Interpretable deep learning for timeseries problems

S Lotz [1,2,3]   M Davel [2,3]
C Grant [2]   T Theunissen [2]   A de Villiers [2]

[1]SANSA
Hermanus
https://sansa.org.za/

[2]MUST Deep Learning
North-West University, Hermanus
https://engineering.nwu.ac.za/must

[3]National Institute of Theoretical and Computational Sciences
(NITheCS)
Stellenbosch
http://nithecs.ac.za

LMAG Workshop, 21-25 Aug 2023
JHU APL

SANSA SOUTH AFRICAN NATIONAL SPACE AGENCY

science & innovation
Department:
Science and Innovation
REPUBLIC OF SOUTH AFRICA

NITheCS
National Institute for
Theoretical and Computational Sciences

NWU®
NORTH-WEST UNIVERSITY
NOORDWES-UNIVERSITEIT
YUNIBESITI YA BOKONE-BOPHIRIMA

Outline

1. Define time-series problem
2. List some ways to interpret DL models
3. Show example from SW-GMD case
4. Describe our toolkit (under development)
5. Roadmap for further development

Our definition of a time-series regression problem

Problem

Approximate the unknown function $\boxed{y(t) = f(\mathbf{X}, t)}$ given data sets of the

- output $y(t) \in \mathbb{R}$ at time $t \in [1, 2, \ldots, n]$, and
- k inputs $\mathbf{X} \in \mathbb{R}^{k \times n}$ over n time instances
  - $\mathbf{X} = [\mathbf{x_1}(\tau_1), \mathbf{x_2}(\tau_2), \ldots, \mathbf{x_k}(\tau_k)]$
  - Each τ_i has some specific relation to t,
    - constant lag: $\tau_i = t - a$
    - identity: $\tau_i = t$
    - correlates with X: $\tau_i = x_1/x_2$

Assumptions

1. Causality $y(t) \not\leftarrow y(t+a)$
2. Inter-dependence $I(x_i, x_j) > 0$
3. Non-stationarity of dependencies
4. Delay functions $\tau_i(t)$ are unknown, not constant
5. Temporal feedback: $x_i(t) = y(t-a), a > 0$

Example

$$\boxed{y(t) = \mathbf{x_1}(\tau_1) + \mathbf{x_2}^2(\tau_2) + \mathbf{x_3}^6(\tau_3)}$$

Our definition of a time-series regression problem

Problem

Approximate the unknown function

$\boxed{y(t) = f(\mathbf{X}, t)}$ given data sets of the

- output $y(t) \in \mathbb{R}$ at time $t \in [1, 2, \ldots, n]$, and
- k inputs $\mathbf{X} \in \mathbb{R}^{k \times n}$ over n time instances
- $\mathbf{X} = [\mathbf{x_1}(\tau_1), \mathbf{x_2}(\tau_2), \ldots, \mathbf{x_k}(\tau_k)]$
- Each τ_i has some specific relation to t,
  - constant lag: $\tau_i = t - a$
  - identity: $\tau_i = t$
  - correlates with X: $\tau_i = x_1/x_2$

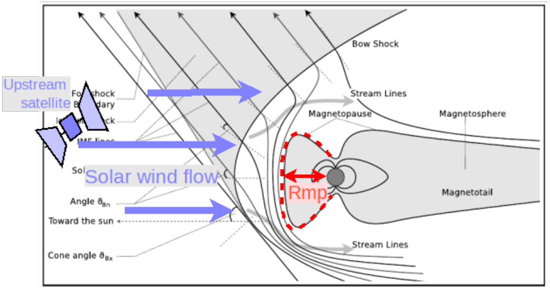
Assumptions

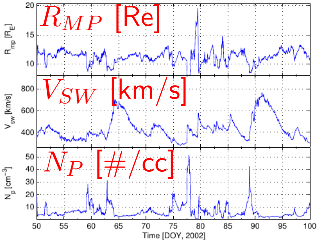
1. Causality $y(t) \not\leftarrow y(t + a)$
2. Inter-dependence $I(x_i, x_j) > 0$
3. Non-stationarity of dependencies
4. Delay functions $\tau_i(t)$ are unknown, not constant
5. Temporal feedback: $x_i(t) = y(t - a), a > 0$

Example

$$\boxed{y(t) = \mathbf{x_1}(t) + \mathbf{x_2}^2(t - 2) + \mathbf{x_3}^6(\lfloor x_2/x_1 \rfloor + 3)}$$

A simple example from space physics





Problem details

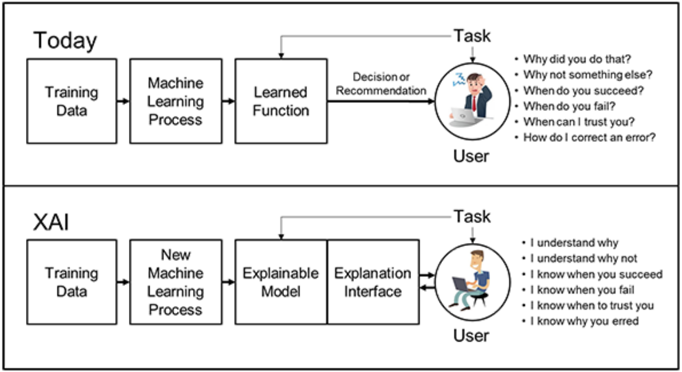
- Magnetosphere size R_{MP} determined by solar wind pressure $\boxed{P \sim V_{SW}^2 N_P}$
- Varies wildly with P
- τ_1 is dynamic, depends on *input* V_{SW}
- V_{SW} and N_P measured upstream by satellite (distance S)

Magnetopause standoff distance R_{MP}

$$\begin{aligned} R_{MP}(t) &\approx 110.2[V_{SW}(\tau_1)^2 N_P(\tau_1)]^{-1/6} \\ \tau_1(t) &= t - S(t)/V(t) \end{aligned}$$

$S(t)$: Distance to satellite at time t [km]
$V_{SW}(t)$: Solar wind speed at t [km/s]

What do we mean by *Interpretability*?

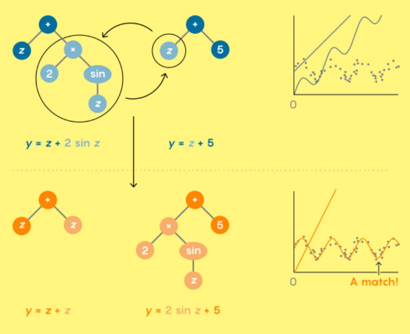


- Data explanation: first prize. Just explain the problem by understanding the data
- Global explanation: Directly interpretable models
- Local: Evaluate individual predictions to guide understanding
- Post hoc (global): apply to black box model
- Post hoc (local): Run various samples throught the black box to gain understanding

Directly Interpretable: Symbolic Regression

Symbolic regression

- Start with a "library" of simple equations
- and input (X) / output (y) data set $y = F(X)$
- Brute force combinatorics to
- Iteratively build up the model G
- $Y = G(X) + \epsilon \approx F(X)$



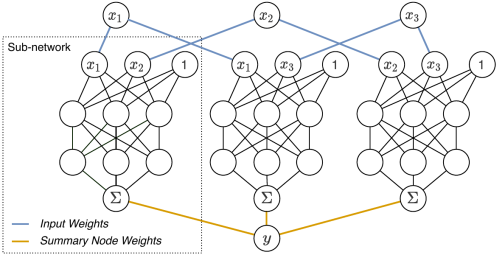
Some recent examples

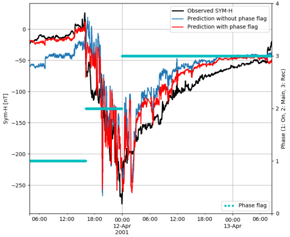
- Derivation of Newton's law of gravitation from planet trajectories [1]
- Complex fluid models [2]
- Discovery of novel cell divison triggers [3]
- AI-Feynman project solved 100 equations from Feynman lectures automatically [4]

Directly Interpretable: Designed for Interpretation

- Pairwise Net [5] Decompose net in to pairwise sub-nets
- Enable variable feature-ranking and feature-interactions
- Non-linear and interpretable (a step up from low-order regression)
- computationally expensive
- See also *GAMI-Net* [6]

PWNet: Summary node weights quantify attribution of a pair; net structure combines all possible pairs





Post-hoc: Feature Attribution I

Post-hoc, model agnostic, methods to assign attribution score to each input (i.e. apply to trained net).

