Applied Machine Learning, Final Project 2025

Atmospheric measurements – going from balloons to satellites

Martin M. Hyldahl, Sebastian M. Nordgren, Simon M. Kjær, Thomas M. Schiøler, Vitus M. Nielsen

Overview

- Introduction
- Problem
- Motivation
- Theory
- Data and treatment
- Results
- Conclusion

Problem & motivation

- **Balloons with radiosondes (temporal, spatial & economic data gaps)**
  - Infrequent (max twice a day)
  - Expensive (~400 € per launch)
  - Labour intensive
  - Scarcities (limited to fixed launch-sites)
    - Land > Ocean
    - Middle latitudes > Tropics & Poles
    - Global North > Global South

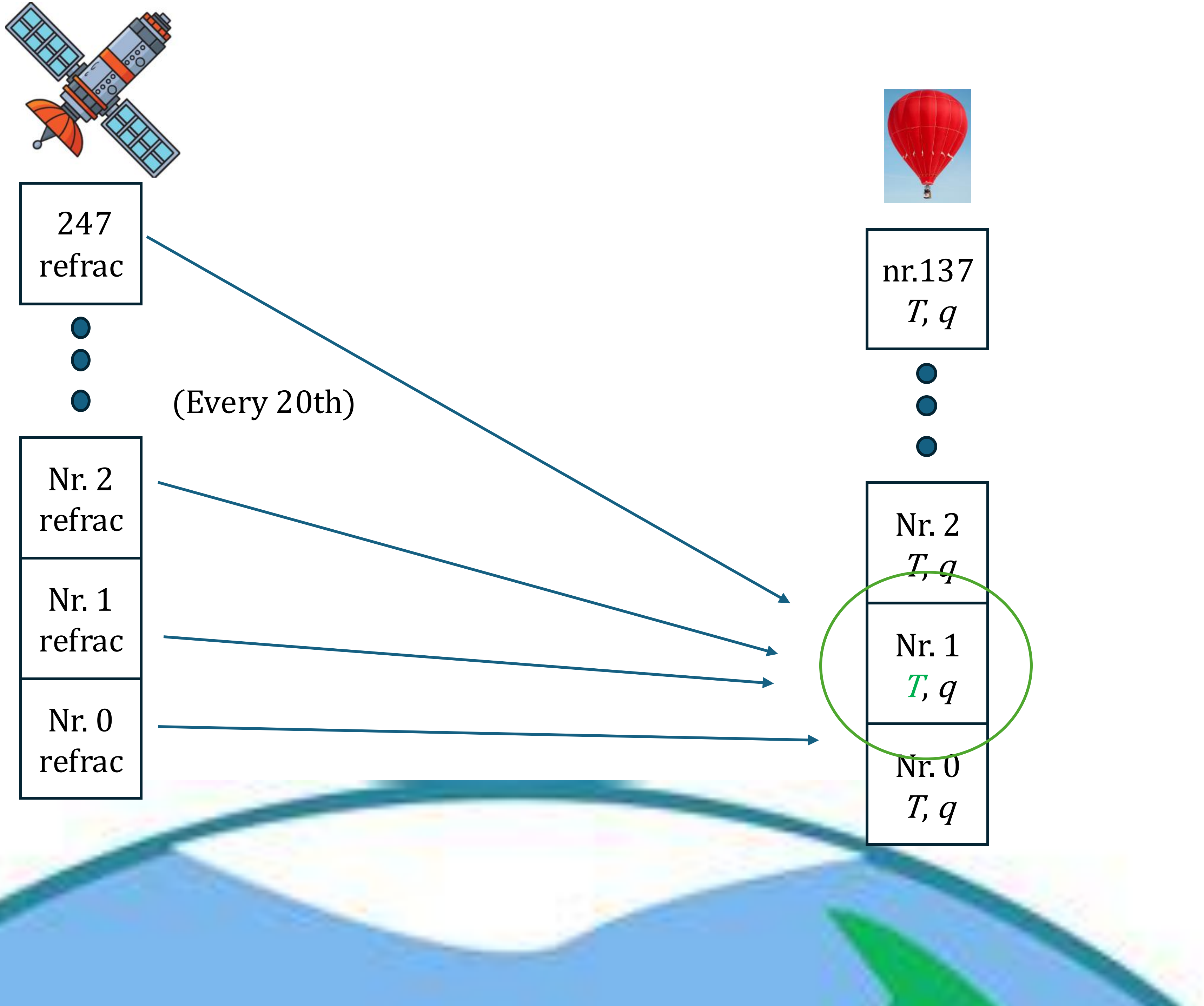
Measured quantities include:
Temperature: T
Specific humidity: q

Launch from Hamburg (28-05-2025)

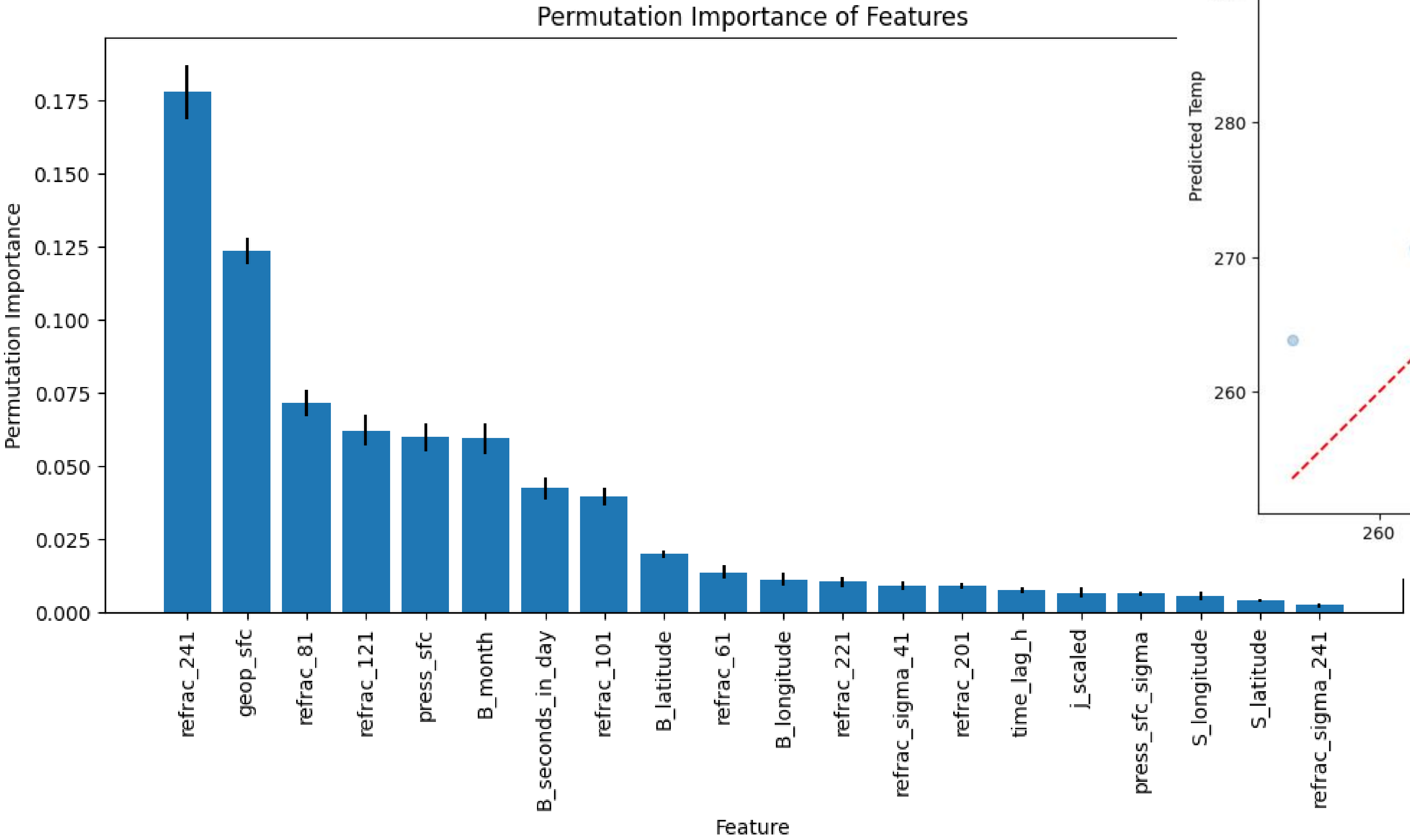
- **Satellites (radio occultation data)**
  - Automated
  - Global coverage
  - Frequent: one receiver can measure up to 6000 occultations per day: (setting and rising GNSS satellites)

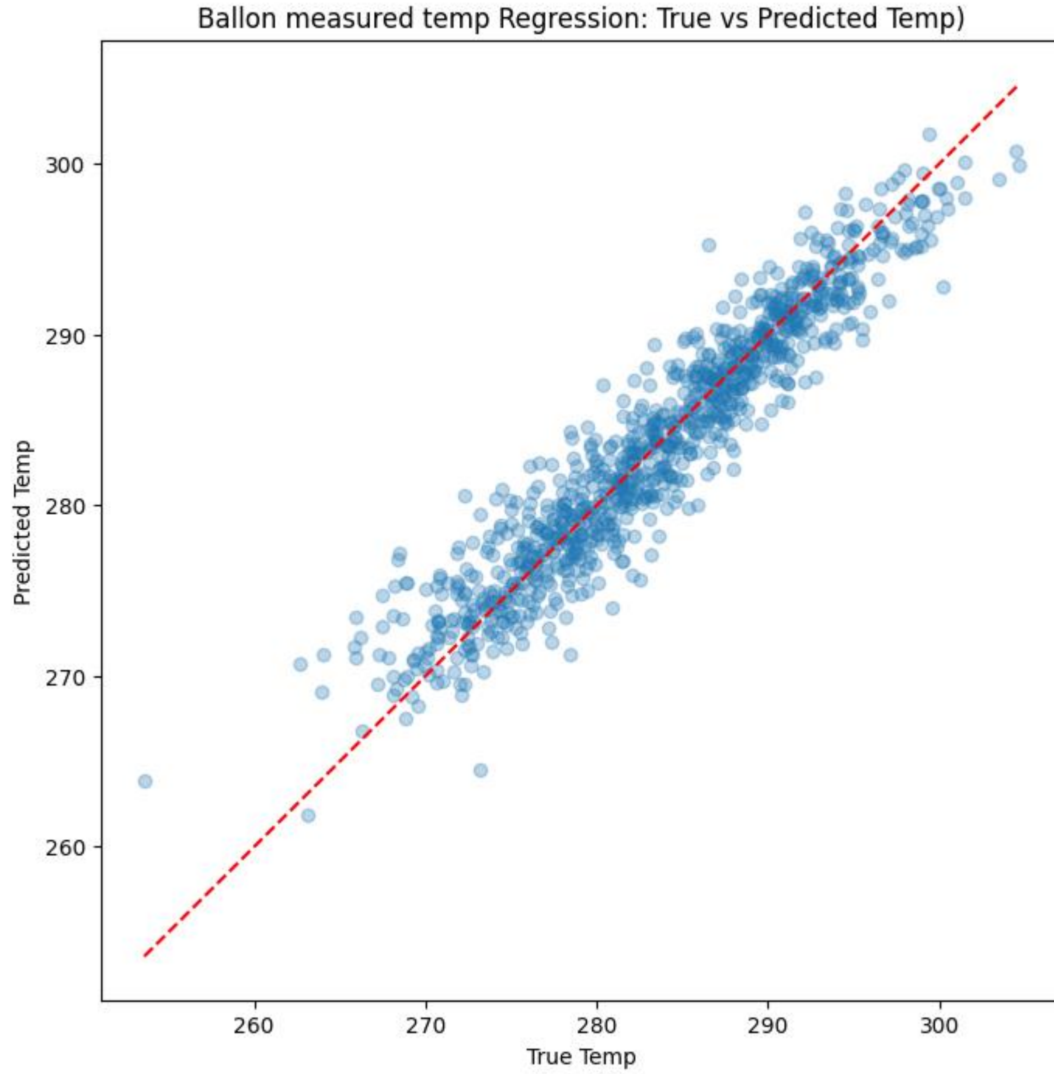
Quick n' dirty:
LGBM Regression to predict ground temperature from whole profile.

- Feature Ranking
- Performance to beat



```
Validation RMSE: 6.712
Validation R2: 0.897
```





Radio refractivity equation:

N : Refractivity
P : Pressure
T : Temperature
q : Specific humidity
n_e : Electron density
f : Radio frequency

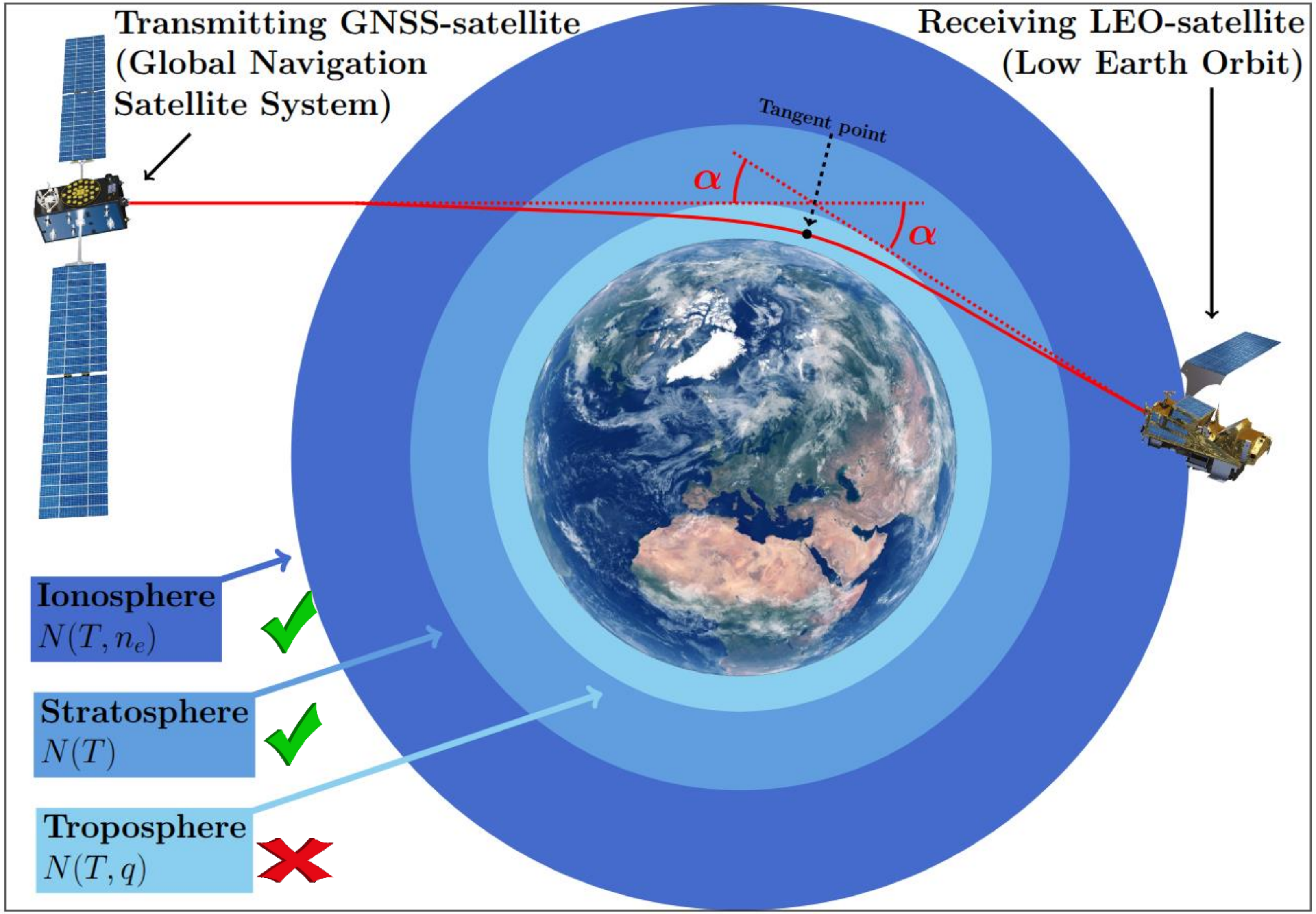
$$N \approx 77.6\,\frac{P}{T} + 6.04\times10^{5}\,\frac{P\,q}{T^2} + 4.03\times10^{7}\,\frac{n_e}{f^2}$$

Refractive index n from bending angle $\boldsymbol{\alpha}$:

$$n(r) \approx \exp\left[\frac{1}{\pi}\int_r^{\infty}\frac{\boldsymbol{\alpha}(a)}{\sqrt{a^2-r^2}}\,\mathrm{d}a\right]$$

Refractivity N from refractive index n:

$$N = (n-1)\times10^6$$



Forward and Inverse model

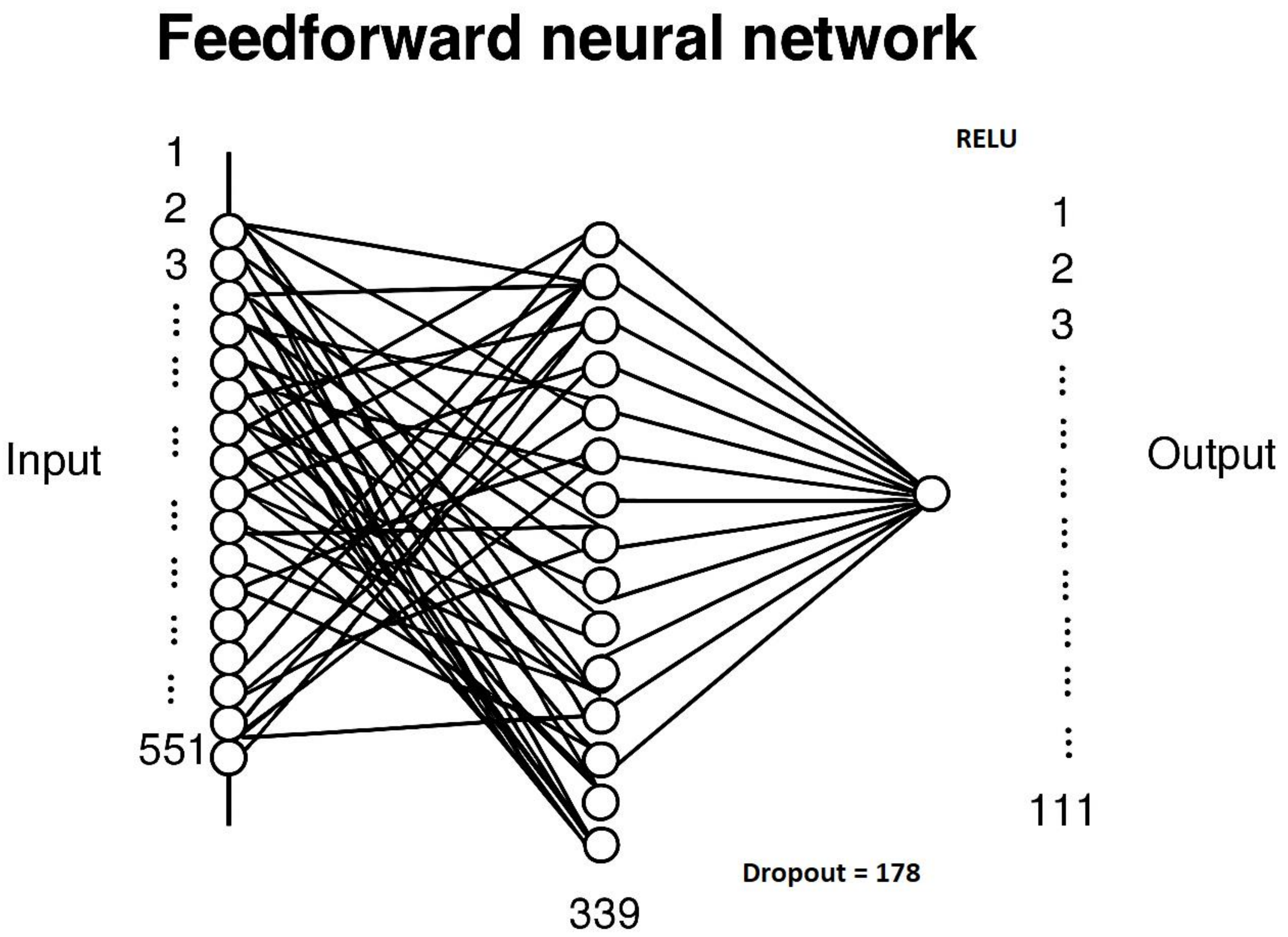
- Forward model $T, q \rightarrow N$
- Inverse model $N \rightarrow T, q$

$$N \approx 77.6 \frac{P}{T} + 6.04 \times 10^5 \frac{P\,q}{T^2} + 4.03 \times 10^7 \frac{n_e}{f^2}$$

Training and testing of *forward* FF-NN

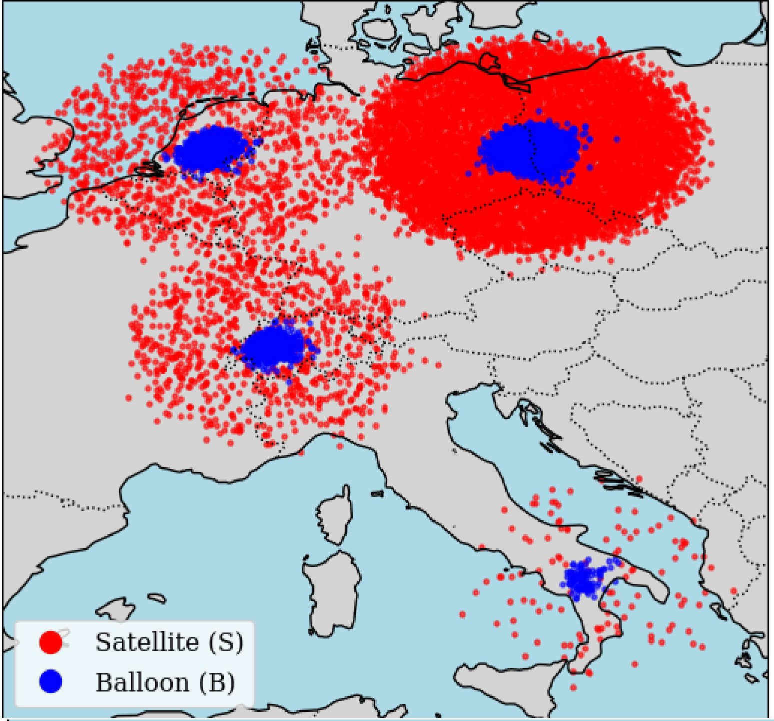
- Test provided simulated climate data
- Compare performance of neural network and RRE
  - Predict values of dataset using a neural network
    - