Towards Machine Learning in NowCastMIX

… learning to make something learned …

Bölöni Gergely, DWD, FE22

Towards Machine Learning in NowCastMIX

… learning to make something learned …

Write a four line poem about the relationship of severe weather warning and machine learning

2/2

Machine learning predicts the storm's fierce might,
Data-driven insights guide through the night,
Algorithms warn, safeguarding our way,
Human and tech united, weather kept at bay.

Írj egy négysoros verset a viharjelzés és a gépi tanulás kapcsolatáról

2/2

Viharjelzés pontosságát gép tanulja meg,
Adatokból jósol, mint bölcs jós, figyel,
Algoritmusokban az előrejelzés rejlik,
Ember és gép együtt, vihartól nem félünk.

Írj egy négysoros verset a viharjelzés és a gépi tanulás kapcsolatáról de csak magyar nyelvű szövegen tanulj

Gépi tanulás szélvészt fürkész,
Adatokból épít jövőt és igéz,
Pontosan jósol, viharjelzést ad,
Biztonságot nyújt, míg az ég harcba száll.

ChatGPT-4

Bölöni Gergely, DWD, FE22

NowCastMIX in a nutshell

NowCastMIX in a nutshell

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

NowCastMIX: Automatic Integrated Warnings for Severe Convection on Nowcasting Time Scales at the German Weather Service

PAUL M. JAMES, BERNHARD K. REICHERT, AND DIRK HEIZENREDER

Deutscher Wetterdienst, Offenbach, Germany

(Manuscript received 9 March 2018, in final form 15 August 2018)

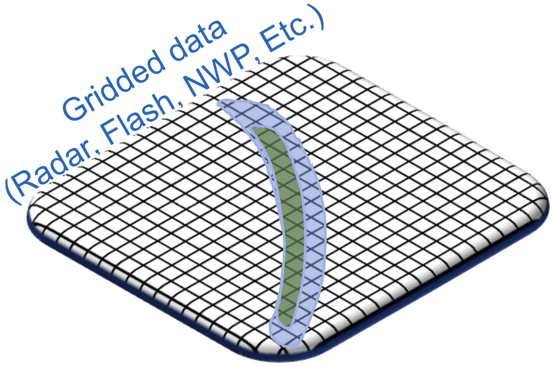
ABSTRACT

NowCastMIX is the core nowcasting guidance system at the German Weather Service. It automatically monitors several systems to capture rapidly developing high-impact mesoscale convective events, including 3D radar volume scanning, radar-based cell tracking and extrapolation, lightning detection, calibrated precipitation extrapolations, NWP, and live surface station reports. Within the context of the larger warning decision support process AutoWARN, NowCastMIX integrates the input data into a high-resolution analysis, based on a fuzzy logic approach for thunderstorm categorization and extrapolation, to provide an optimized warning solution with a 5-min update cycle for lead times of up to 1 h. Feature tracking is undertaken to optimize the direction of warning polygons, allowing individual, tangentially moving cells or cell clusters to be tracked explicitly. An adaptive ensemble clustering is deployed to reduce the spatial complexity of the resulting warning fields and smooth noisy temporal variations to a manageable level for duty forecasters. Further specialized outputs for civil aviation and for a public mobile phone warning app are generated. Now in its eighth year of operation, a comprehensive and complete set of thunderstorm analyses and nowcasts over Germany has been created, which is of unique value for ongoing research and development efforts for improving the system, as well as for addressing climatological aspects of severe convection. Verification has shown that NowCastMIX has helped to significantly improve the quality of the official warnings for severe convective weather events when used within the AutoWARN process.

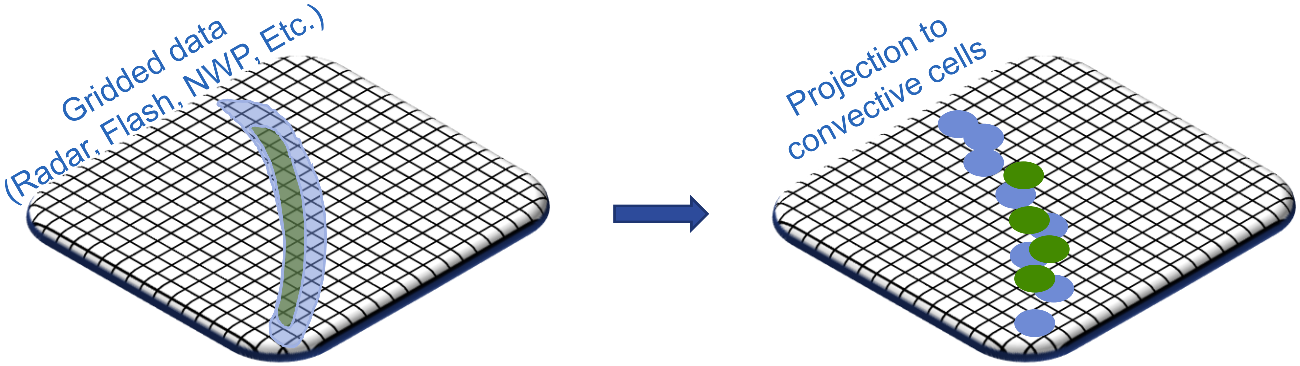
James et al., 2018, WAF

NowCastMIX in a nutshell

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

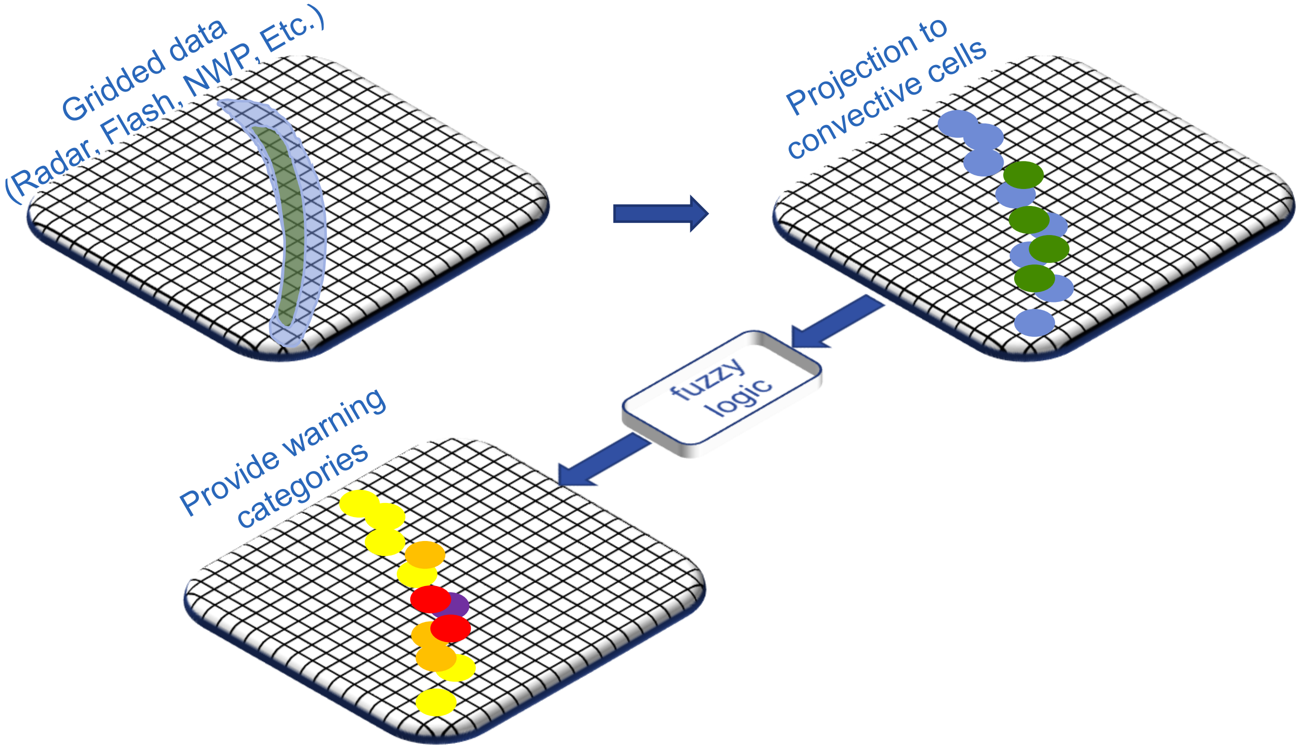


NowCastMIX in a nutshell

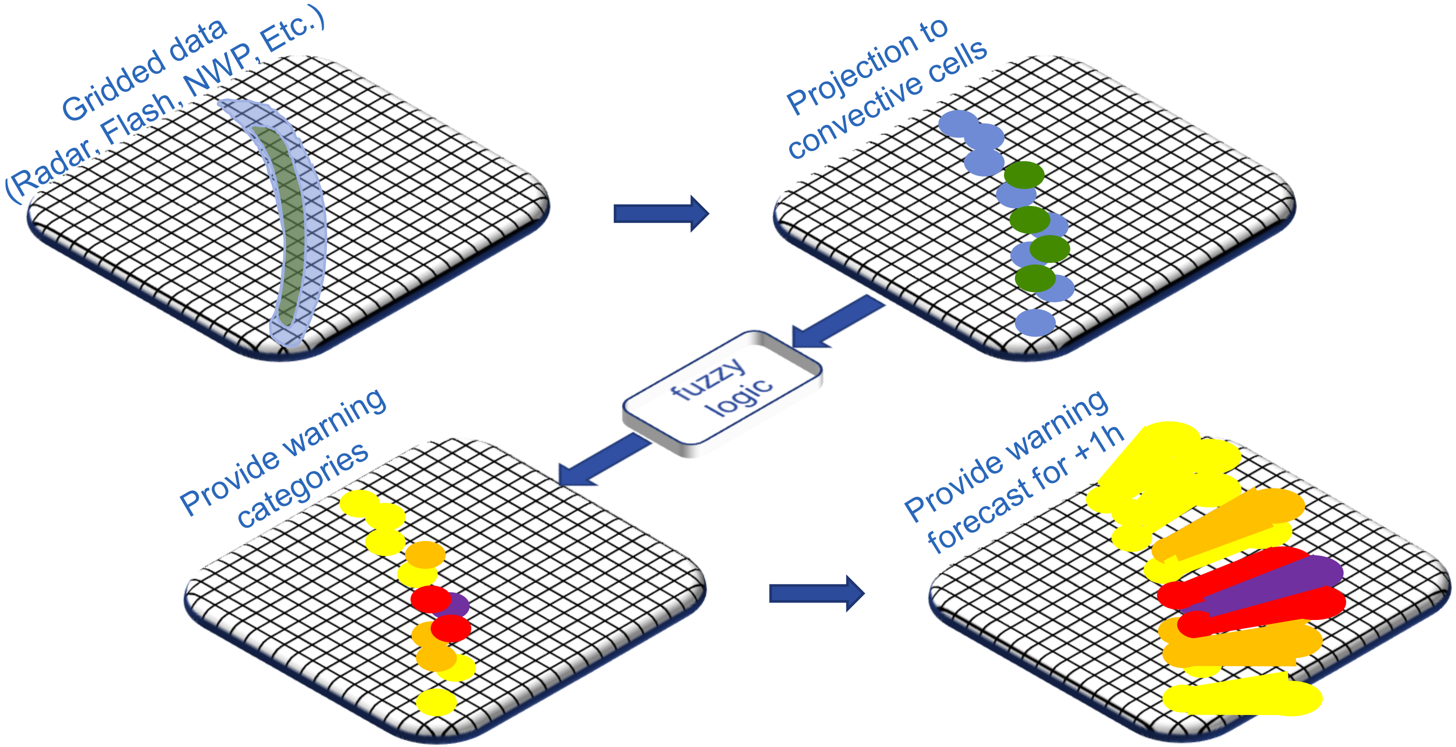


NowCastMIX in a nutshell

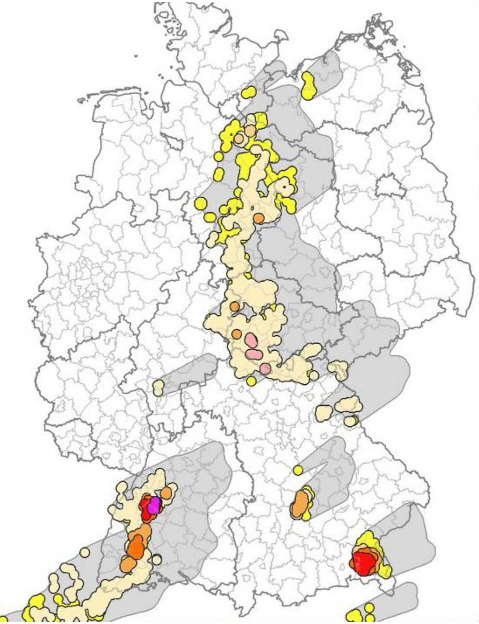
Deutscher Wetterdienst
Wetter und Klima aus einer Hand



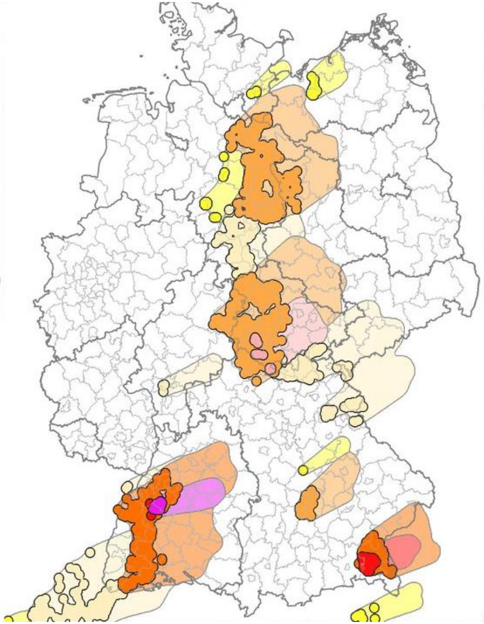
NowCastMIX in a nutshell



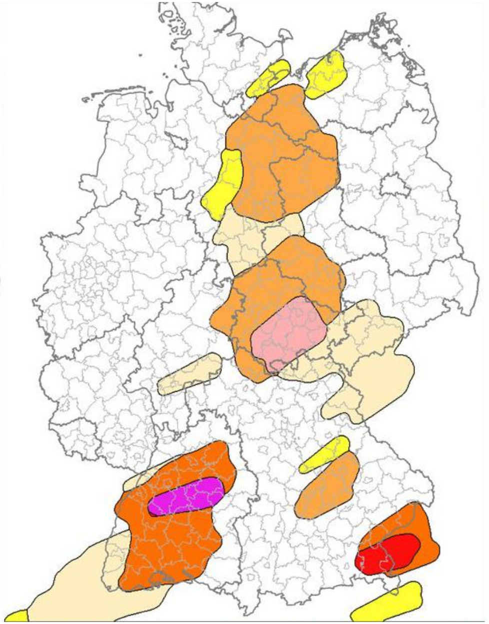
NowCastMIX in a nutshell



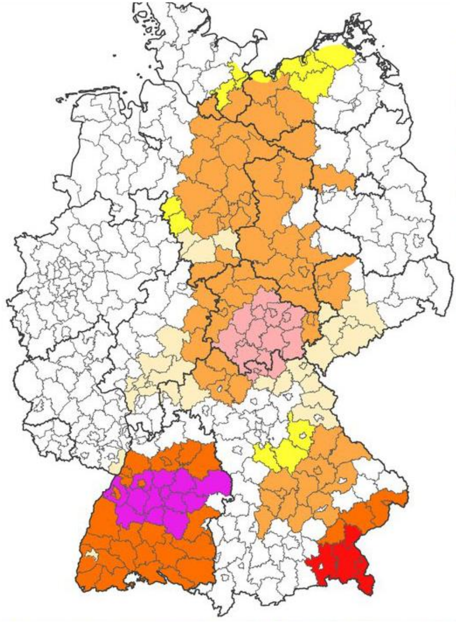
Analysis + Warning areas



Clusters + Warning polygons



Warning suggestion for the next hour

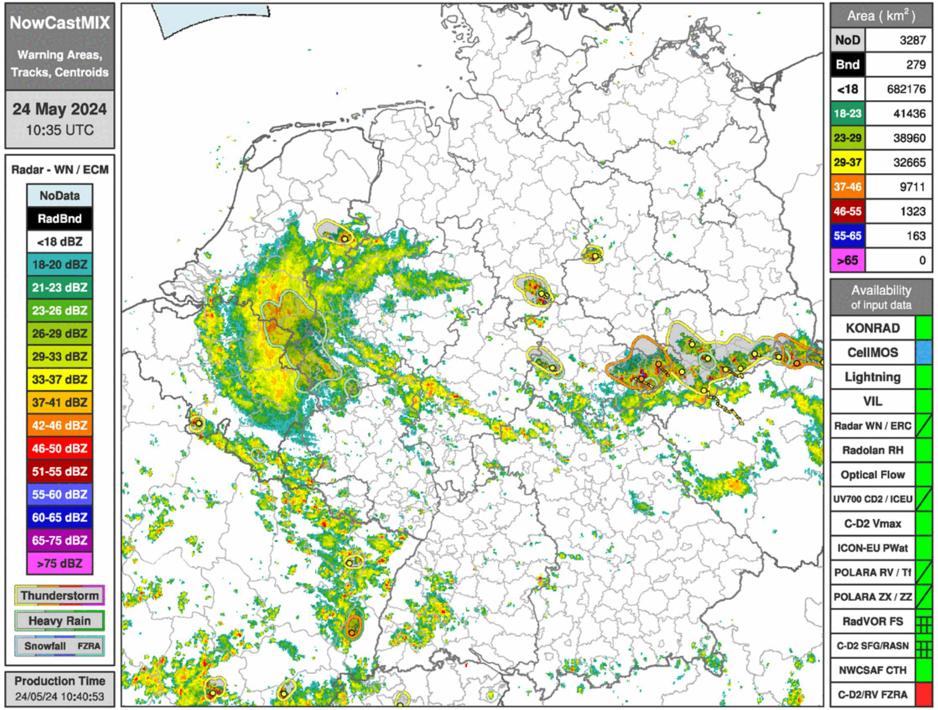


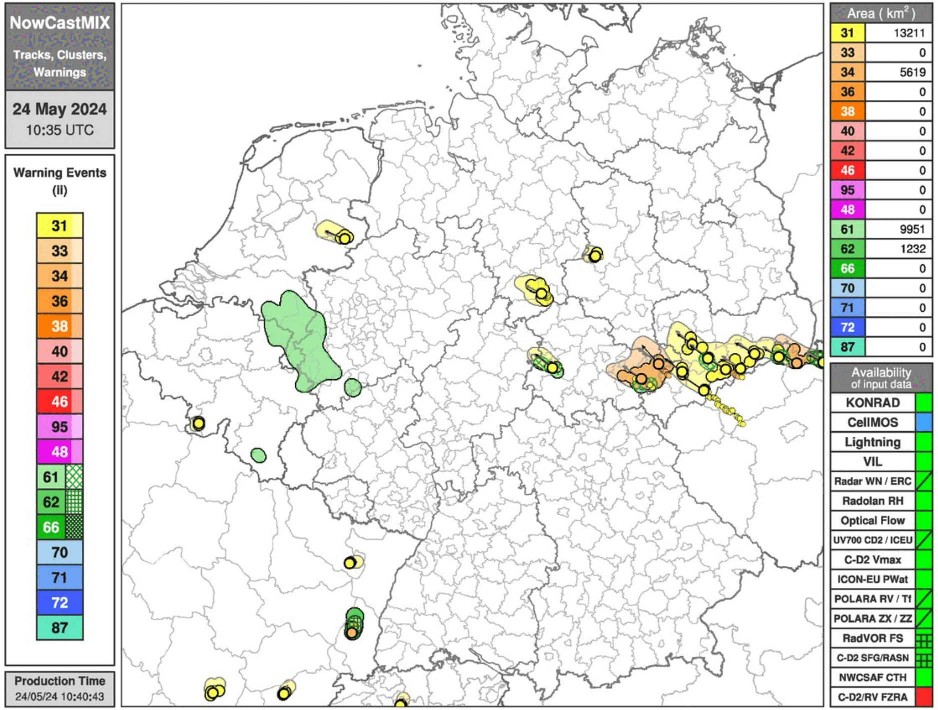
Example: projection on counties

courtesy: Paul James

NowCastMIX in a nutshell

Deutscher Wetterdienst
Wetter und Klima aus einer Hand





Disclaimer

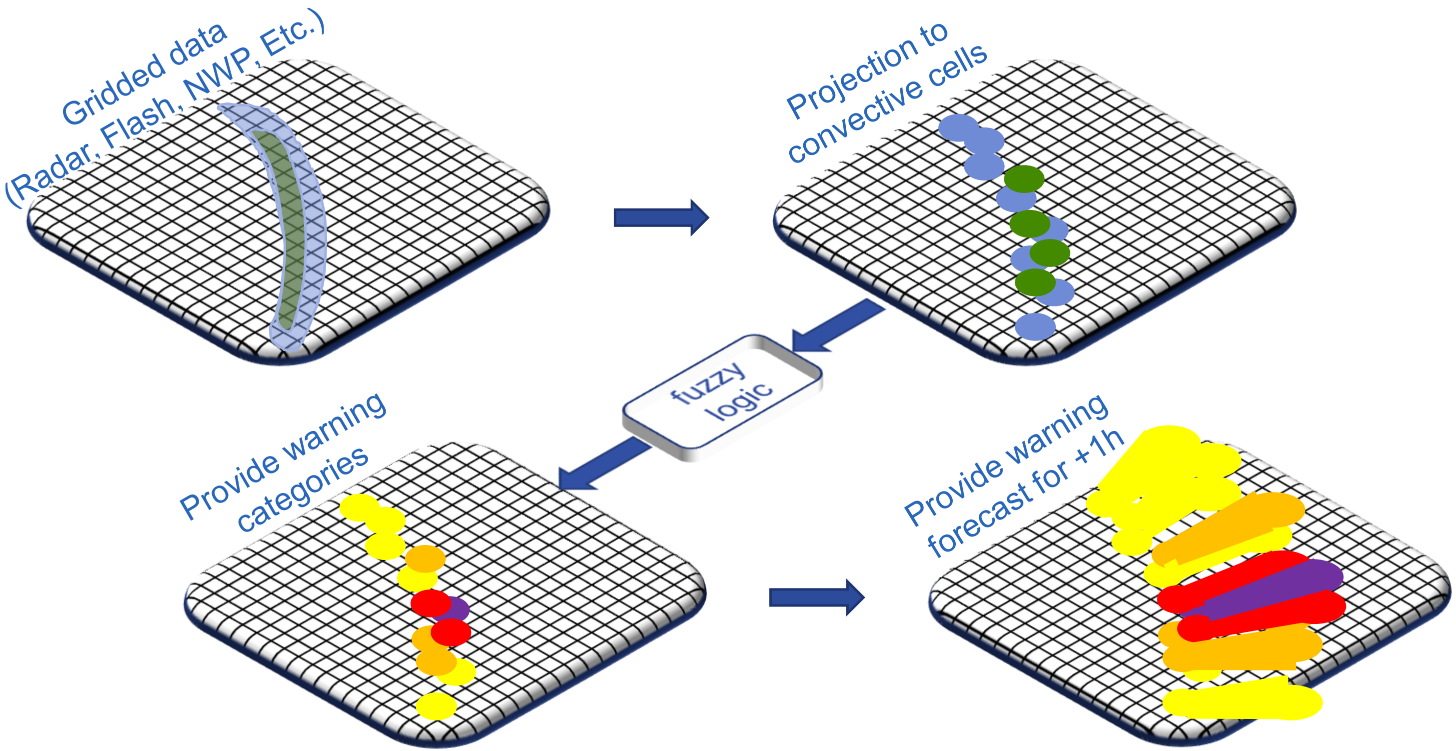
Disclaimer

gifdb.com

EXPERIMENTAL !

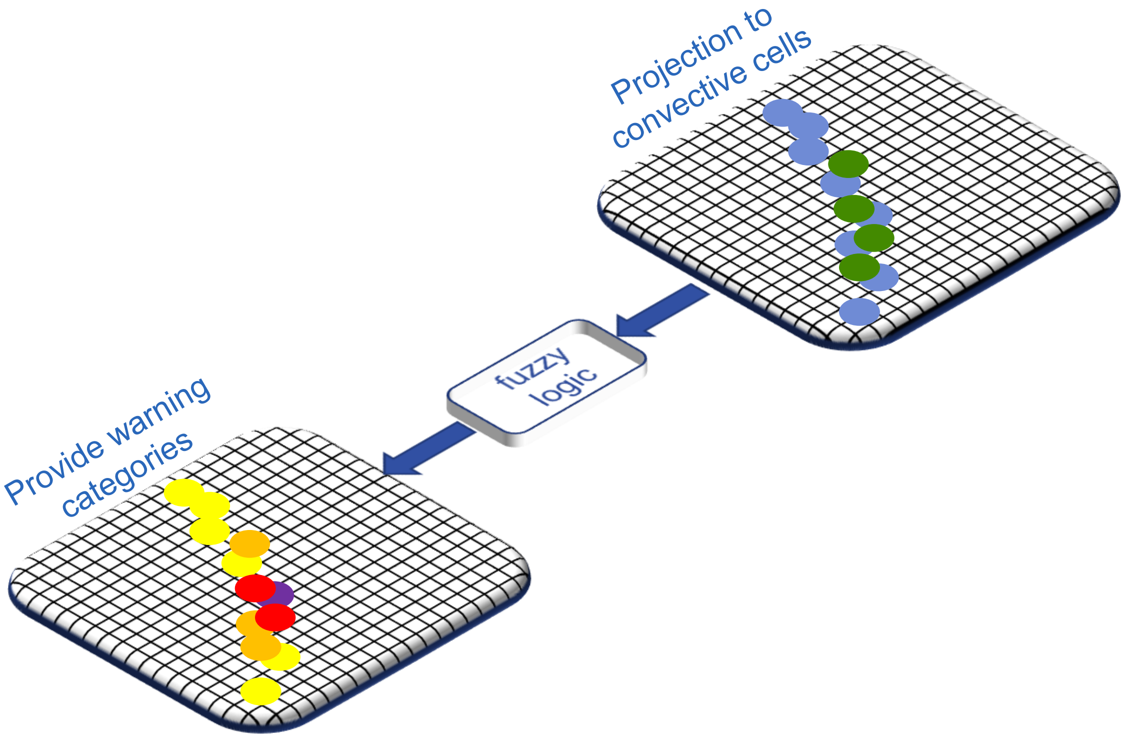
Motivation

Motivation



Gridded data
(Radar, Flash, NWP, Etc.)
Projection to
convective cells
fuzzy
logic
Provide warning
categories
Provide warning
forecast for +1h

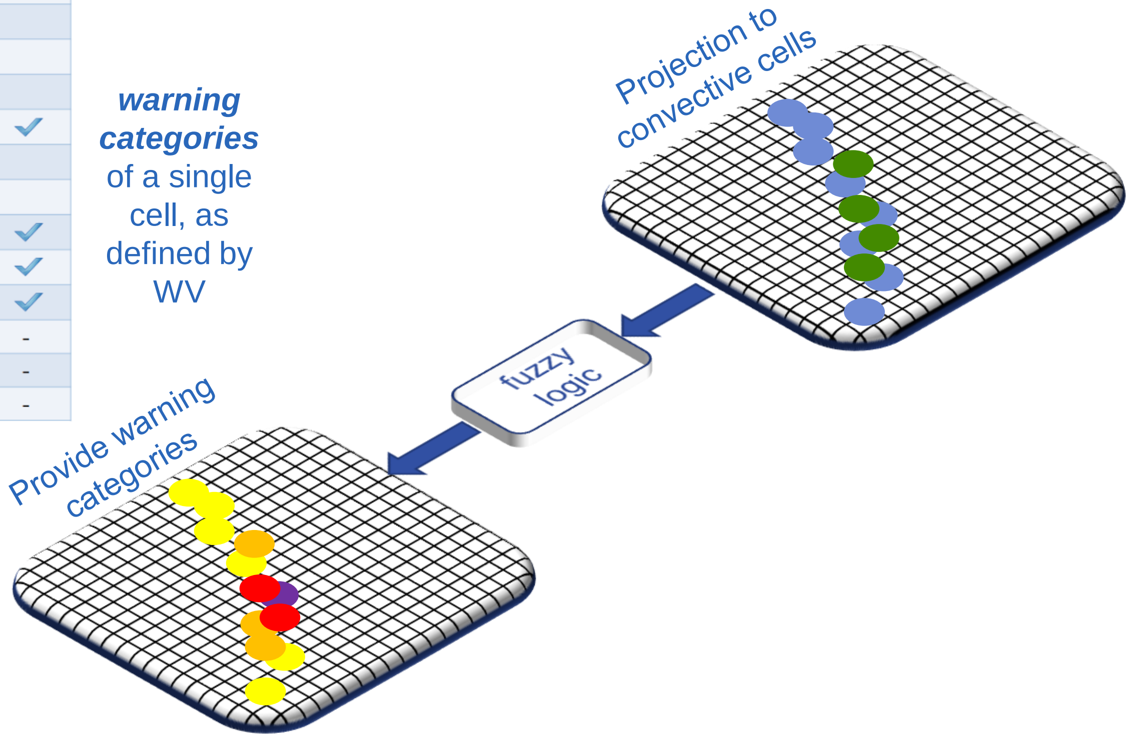
Motivation


Projection to convective cells
fuzzy logic
Provide warning categories

Motivation

ii	Böen Bft.	Regen mm/h	Hagel
31	≤ 7	<15	
33	≤ 10	<15	
34	≤ 7	15-25	
36	≤ 10	15-25	
38	≤ 10	15-25	✓
40	≤ 12+	<15	
42	≤ 9	25-40	
46	≤ 10	25-40	✓
95	≤ 9	>40	✓
48	≤ 12+	>25-40	✓
61	-	15-25	-
62	-	25-40	-
66	-	>40	-

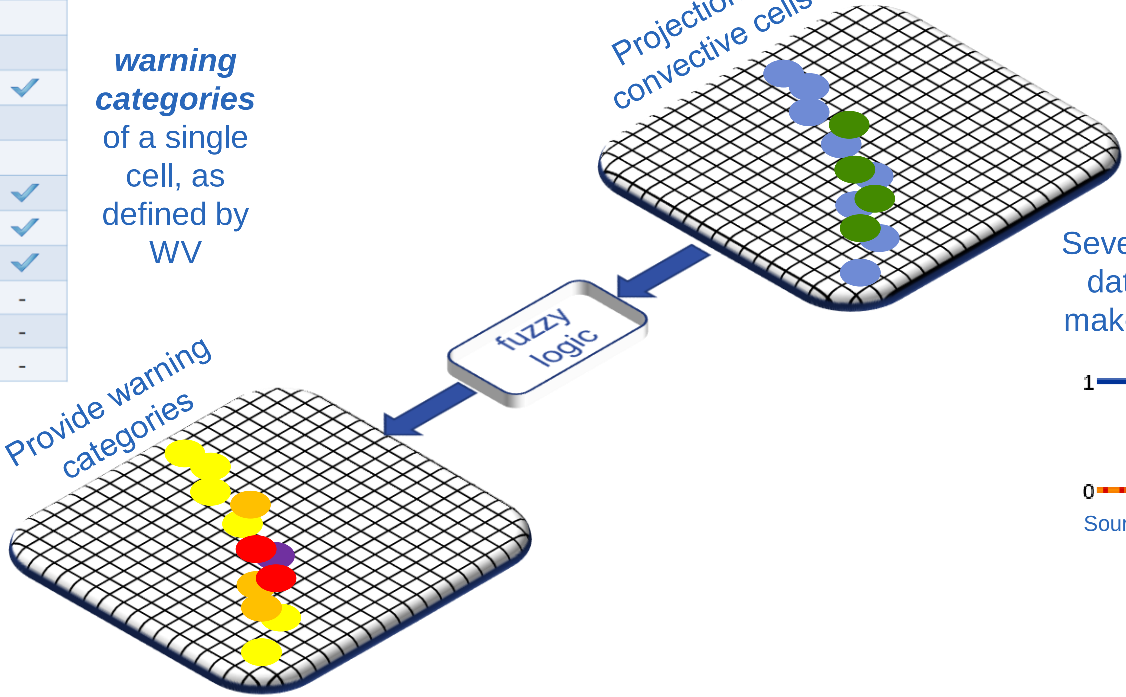
warning categories of a single cell, as defined by WV



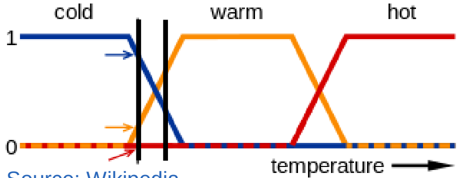
Motivation

ii	Böen Bft.	Regen mm/h	Hagel
31	≤ 7	<15	
33	≤ 10	<15	
34	≤ 7	15-25	
36	≤ 10	15-25	
38	≤ 10	15-25	✓
40	≤ 12+	<15	
42	≤ 9	25-40	
46	≤ 10	25-40	✓
95	≤ 9	>40	✓
48	≤ 12+	>25-40	✓
61	-	15-25	-
62	-	25-40	-
66	-	>40	-

warning categories of a single cell, as defined by WV



Fuzzy Logic:
Several (***possibly contradicting***) data sources at the same time makes the categorization ***fuzzy***…

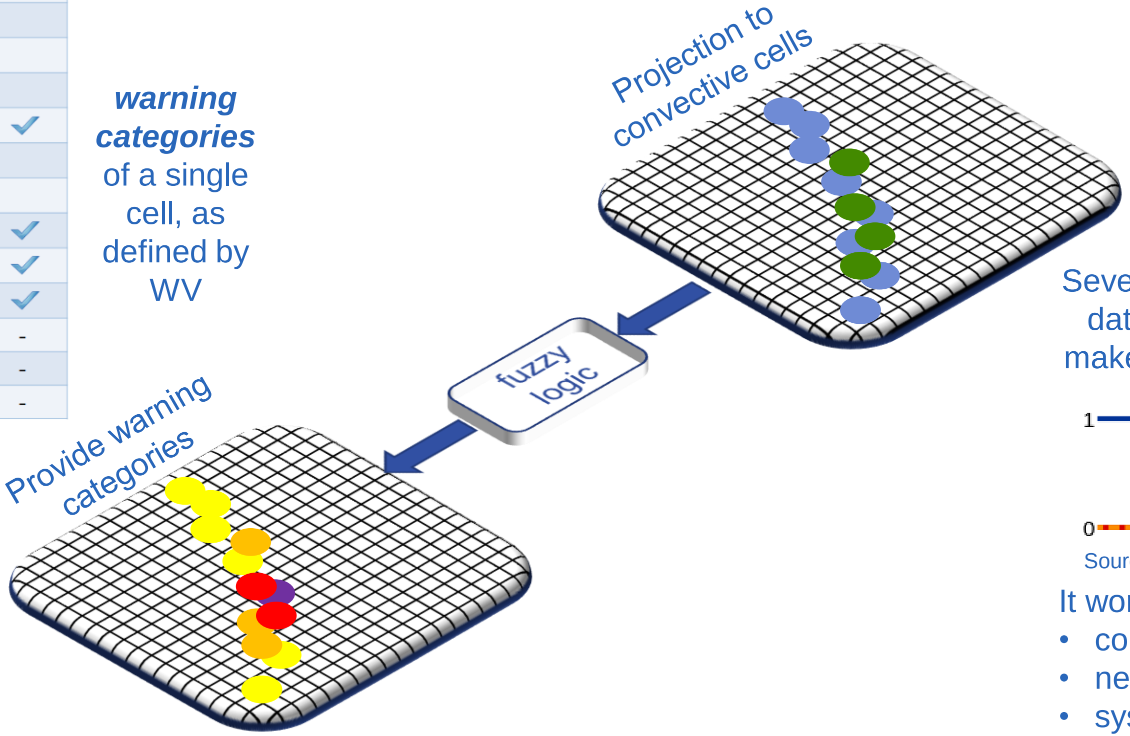


Source: Wikipedia

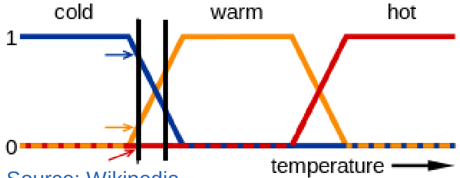
Motivation

ii	Böen Bft.	Regen mm/h	Hagel
31	≤ 7	<15	
33	≤ 10	<15	
34	≤ 7	15-25	
36	≤ 10	15-25	
38	≤ 10	15-25	✓
40	≤ 12+	<15	
42	≤ 9	25-40	
46	≤ 10	25-40	✓
95	≤ 9	>40	✓
48	≤ 12+	>25-40	✓
61	-	15-25	-
62	-	25-40	-
66	-	>40	-

warning categories of a single cell, as defined by WV



Fuzzy Logic:
Several (***possibly contradicting***) data sources at the same time makes the categorization ***fuzzy***…



Source: Wikipedia

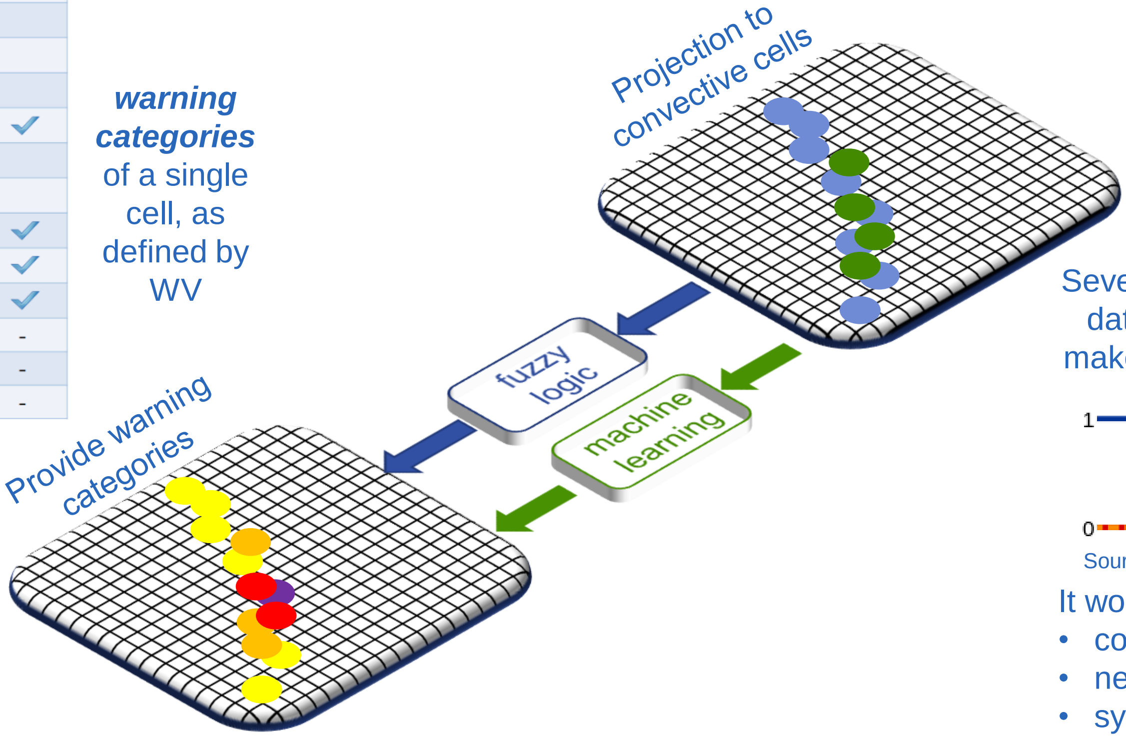
It works well, but:

- complex
- new data → expert tuning
- system upgardes → expert tuning

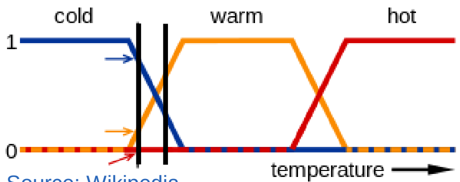
Motivation

ii	Böen Bft.	Regen mm/h	Hagel
31	≤ 7	<15	
33	≤ 10	<15	
34	≤ 7	15-25	
36	≤ 10	15-25	
38	≤ 10	15-25	✓
40	≤ 12+	<15	
42	≤ 9	25-40	
46	≤ 10	25-40	✓
95	≤ 9	>40	✓
48	≤ 12+	>25-40	✓
61	-	15-25	-
62	-	25-40	-
66	-	>40	-

warning categories of a single cell, as defined by WV



Fuzzy Logic:
Several (***possibly contradicting***) data sources at the same time makes the categorization ***fuzzy***…



Source: Wikipedia

It works well, but:
- complex
- new data → expert tuning
- system upgardes → expert tuning

Motivation

Features:

Meteorological state at cell centres

Variable	Source
Time (yy,mm,dd,hh,mm)	Cell occurence
Location (lat, lon)	Cell location
Wind (speed, dir)	KONRAD3D AutoRadSatW ZVF ICON-D2 V-CVF
Precipitation	Radolan RH Radolan RW
Precipitable Water	Radar (VIL) ICON-D2
Hail flag	KONRAD 3D
Cell category	KONRAD3D
Flash rate	LINET
Cloud-top height	NWC-SAF

machine learning: classification problem

Target (fuzzy-logic):

warning categories

ii
31
33
34
36
38
40
42
46
95
48
61
62
66

Motivation

Features:

Meteorological state at cell centres

Variable	Source
Time (yy,mm,dd,hh,mm)	Cell occurence
Location (lat, lon)	Cell location
Wind (speed, dir)	KONRAD3D AutoRadSatW ZVF ICON-D2 V-CVF
Precipitation	Radolan RH Radolan RW
Precipitable Water	Radar (VIL) ICON-D2
Hail flag	KONRAD 3D
Cell category	KONRAD3D
Flash rate	LINET
Cloud-top height	NWC-SAF

machine learning: classification problem

NowCastMIX archive:

- ~ 5 years of labeled data
- ~ 10^6 data instances / year
- ground truth: fuzzy-logic!

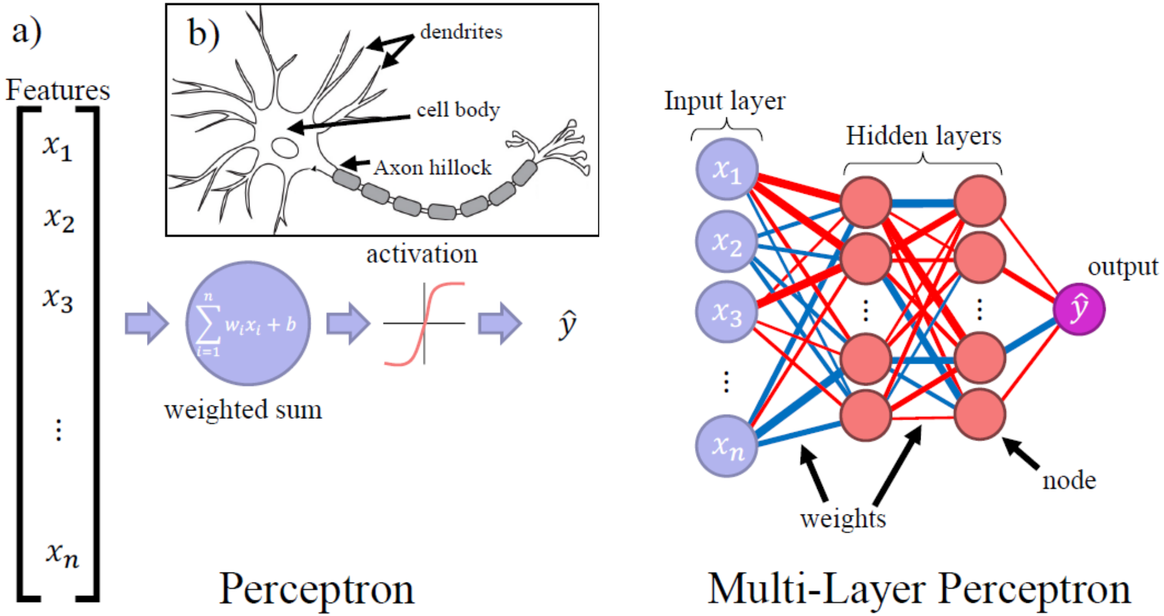
Target (fuzzy-logic):

warning categories

ii
31
33
34
36
38
40
42
46
95
48
61
62
66

My ANN learning process

My ANN learning process



Chase et al., 2023

How good is my prediction?
→ Loss function:

$$f(x, w) = (\hat{y}_t - \hat{y}_p)^2$$

How can I improve it?
→ Loss function gradient:

$$\nabla_w f(x, w) = \cdots$$

→ Gives *sensitivity* to w i.e. *direction* for nudging w so that the loss will be smaller

My ANN learning process

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

Install User Guide API Examples Community More Go

Prev Up Next

scikit-learn 1.4.1
Other versions

Please **cite us** if you use the software.

1.17. Neural network models (supervised)

Warning: This implementation is not intended for large-scale applications. In particular, scikit-learn offers no GPU support. For much faster, GPU-based implementations, as well as frameworks offering much more flexibility to build deep learning architectures, see Related Projects.

1.17.1. Multi-layer Perceptron

Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function $f(\cdot): R^m \rightarrow R^o$ by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output. Given a set of features $X = x_1, x_2, \ldots, x_m$ and a target y, it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. Figure 1 shows a one hidden layer MLP with scalar output.

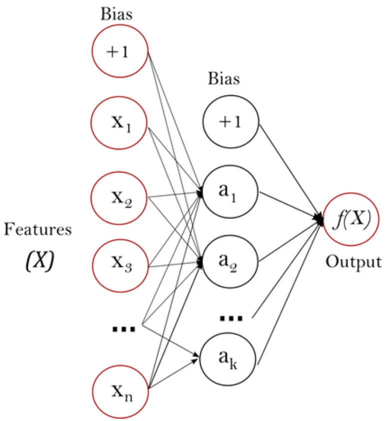


Figure 1 : One hidden layer MLP.

Tool: scikit learn

- Free / open source @ GitHub
- Commercially usable
- Python (based on Numpy, Scipy)
- Runs on CPU (not on GPU)
- Not meant for large-scale problems

My ANN learning process

How do I set up my ANN?

My ANN learning process

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

How do I set up my ANN?

I need a „toy problem“ to figure out…

My ANN learning process

My „toy problem“ (MLPfog):

- fog detection
- synthetic random data:

```
T, Td, FF
```

- simple physical model:

```
if T == Td and FF < 0.8
     fog = 1
else
     fog = 0
```

- ML model (ANN/MLP scikit-learn) to learn what the physical model does (single class classification)

My ANN learning process

My „toy problem“ (MLPfog):

- fog detection
- synthetic random data:

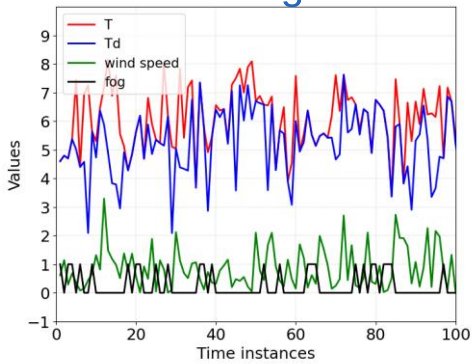
```
T, Td, FF
```

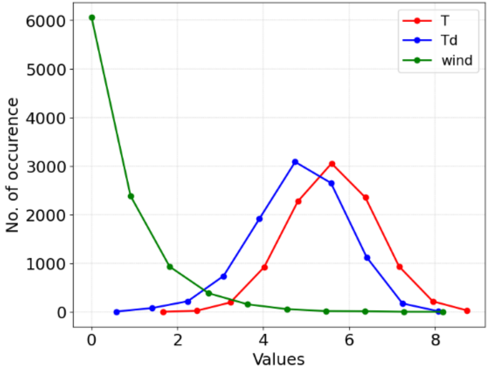
- simple physical model:

```
if T == Td and FF < 0.8
     fog = 1
else
     fog = 0
```

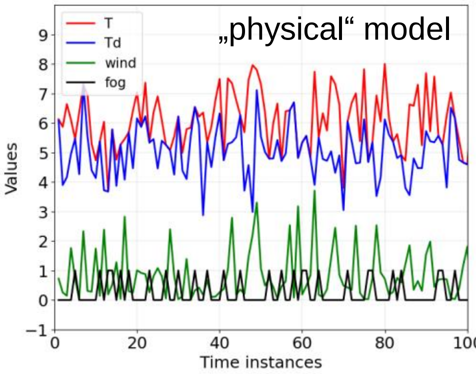
- ML model (ANN/MLP scikit-learn) to learn what the physical model does (single class classification)

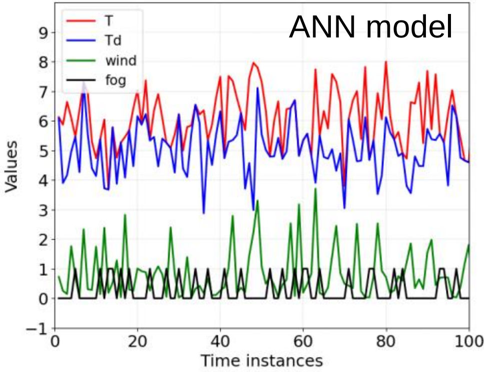
Training set





Validation set





My ANN learning process

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

Verification

	observation		
estimate		Yes	No
	Yes	a	b
	No	c	d

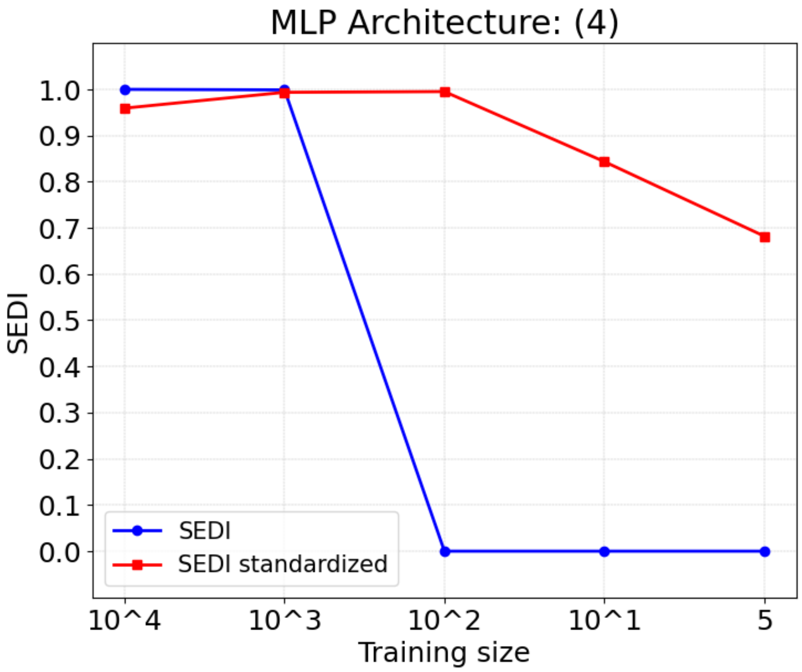
Probability Of Detection: $$\mathrm{POD} = \frac{a}{(a+c)}$$

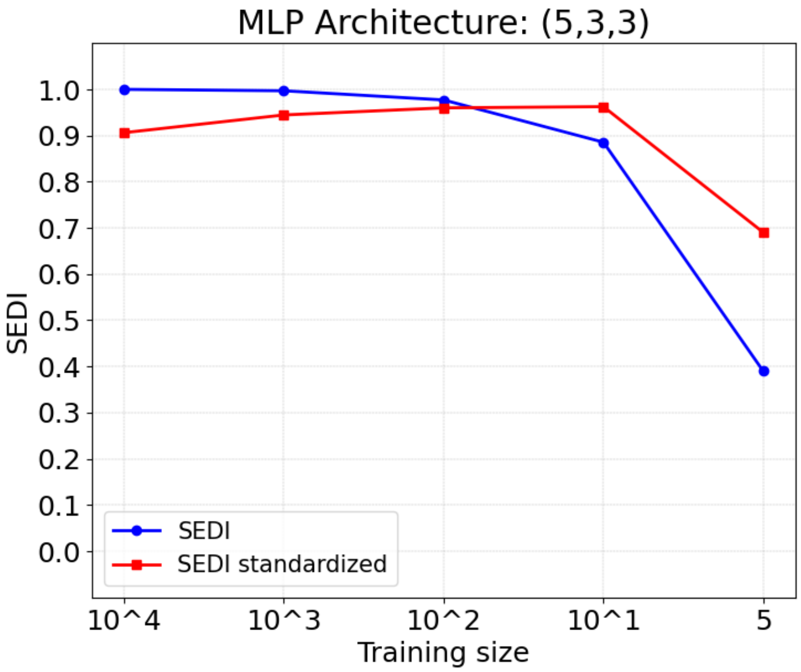
False Alarm Rate: $$\mathrm{FAR} = \frac{b}{(b+d)}$$

Symmetric Extremal Dependency Index: $$\mathrm{SEDI} = \frac{\ln(\mathrm{FAR}) - \ln(\mathrm{POD}) - \ln(1-\mathrm{FAR}) + \ln(1-\mathrm{POD})}{\ln(\mathrm{FAR}) + \ln(\mathrm{POD}) + \ln(1-\mathrm{FAR}) + \ln(1-\mathrm{POD})}$$

My ANN learning process

Question: How important standardization w.r.t. the training size?





My ANN learning process

How do I set up my ANN?

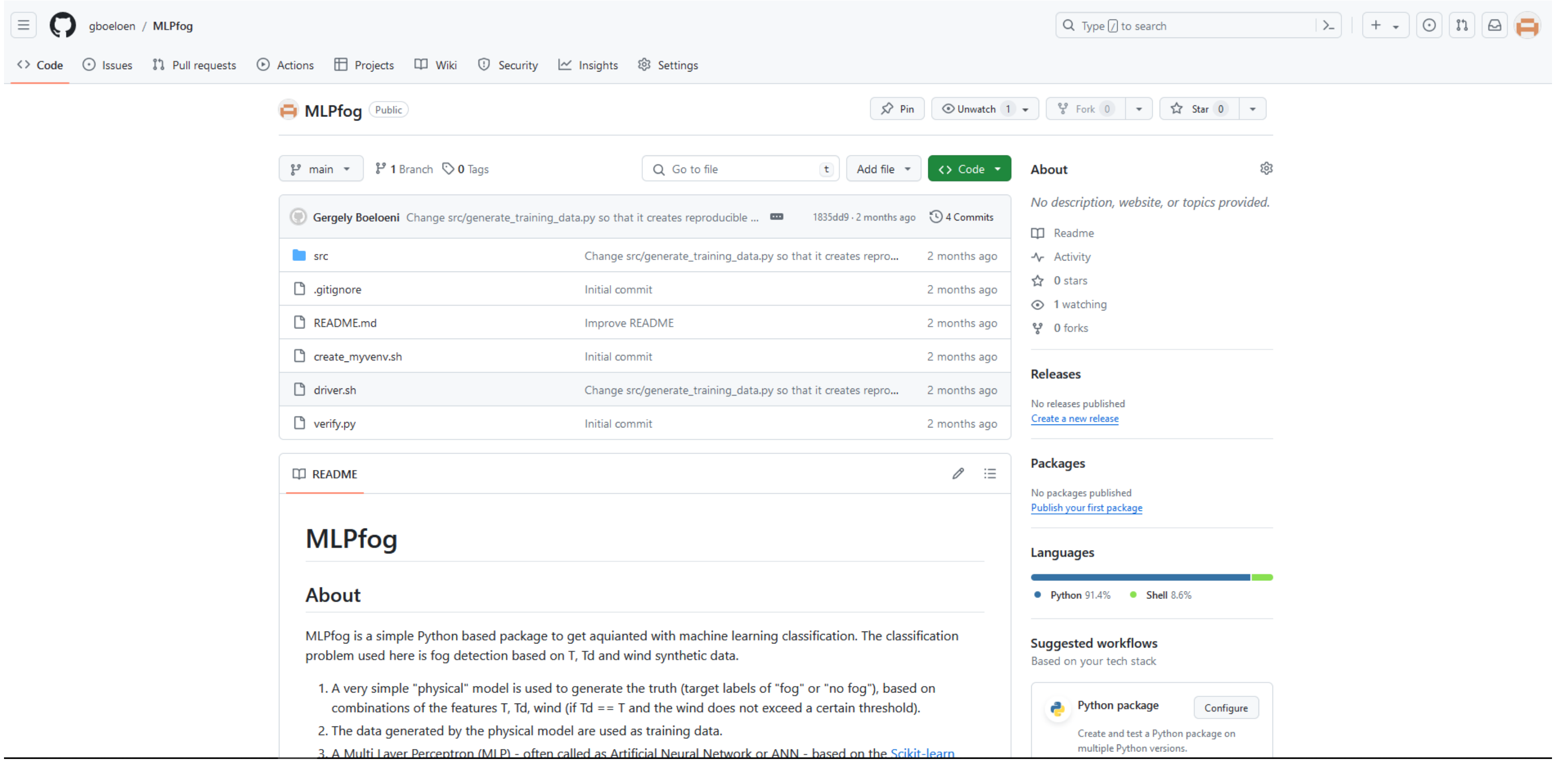
Some vague answers:

- by experience (rather „trial & error“ than a logic based approach, did not find any valid rules of thumb, lack of deeper undrestanding…)
- experience (a set-up) not transferable from small-size problems to large-size problems
- risk of missing important (or most relevant) settings is non-negligible

My ANN learning process

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

MLPfog: https://github.com/gboeloen/MLPfog



ML for emulating NowCastMIX Fuzzy-logic

ML for emulating NCM Fuzzy-logic

Features:

Meteorological state at cell centres

Variable	Source
Time (yy,mm,dd,hh,mm)	Cell occurence
Location (lat, lon)	Cell location
Wind (speed, dir)	KONRAD3D AutoRadSatW ZVF ICON-D2 V-CVF
Precipitation	Radolan RH Radolan RW
Precipitable Water	Radar (VIL) ICON-D2
Hail flag	KONRAD 3D
Cell category	KONRAD3D
Flash rate	LINET
Cloud-top height	NWC-SAF

→ **machine learning: classification problem** →

Target (fuzzy-logic):

warning categories

ii
31
33
34
36
38
40
42
46
95
48
61
62
66

ML for emulating NCM Fuzzy-logic

Features:

Meteorological state at cell centres

Variable	Source
Time (yy,mm,dd,hh,mm)	**Cell occurence**
Location (lat, lon)	**Cell location**
Wind (speed, dir)	**KONRAD3D AutoRadSatW ZVF ICON-D2 V-CVF**
Precipitation	**Radolan RH Radolan RW**
Precipitable Water	**Radar (VIL) ICON-D2**
Hail flag	**KONRAD 3D**
Cell category	**KONRAD3D**
Flash rate	**LINET**
Cloud-top height	**NWC-SAF**

machine learning: classification problem

Target (fuzzy-logic):

warning categories

ii
31
33
34
36
38
40
42
46
95
48
61
62
66

New w.r.t. MLPfog:

- 10+ target labels → ***multiclass classification***
- larger amount & variety of features
- computationally more expensive (training)

ML for emulating NCM Fuzzy-logic

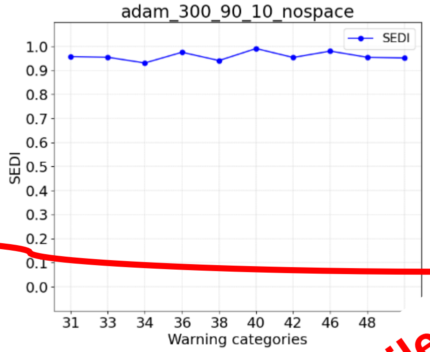
Multiclass classification with scikit learn

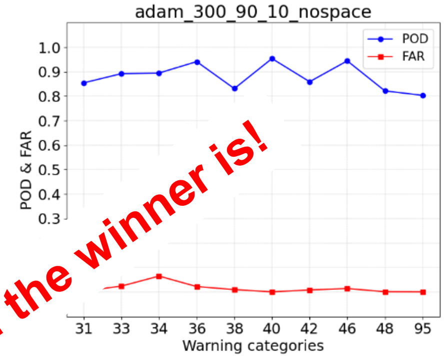
- **Inherently multiclass:**
  - naive_bayes.BernoulliNB
  - tree.DecisionTreeClassifier
  - tree.ExtraTreeClassifier
  - ensemble.ExtraTreesClassifier
  - naive_bayes.GaussianNB
  - neighbors.KNeighborsClassifier
  - semi_supervised.LabelPropagation
  - semi_supervised.LabelSpreading
  - discriminant_analysis.LinearDiscriminantAnalysis
  - svm.LinearSVC (setting multi_class="crammer_singer")
  - linear_model.LogisticRegression (setting multi_class="multinomial")
  - linear_model.LogisticRegressionCV (setting multi_class="multinomial")
  - neural_network.MLPClassifier
  - neighbors.NearestCentroid
  - discriminant_analysis.QuadraticDiscriminantAnalysis
  - neighbors.RadiusNeighborsClassifier
  - ensemble.RandomForestClassifier
  - linear_model.RidgeClassifier
  - linear_model.RidgeClassifierCV
- **Multiclass as One-Vs-One:**
  - svm.NuSVC
  - svm.SVC.
  - gaussian_process.GaussianProcessClassifier (setting multi_class = "one_vs_one")
- **Multiclass as One-Vs-The-Rest:**
  - ensemble.GradientBoostingClassifier
  - gaussian_process.GaussianProcessClassifier (setting multi_class = "one_vs_rest")
  - svm.LinearSVC (setting multi_class="ovr")
  - linear_model.LogisticRegression (setting multi_class="ovr")
  - linear_model.LogisticRegressionCV (setting multi_class="ovr")
  - linear_model.SGDClassifier
  - linear_model.Perceptron
  - linear_model.PassiveAggressiveClassifier

ML for emulating NCM Fuzzy-logic

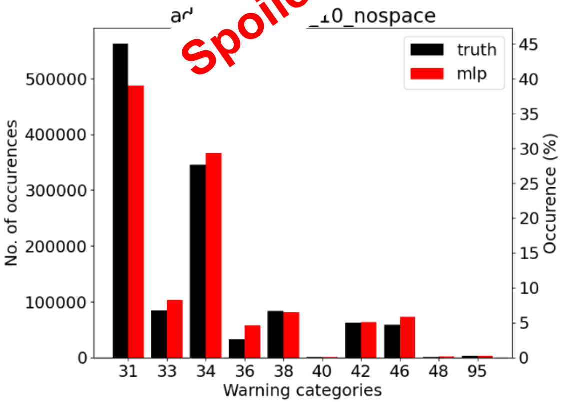
Multiclass classification with scikit learn

- **Inherently multiclass:**
  - naive_bayes.BernoulliNB
  - tree.DecisionTreeClassifier
  - tree.ExtraTreeClassifier
  - ensemble.ExtraTreesClassifier
  - naive_bayes.GaussianNB
  - neighbors.KNeighborsClassifier
  - semi_supervised.LabelPropagation
  - semi_supervised.LabelSpreading
  - discriminant_analysis.LinearDiscriminantAnalysis
  - svm.LinearSVC (setting multi_class="crammer_singer")
  - linear_model.LogisticRegression (setting multi_class="multinomial")
  - linear_model.LogisticRegressionCV (setting multi_class="multinomial")
  - neural_network.MLPClassifier
  - neighbors.NearestCentroid
  - discriminant_analysis.QuadraticDiscriminantAnalysis
  - neighbors.RadiusNeighborsClassifier
  - ensemble.RandomForestClassifier
  - linear_model.RidgeClassifier
  - linear_model.RidgeClassifierCV
- **Multiclass as One-Vs-One:**
  - svm.NuSVC
  - svm.SVC.
  - gaussian_process.GaussianProcessClassifier (setting multi_class = "one_vs_one")
- **Multiclass as One-Vs-The-Rest:**
  - ensemble.GradientBoostingClassifier
  - gaussian_process.GaussianProcessClassifier (setting multi_class = "one_vs_rest")
  - svm.LinearSVC (setting multi_class="ovr")
  - linear_model.LogisticRegression (setting multi_class="ovr")
  - linear_model.LogisticRegressionCV (setting multi_class="ovr")
  - linear_model.SGDClassifier
  - linear_model.Perceptron
  - linear_model.PassiveAggressiveClassifier





Spoiler: the winner is!

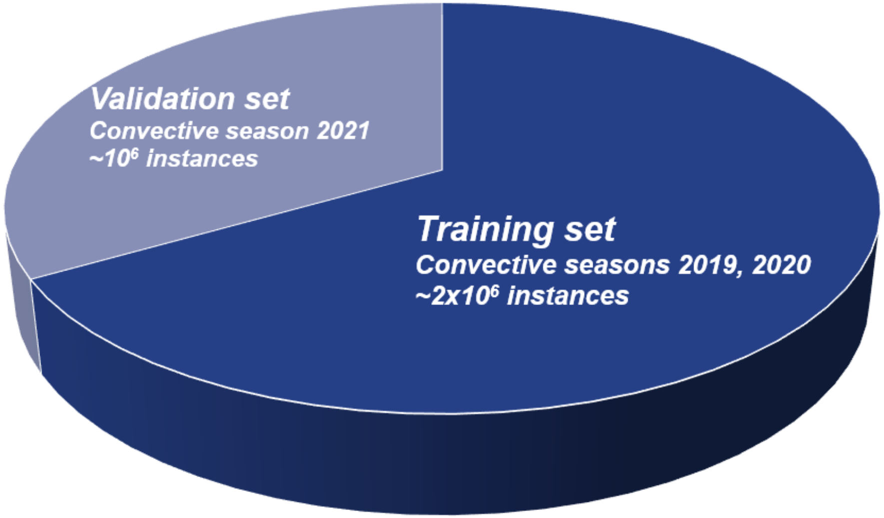


ML for emulating NCM Fuzzy-logic

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

Experimental setup

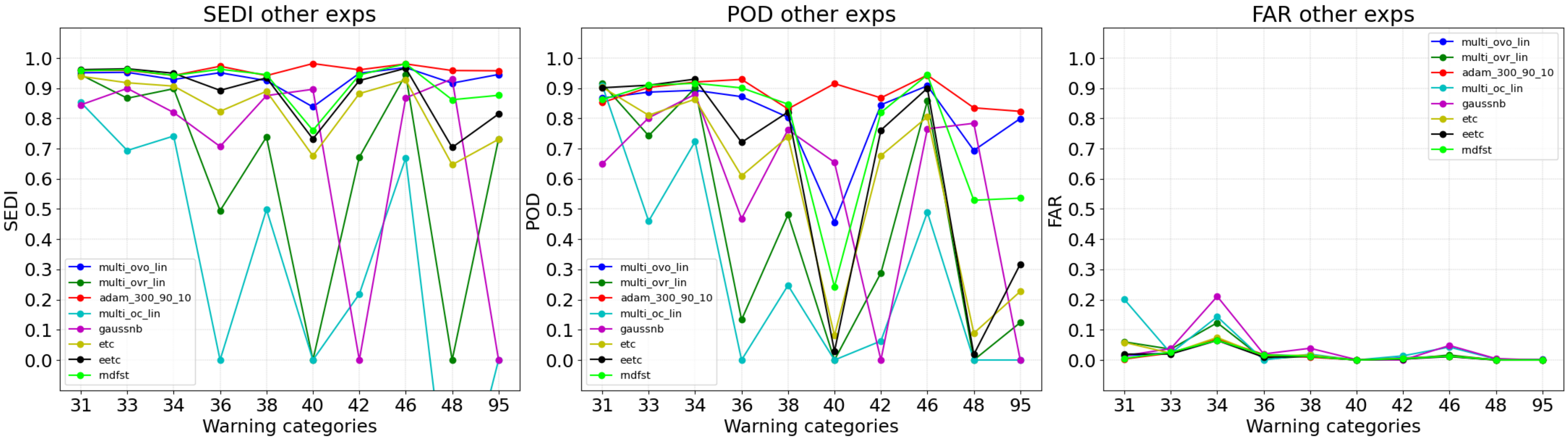
- offline tests (100% Python - not yet in the NowCastMIX code)
- separate training & validation set
- verification of each (10) warning category via SEDI (POD & FAR)
- check distribution of warning categories
- check distribution of missed events for each warning category



ML for emulating NCM Fuzzy-logic

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

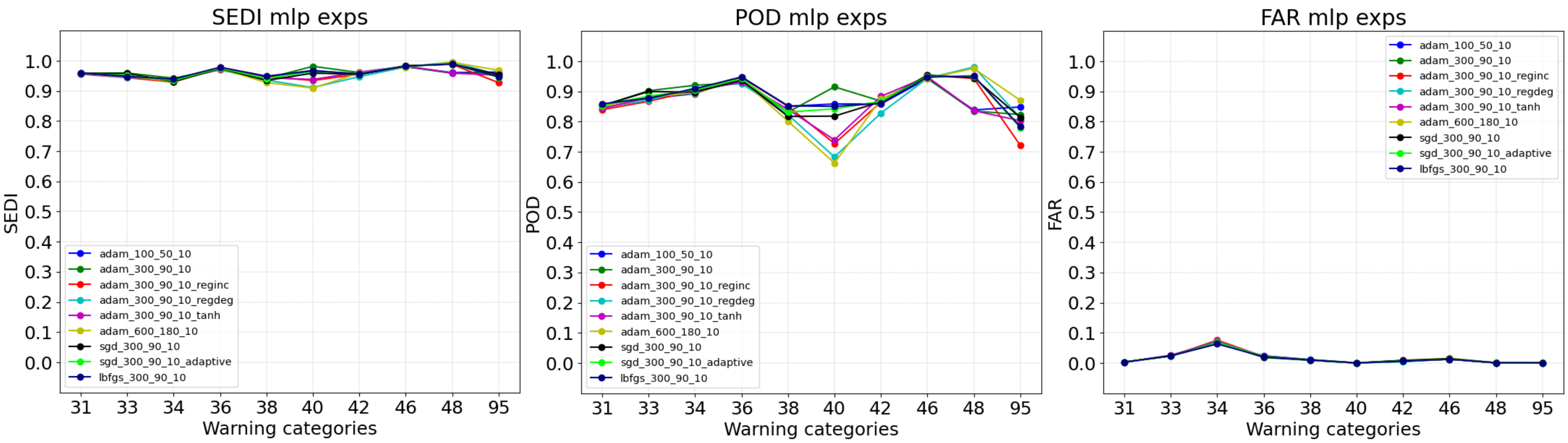
Sensitivity to ***ML method*** ***(ANN softmax, svm, decision tree, random forest, GaussNB, etc.)***



ML for emulating NCM Fuzzy-logic

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

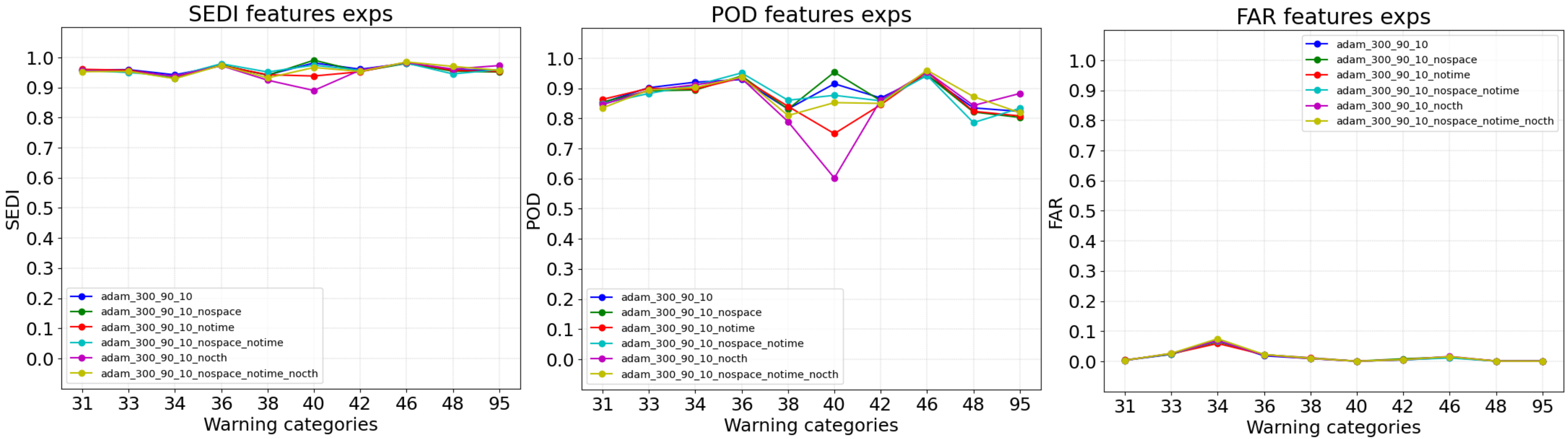
Sensitivity to ***ANN (MLP) setup***



ML for emulating NCM Fuzzy-logic

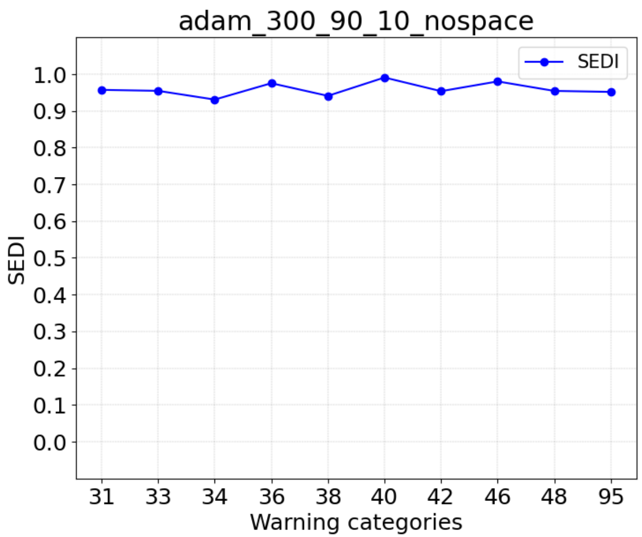
Deutscher Wetterdienst
Wetter und Klima aus einer Hand

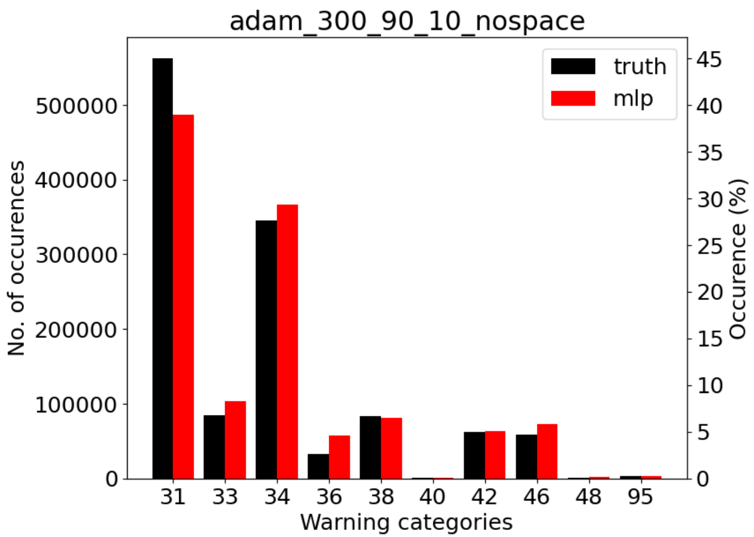
Sensitivity to ***feature set***



ML for emulating NCM Fuzzy-logic

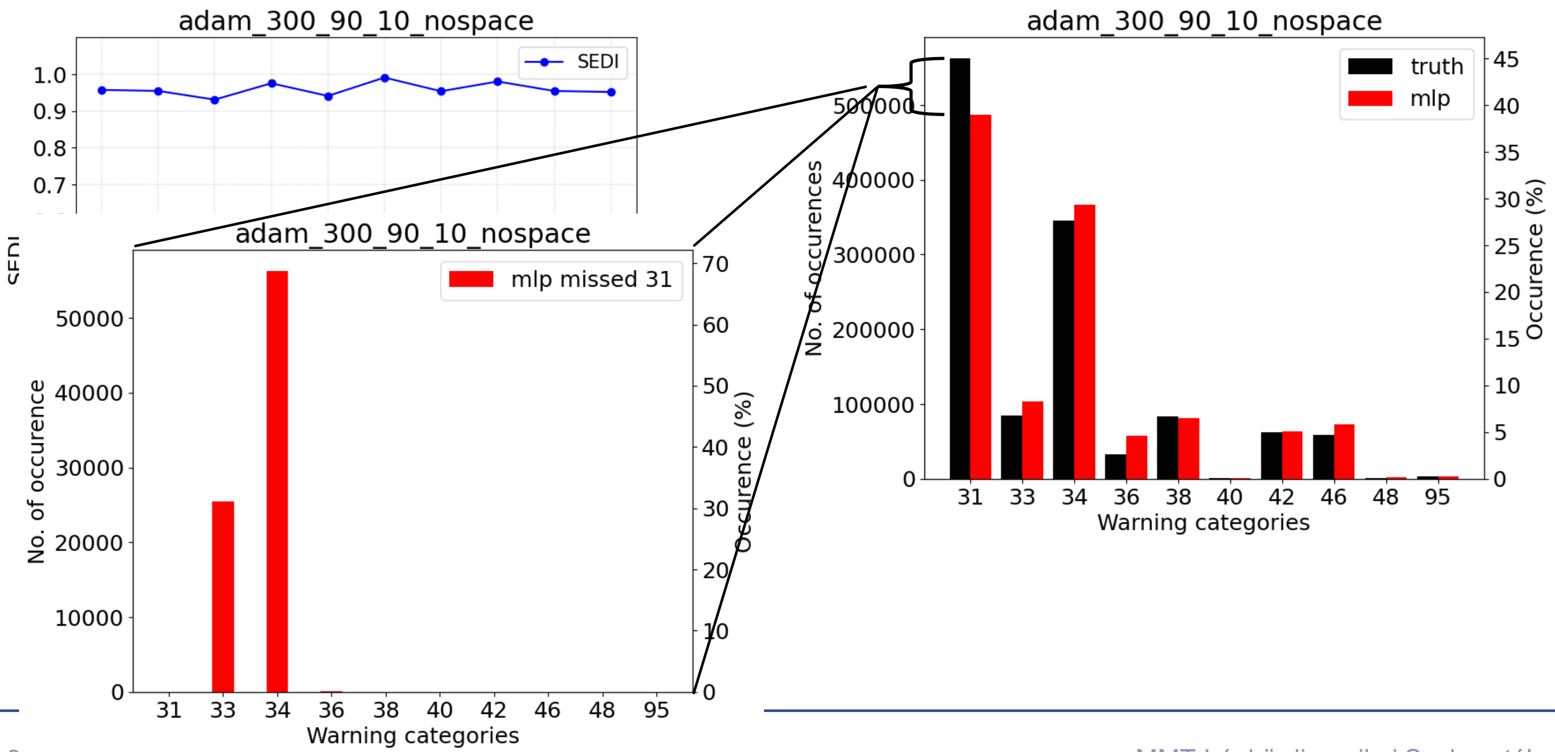
So, the winner is! `(solver: adam, activ. func.: relu, arch=300,90,10, alpha=10`$^{-3}$ `, etc)`





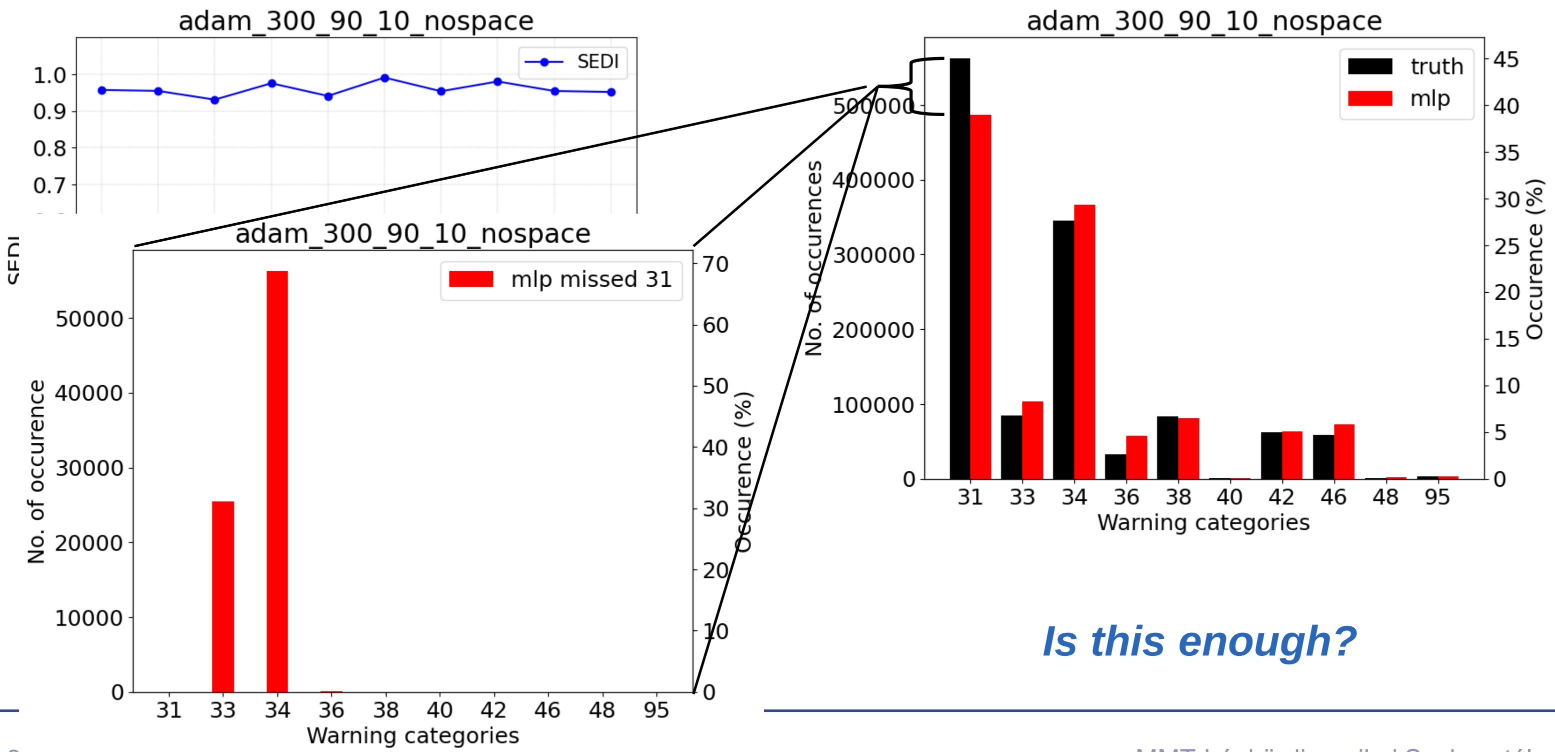
ML for emulating NCM Fuzzy-logic

So, the winner is! `(solver: adam, activ. func.: relu, arch=300,90,10, alpha=10`$^{-3}$ `, etc)`



ML for emulating NCM Fuzzy-logic

So, the winner is! `(solver: adam, activ. func.: relu, arch=300,90,10, alpha=10`$^{-3}$ `, etc)`

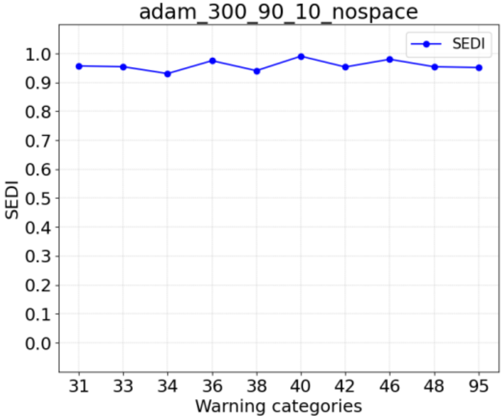


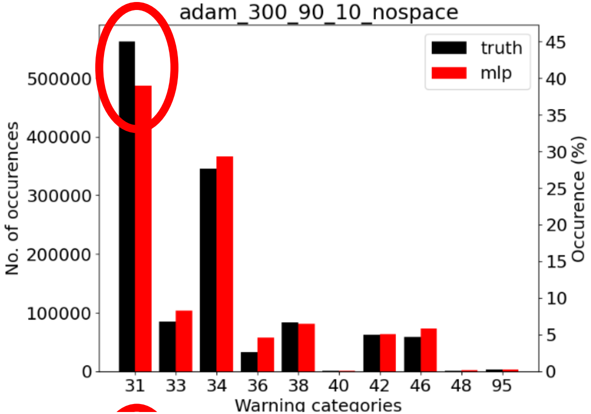
Is this enough?

ML for emulating NCM Fuzzy-logic

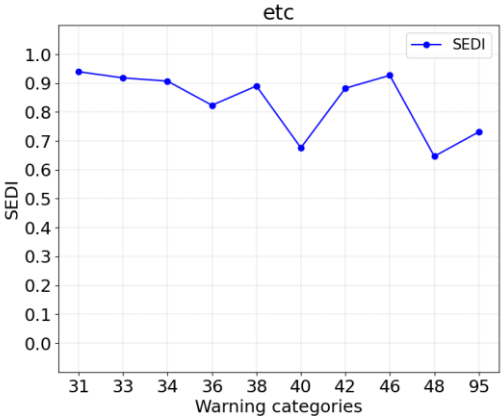
SEDI vs Distribution?

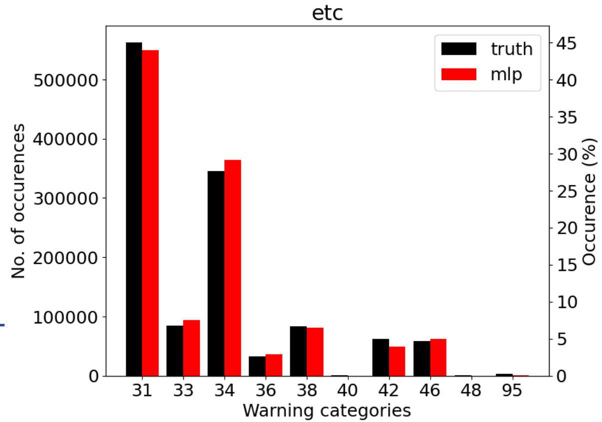
ANN





Extra Tree Classifier
(decision tree)

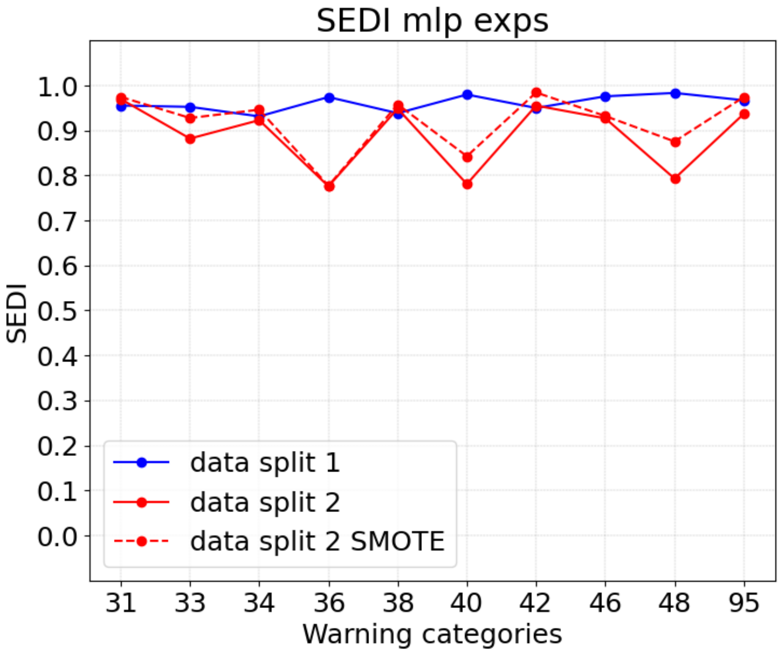




ML for emulating NCM Fuzzy-logic

Deutscher Wetterdienst
Wetter und Klima aus einer Hand

Not enough robust?



Reasons & solutions(?)

- Very much imbalanced data set, i.e. $f_{cat40} = 10^5 * f_{cat31}$
  - Even more data needed!
  - Real data: probably not enough / expensive
  - SMOTE (Synthetic Minority Oversampling Technique) → first attempt did not help
- Optimal set-up not yet found?
  - Hyperparameter tuning

Discussion & Outlook

Discussion & Outlook

- To me, using an ANN still feels like a „black box“ → lot to learn!

- ANN for warning category classification:
  - not a „low-hanging fruit“ → more data needed / hyperparameter tuning
  - no real ground truth (no warning category „observations“) → fuzzy-logic best we have → no perpective to improve
  - a nice playground but not convienced about continuing

- Probably the way forward: improve forecast step
  - So far: stretching based on extrapolation by wind
  - Plan: Machine Learning (CNN, cell-based ←→ grid based?)

- PyTorch or TensorFlow (CPU & GPU) instead of Scikit-Learn (only CPU)?