Shapley values $\rightarrow$ SHAP (SHapley Additive exPlanations)

- In coalition game, players (features) compute for payout (prediction)
- Shapley values ϕ_j are fair distribution of the "payout" (ϕ) among the "players" (j)
- SHAP use this idea to create additive features based on ϕ_j:

$$\boxed{g(Z) = \phi_0 + \sum_j \phi_j Z_j}$$

- This can be local (per sample) or global (over the entire dataset):

$$\boxed{I_j = \frac{1}{n} \sum_{i=1}^{n} |\phi_j^{(i)}|}$$

- Score can be positive or negative.

Post-hoc: Feature Attribution II

Backprop-based: LRP

Layer-wise relevance propagation is decomposition of NN, redistributing output of neuron $\boxed{x_j = \max(0, \sum x_{ij} + b_j)}$ by Taylor expansion

$$\boxed{x_j = \sum \partial x_j / \partial x_i \Big|_{x_i = \tilde{x}_i} \cdot (x_i - \tilde{x}_i)}$$

around $\tilde{x}_i$ toward previous layers. Input variables (pixels) get a relevance score.

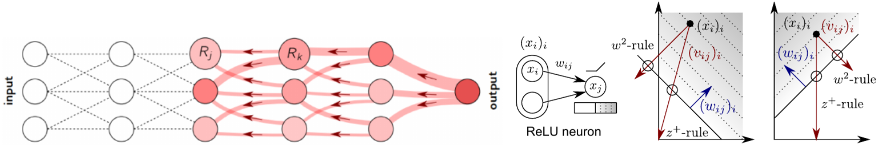
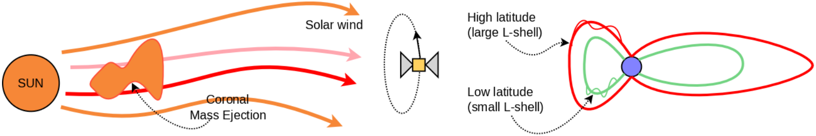


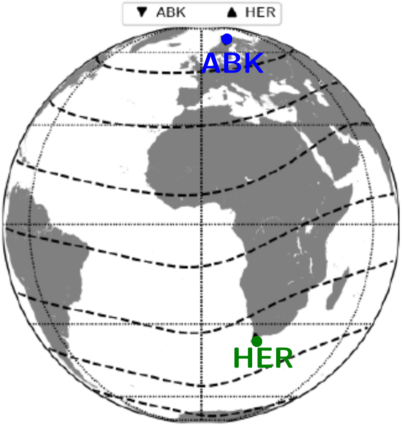
Figure: Flow of information towards input layer [7].

Intepretation of regression DL models

Current work: Solar wind → Geomagnetic Index



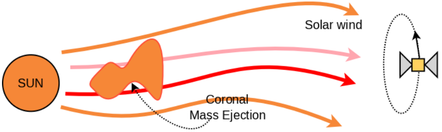
Model two regions on same dataset

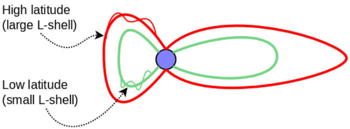


- HER (low lat, 34S)
  ABK (high lat, 68N)
- Two models:
  $eh_{ABK} = F(SW, \mathbf{w}_{ABK}) + \epsilon_{ABK}$
  $eh_{HER} = F(SW, \mathbf{w}_{HER}) + \epsilon_{HER}$[a]
- Do input feature attribution on SW parameters
  - Use DeepSHAP (based on Shapley values)
  - Are there different drivers at diff. lat?

[a] e_h induced E-field index from [Wintoft].

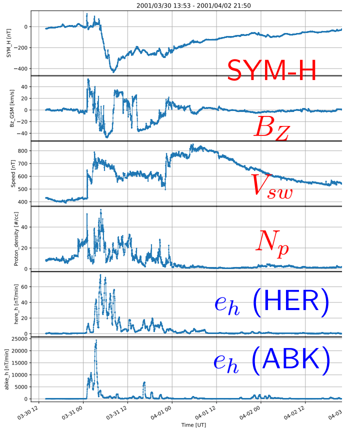
Current work: Solar wind → Geomagnetic Index





Model two regions on same dataset

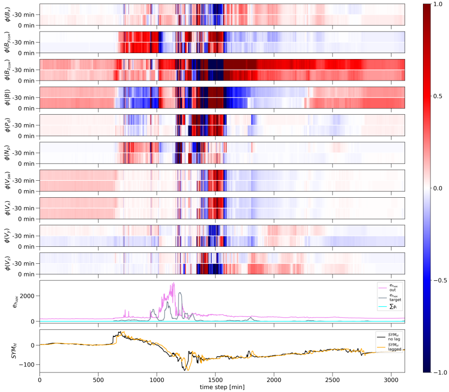
- HER (low lat, 34S)
  ABK (high lat, 68N)
- Two models:
  $eh_{ABK} = F(SW, \mathbf{w}_{ABK}) + \epsilon_{ABK}$
  $eh_{HER} = F(SW, \mathbf{w}_{HER}) + \epsilon_{HER}$[a]
- Do input feature attribution on SW parameters
  - Use DeepSHAP (based on Shapley values)
  - Are there different drivers at diff. lat?



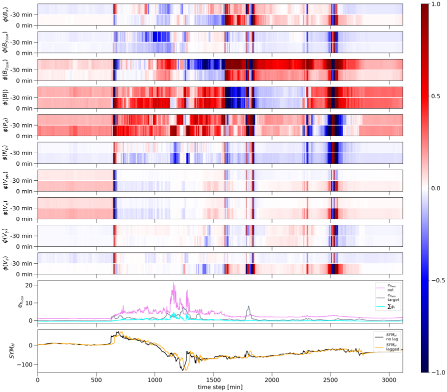
[a] e_h induced E-field index from [Wintoft].

Current work: Solar wind → Geomagnetic Index

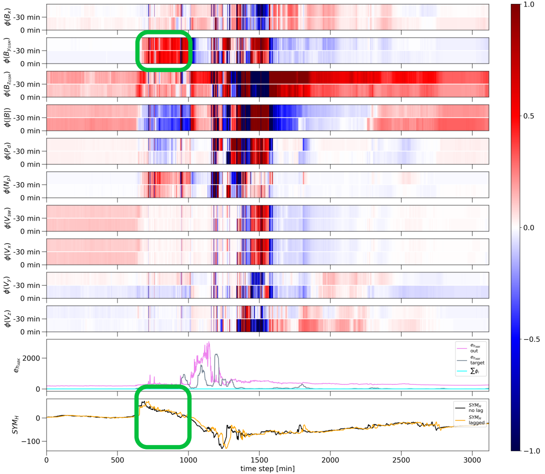
Abisko (high N lat)



Hermanus (low S lat)



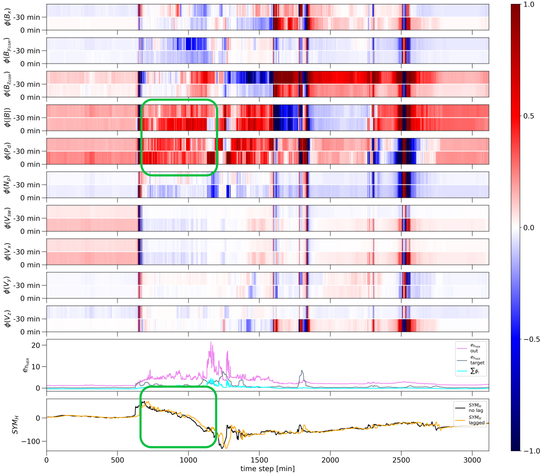
Current work: Solar wind → Geomagnetic Index



