



The added value of machine-learning methods in forecasting wind turbine icing

LUKAS KUGLER

MASTERSEMINAR

2. VORTRAG, 8.5.2019

Betreuer: Ass.-Prof. Mag. Dr. Manfred Dorninger, Dipl.-Ing. Dr. Lukas Strauss

Überblick

Thema: Vorhersage der Vereisung von Windkraftanlagen

- Erhöht Planbarkeit der Produktion (senkt Ausfallkosten)
- Ermöglicht bessere Personalplanung der Technik
- Ermöglicht frühere Wiederinbetriebnahme und dadurch Mehrproduktion

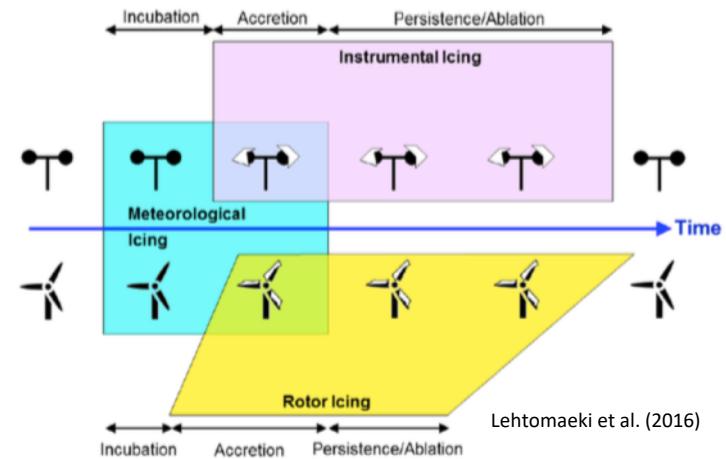
Im Rahmen des Projekts wurde ein WRF-Ensemble für zwei Winter gerechnet.

Ziel: verbesserte Vorhersage mittels Post-Processing

Vorhergesagt wird z.B.:

Morgen zwischen 9 und 10 UTC

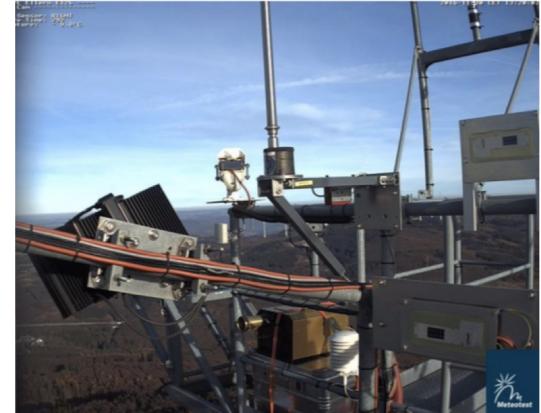
- Liegen Vereisungsbedingungen vor? (25%)
- Springen die Vereisungssensoren an? (23%)
- Wird sich sichtbar Eis ansetzen? (9%)



Welche Informationen sind verfügbar?

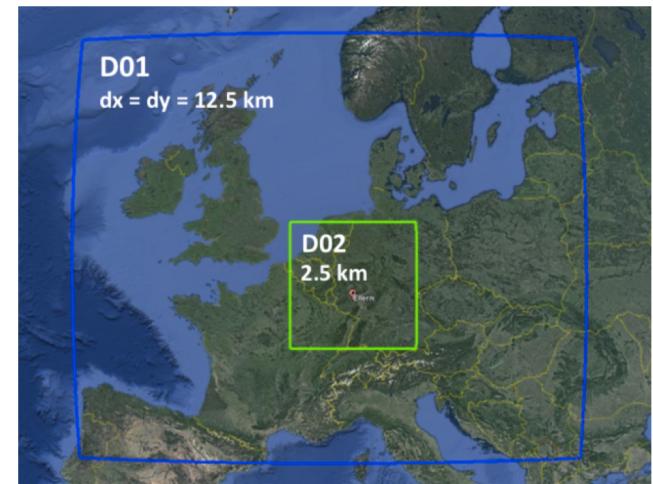
Beobachtungsdaten

- Numerisch:
Temperatur, Relative Feuchte, Wind, Sichtweite
- Kategorisch:
 - Vereisungsgrad der Sensoren
 - Binär: Eisansatz in der letzten Stunde (optisch bestimmt)



Vorhersagedaten:

- WRF mit 11 Ensemble Membern
 - Modellvariablen (T, p, ff, q_vapor, q_cloud, ...)
 - Räumliche Auflösung 2.5 km
 - Zeitliche Auflösung stündlich bis max. 72h
- Verschiedene Kombinationen von Eingangs faktoren zu testen



Ziele und Fragestellungen

Hypothese: Machine-Learning-Methoden
gefüttert mit allen Informationsquellen
= bessere Vorhersagen

Informationsquelle	berücksichtigt
Historische Vorhersagen und zugehörige Beobachtungen	systematische Modellfehler
Ensemble-Läufe	Abschätzung der Unsicherheit
Messwerte der letzten Stunden	Aktuelle Modellfehler
Sensibilität auf False Alarms, ... (C/L Verhältnis)	Optimierung auf den Nutzer

Ziele und Fragestellungen

Hypothese: Machine-Learning-Methoden
gefüttert mit allen Informationsquellen
= bessere Vorhersagen

Fragestellungen

- Bieten Machine-Learning Methoden einen Zusatznutzen gegenüber klassischem Postprocessing (z.B.: logistischer Regression)?
- Worin liegen die Stärken der jeweiligen Modelle?
- Wie hoch ist der Zusatznutzen / Unsicherheit?
- Wodurch kommt die Verbesserung zustande?
 - Nutzung von Ensembledaten
 - Zeitliche Trägheit des Vorhersagefehlers (Nowcasting)
 - Nichtlineare Effekte, ‚komplexere Funktionen‘ der Eingangsfunktionen

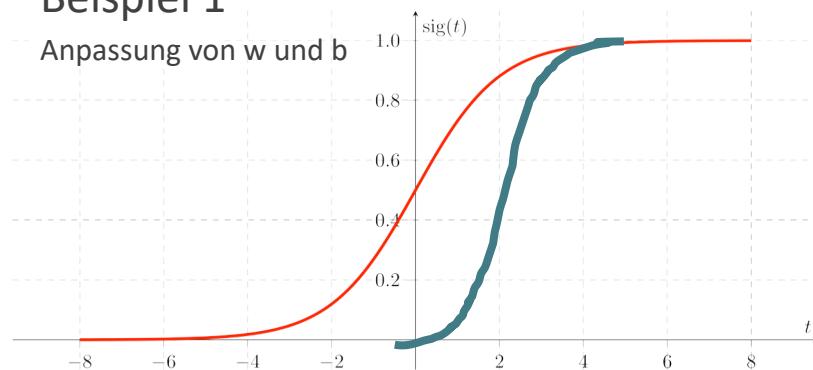
Methoden: Neuronales Netz

- engl.: Artificial Neural Net
 - nur entfernt mit biologischen Neuronen verwandt

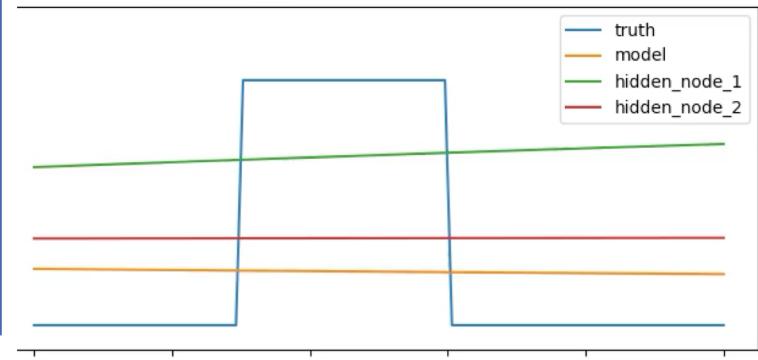
Struktur

Beispiel 1

Anpassung von w und b



Beispiel 2: Summe zweier Sigmoid



Methoden: Neuronales Netz

Struktur

1 Schicht: $y = f \left(\sum_i w_i x_i + b \right)$

2 Schichten: $= f \left(\sum_i \tilde{w}_i (f(\sum_i w_i y_i + b)) + \tilde{b} \right)$

...

Optimierung: Gradientenverfahren „Backpropagation“

Wie muss man die Parameter w verändern, um den Fehler E zu minimieren?

→ - $\frac{\partial E}{\partial w_i}$ d.h. Verändert man die Gewichte in diese Richtung, wird der Fehler kleiner.

- Wie quantifiziert man die Abweichung von Vorhersage und ‚Wahrheit‘?
- Welche Größe soll optimiert werden?

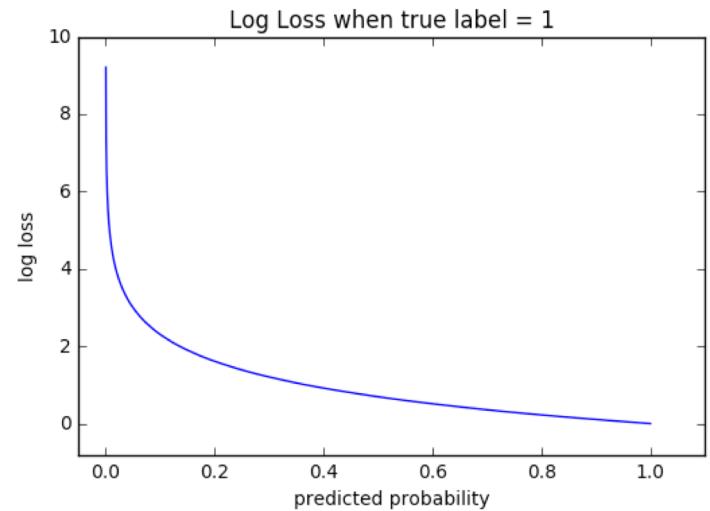
Methoden: Neuronales Netz

Welche Größe soll optimiert werden?

- normalverteilte Fehler: Summe der Fehlerquadrate $E = \sum_i (\tilde{y}_i - y_i)^2$
- binäre Variable: *Cross-Entropy*
für Abweichungen der Wahrheit (0/1) von vorhergesagter Wahrscheinlichkeit (0-1)
 - konfidente Falsch-Vorhersage wird überproportional bestraft

$$E = \sum_i -(y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$$

Wahrheit Vorhersage



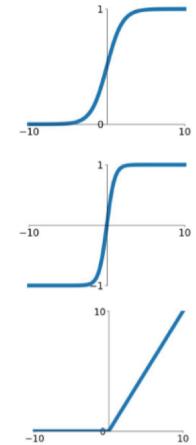
Methoden: Neuronales Netz

Typ: *Multilayer Perceptron* (Werbos, 1974)

- Python Modul *Keras*
- Optimierung des Netzes
 - Gradientenverfahren mit stochastischem Anteil *RMSprop* (Tieleman et.al. 2012)

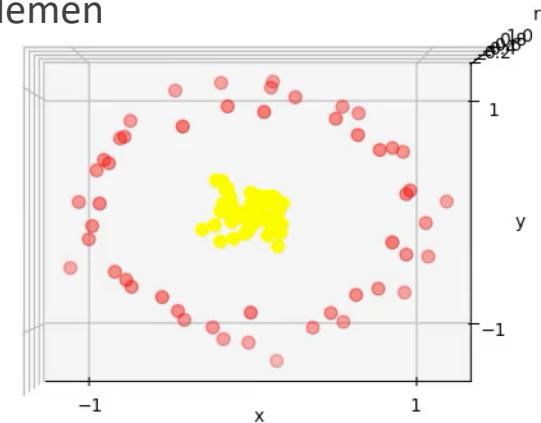
Welche Entscheidungen bleiben?

- Architektur
 - Wieviele Knoten und Schichten
 - Welche Aktivierungsfunktion (Sigmoid, Tanh, ReLU, ...)
 - Welche Metrik (MSE, Cross-Entropy)
- Preprocessing
 - Mittelwert und Varianz standardisieren
 - Dimensionalität reduzieren, Prediktoren weglassen?
- Details im Algorithmus
 - Wieviele Samples werden auf einmal verwendet um den Gradienten zu berechnen (*batch_size*)
 - Wieviele Iterationen werden berechnet
 - Konvergenzkriterium (Vorzeitiger Iterationsstopp *Early Stopping*) – schützt vor Overfitting
 - Zufälliges Auslassen von Neuronen *Dropout* – schützt vor Overfitting



Methoden: Weitere Modelle

- **Support Vector Klassifikation** bereits im ersten Vortrag erklärt
 - funktioniert besonders bei hochdimensionalen Problemen
-> bei vielen Eingangsfaktoren (= Prediktoren)
- **Entscheidungsbaum-Modelle** *Decision-Trees*
 - *Bagging und Boosting Methoden, ...*
- **GAM Generalized Additive Models**
 - Erweiterung der linearen Regression
 - z.B. Spline-Regression



Lin. Reg:

$$y = \beta_0 + \sum_i^p \beta_i x_i$$

GAM:

$$y = \beta_0 + \sum_i^p f_i(x_i)$$

Wert der Prognosen

Quantifizierung der Einsparungen im Cost-Loss Entscheidungsmodell
mittels Potential Economic Value PEV

PEV = f(Hits, Correct Rejections,
Missed Events, False Alarms)

PEV = 1 perfekt

PEV = 0 klimatologisch

		Yes-Event	No-Event
Yes-Fcst	Yes-Event	C	C
	No-Event	L	0
Kostenmatrix			

Voraussetzung ist

- False Alarm kostet **C** Euro
- Missed Event kostet **L** Euro

z.B.: Miss 2x so teuer wie False Alarm
→ C/L Verhältnis = 0,5

→ Beispiel Gewittervorhersage

Wert der Prognosen

PEV = f(Hits, Missed Events, False Alarms, Correct Rejections)

PEV = 1	perfekt
PEV = 0	klimatologisch

Beispiel: Gewittervorhersage

- Wahrscheinlichkeit für Gewitter liegt bei 25%

- 1) Ja, Gewitterwarnung
→ ist in 75% falsch
- 2) Nein, keine Gewitterwarnung
→ ist in 25% falsch

	Richtig	Falsch
in 25% Hit		in 75% False Alarm
in 75% Correct Rejection		in 25% Miss

Warnen oder nicht?

- wenn C/L = 0.2, dann Option 1
- wenn C/L = 0.3, dann Option 2

→ Wert der Vorhersage abhängig von C/L

Vergleich der Modelle

Zusätzlich zu Kontroll-Lauf und Ensembles:

- Time-Lagged Ensemble Tägliche Vorhersagen bis 72 h → 3-fach Überschneidung bis 24 h

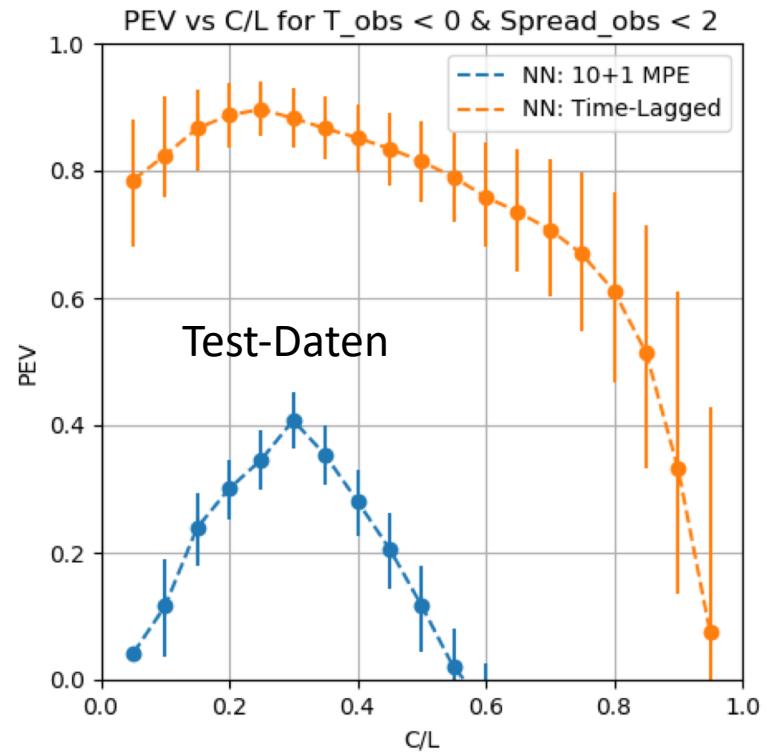
Prediktand: Vereisung

(Temperatur < 0 °C & Taupunkt < 2 K)

Vorhersagestunden 3 - 20

Predictoren:

- NWP-Vorhersageläufe
der letzten 3 Tage (*Time-Lagged*)
- Multi-Physik Ensemble (*MPE*)



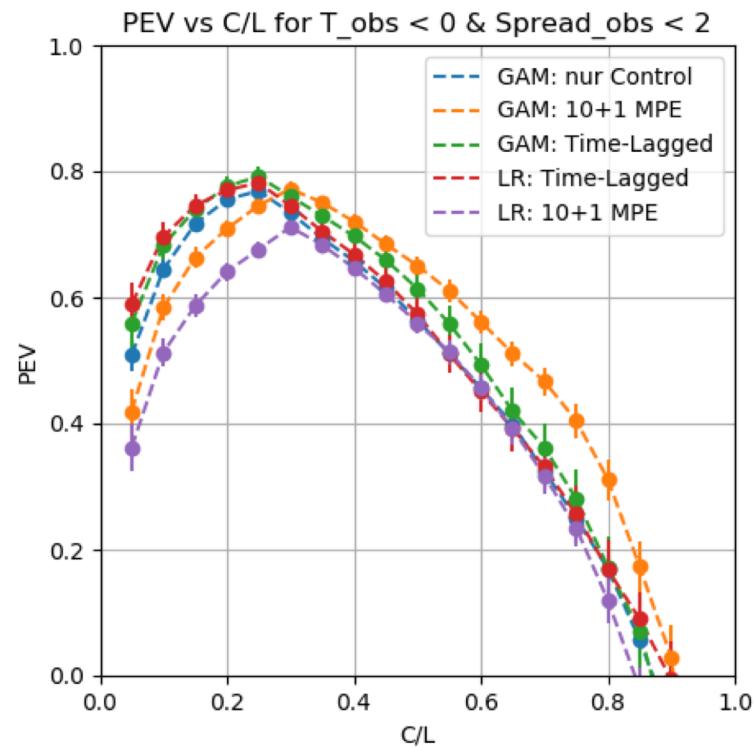
Vergleich der Modelle

Prediktand: Vereisung (Temperatur < 0 °C & Taupunkt < 2 K)

Vorhersagestunden 3 – 20

Ergebnis:

- Je nach C/L reicht Time-Lagged-Ensemble statt MP-Ensemble aus
- NN: Prediktorselektion unzureichend Training schwierig
- GAM: besser als Logistische Regression robuster als Neuronales Netz



Neuronale Netze: Welchen Einfluss hat Tiefe?

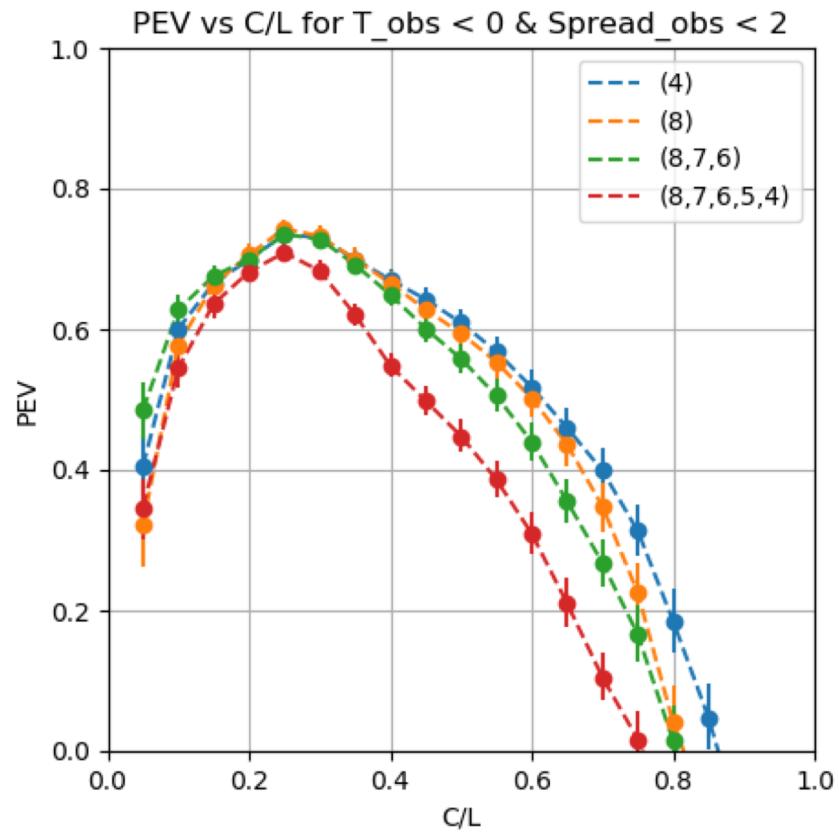
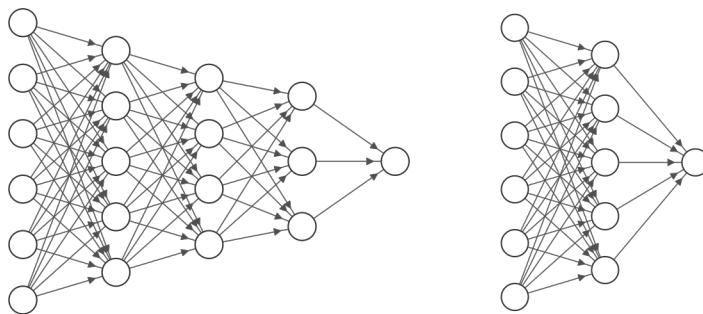
Prediktand: Vereisung (Temperatur < 0 °C & Taupunkt < 2 K)

Predictoren:

- NWP-Vorhersageläufe
der letzten 3 Tage

Variation:

- Anzahl der Schichten

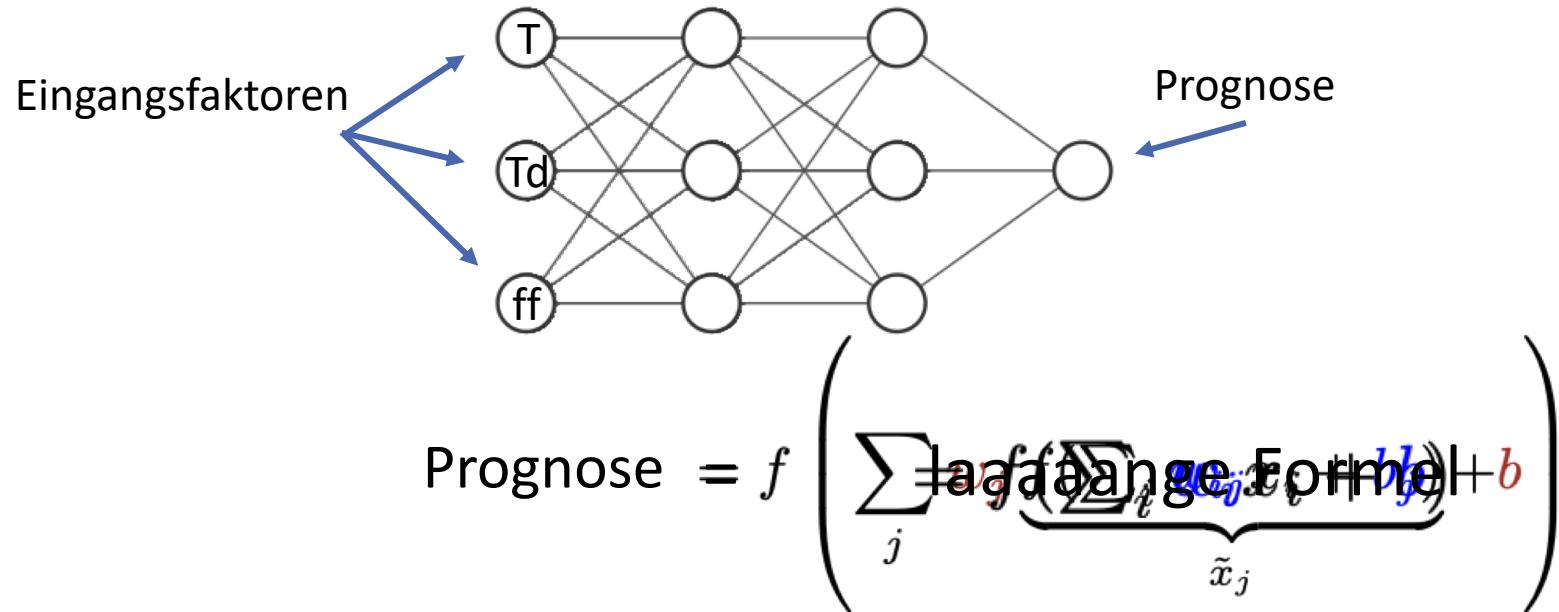


Erkenntnisse und Tipps

- Wieviel Leistung durch komplexe ML Methoden „gehoben“ werden kann, ist von
 - der verfügbaren Datenmenge und
 - wahrscheinlich von der Komplexität der modellierten Relation abhängig
- Als Start ist logistische Regression (mit PCA) zu empfehlen
Liefert stabile Ergebnisse als Basis für Weiteres
- Danach Entscheidungsbäume und GAM
- Neuronale Netze
 - Parameter-Tuning ungemein wichtig, es existiert keine klare Vorgangsweise
 - Gute Ergebnisse nur bei wenigen 100 freien Parametern bei 5-15000 Samples
 - Schlechte Ergebnisse auch bei stark redundanten Prediktoren (Ensembles) ohne PCA

Erkenntnisse und Tipps

- **Oft: „Für neuronale Netze braucht man große Datensätze“**
 - praktisch ist er dann zu kurz, wenn das Modell keine stabilen Ergebnisse liefert (*Overfitting*)
 - Reduktion der freien Parameter immer möglich d.h. weniger Schichten, weniger Knoten



Erkenntnisse und Tipps

- **Oft: „Für neuronale Netze braucht man große Datensätze“**
 - praktisch ist er dann zu kurz, wenn das Modell keine stabilen Ergebnisse liefert (*Overfitting*)
 - Reduktion der freien Parameter immer möglich d.h. weniger Schichten, weniger Knoten
- **Je besser die Ergebnisse, umso mehr Vorsicht ist angebracht**

Wichtig für valide Ergebnisse:

Sorgfältige Trennung von Trainings- und Validierungsdaten

- Keine Validierung auf den Trainingsdaten
- Kein Training mit den Validierungsdaten
 - wenn Modell nach Ansicht der Ergebnisse optimiert wird etwa mit Einstellungen am neuronalen Netz
→ viele Möglichkeiten sich selbst zu täuschen

The added value of ML methods in forecasting wind turbine icing

Lukas Kugler

Masterseminar

2. Vortrag, 8.5.2019

Referenzen:

- Davis (2014), Icing Impacts on Wind Energy Production, PhD thesis
- Jolliffe & Stephenson (Eds.). (2003). Forecast verification: a practitioner's guide in atmospheric science.
- Hsieh (2009). Machine learning methods in the environmental sciences: Neural networks and kernels.
- Breiman (2001): Statistical modeling: The two cultures. Statistical science

Die aktuelle sowie die erste Seminar-Präsentation,
und weitere Informationen sind auf
lkugler.github.io/mthesis verfügbar



universität
wien



imgw
Institut für Meteorologie
und Geophysik