest accuracy for shorter lookback/lookahead combinations (92%).
- Sharp dropoff in accuracy once lookback > 1, lookahead > 6.

Model Performance: GRU

- Very similar to the LSTM model, but slightly better across all metrics.
- 9s shorter training time.
- Fractional accuracy increase.

Model Comparison

Which model is the best? Depends on the use-case.

- If only short lookback/lookahead combinations are required, the GRU model can yield up to 4% better accuracy than DNN/SRNN models.
- If higher lookahead is required, DNN or SRNN models are recommended.
  - DNNs significantly faster, at the cost of slightly lower accuracy.
- If high lookback and lookahead is required, the SRNN model is recommended.

Findings

LSTM and GRU models, specifically designed for timeseries prediction, performs better when not concidering the timeseries aspect.

Using timeseries only seems to confuse the models.

Challanges and limitations

- **Generalization:** Getting the model to transfer its knowledge across different geographic locations and topologies.
- **Data Availability:** Data may vary based on location, source and standardization.
- **Interpretability:** DL models are notoriously difficult to understand. Similar to a surgeon trying to understand the thoughtprocess of a patient by looking at their brain.

Improvements and future work

- **Dataset:** Increasing the dataset. In this study only 50,000 of 3,000,000 lightning strikes could be used. Experimenting with different parameters might also be beneficial.
- **Different topologies:** Examining model performance across different topologies and environments. Is it possible to include topological data as input parameters?
- **Model architecture:** Exploring more complex architectures such as convolution/recurrent hybrids. Running the genetic tuning process for longer periods of time is also recommended.
- **Better dataset balance:** Currently, there is likely significant differences between positive and negative instances. A better evaluation would take more similar and realistic instances into account.

Fin

Fin