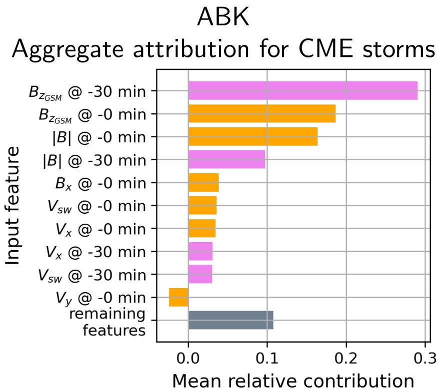
- Abisko: N high lat
- CME-driven storm (2001/08/17-18)
- B_Y postive contribution to prediction during main phase

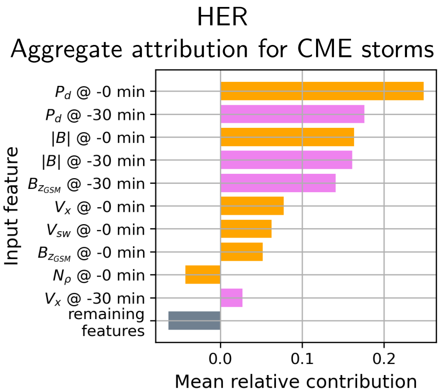
Current work: Solar wind $\rightarrow$ Geomagnetic Index

- Hermanus: S low/mid lat
- CME-driven storm (2001/08/17-18)
- $|B|$ and P_d postive contribution to prediction during main phase



Current work: Solar wind $\rightarrow$ Geomagnetic Index





- Top ranked: B_z, $|B|$, V_{SW}
- IMF orientation (Z) plays important role at **high lat**

- Top ranked: P_d, $|B|$, B_z
- Pressure plays important role at **low lat**

“know-it” toolkit

Knowledge Discovery In Timeseries

- A toolkit that allows knowledge discovery in time series data

"know-it" toolkit

Knowledge Discovery In Timeseries

- A toolkit that allows knowledge discovery in time series data
- Main tasks
  1. Prepare data
  2. Create a model of the data
  3. Interpret the model for knowledge discovery

"know-it" toolkit

Knowledge Discovery In Timeseries

- A toolkit that allows knowledge discovery in time series data
- Main tasks
  1. Prepare data
  2. Create a model of the data
  3. Interpret the model for knowledge discovery
- Codebase
  - Python with conda, numpy, matplotlib, pandas, etc.
  - Pytorch for ML
    - Model training: Darts → Pytorch Lightning
    - Interpretation: Captum

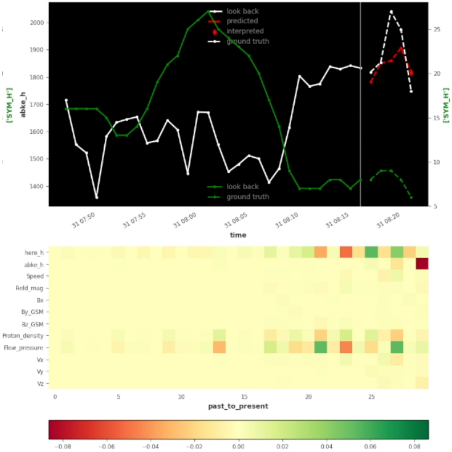
“know-it” toolkit

Knowledge Discovery In Timeseries

- A toolkit that allows knowledge discovery in time series data
- Main tasks
  1. Prepare data
  2. Create a model of the data
  3. Interpret the model for knowledge discovery
- Codebase
  - Python with conda, numpy, matplotlib, pandas, etc.
  - Pytorch for ML
    - Model training: Darts → Pytorch Lightning
    - Interpretation: Captum
- Functionality
  - Data: Multivariate numerical time series data
  - Model: Temporal convolutional network (TCN)
  - Interpretability: Deepliftshap

“know-it” toolkit

Knowledge Discovery In Timeseries



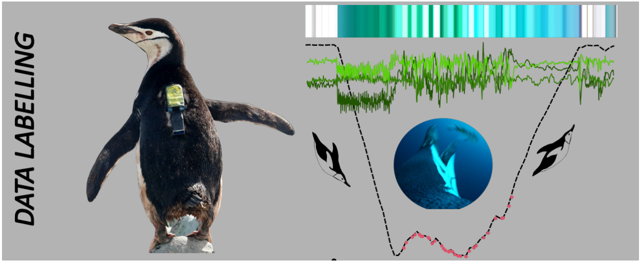
- Output: 5-minute ahead prediction of e_h
- Inputs:
  - SW parameters ($[t-30, \ldots, t-1]$ min window)
  - past values of e_h (30 min window)
- Interpretation: Attribution of each feature from past to present

“know-it” toolkit: Wider collaboration and applications I

Knowledge Discovery In Timeseries

Few interesting collaborations on the cards

1. Prey capture event prediction for free ranging Chinstrap Penguins
   - Train on hand-labelled video footage
   - Deploy on accelerometer-only data sets
   - Interpretation: distinguish capture gesture from other movements
   - Use: southern ocean krill population density studies
   - Collaboration: University of Cape Town



"know-it" toolkit: Wider collaboration and applications II

Knowledge Discovery In Timeseries

3. Predict temporal dynamics in synthetic microbial community
    - Analyse multiple species of wine yeast
    - Predict abundance
    - Understand and highlight temporal dynamics and interactions between species
    - Data exploration phase
    - Collab: University of Stellenbosch
4. Analyse inter-species dynamics in large nature reserve
    - Spatio-temporal problem
    - May need graph-NN instead of TCN
    - Preliminary research phase
    - Collab: University of Stellenbosch, SANParks

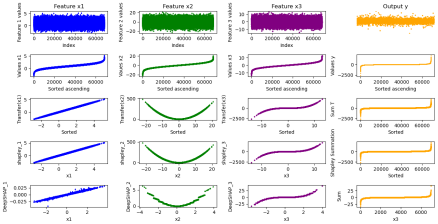
Current work: Synthetic data testbed

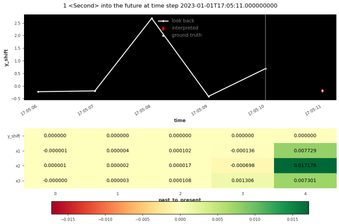
Synthetic data function creation

$$\boxed{F(x,y,z,t) = G(\mathbf{x_1}[\tau_1[t]]) \otimes H(\mathbf{x_2}[\tau_2(t)]) \otimes J(\mathbf{x_3}[\tau_3(t)])}$$

- Sample $\mathbf{x}_i$ from stat. distributions or physical data
- Select interactions $\otimes$
- Select driver functions H, G, J
- Determine time delay functions τ_j

- Test estimated attribution versus actual attribution
- SHAP vs DeepSHAP vs Shapley values
- Other sources of synthetic data exist [8] but lacks granularity





Questions?

Stefan Lotz
SANSA, Hermanus

slotz@sansa.org.za

Bibliography I

[1] Pablo Lemos, Niall Jeffrey, Miles Cranmer, Shirley Ho, and Peter Battaglia.
Rediscovering orbital mechanics with machine learning, 2022.

[2] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz.
Discovering governing equations from data by sparse identification of nonlinear dynamical systems.
Proceedings of the National Academy of Sciences, 113(15):3932–3937, 2016.

[3] Marina Uroz, Sabrina Wistorf, Xavier Serra-Picamal, Vito Conte, Marta Sales-Pardo, Pere Roca-Cusachs, Roger Guimerà, and Xavier Trepat.
Regulation of cell cycle progression by cell–cell and cell–matrix forces.
Nature Cell Biology, 20(6):646–654, Jun 2018.

[4] Silviu-Marian Udrescu and Max Tegmark.
Ai feynman: a physics-inspired method for symbolic regression.
2019.

[5] Pairwise networks for feature ranking of a geomagnetic storm model.
South African Computer Journal, 32(2):35–55, 2020.

[6] Zebin Yang, Aijun Zhang, and Agus Sudjianto.
Gami-net: An explainable neural network based on generalized additive models with structured interactions, 2020.

[7] G. Montavon, Bach S., Binder A., Samek W., and Müller K.R.
Deep taylor decomposition of neural networks.
In *ICML 2016 Workshop on Visualization for Deep Learning*, 2016.

[8] Synthetic data vault.
URL: `https://sdv.dev`, last visited: 2022/09/19.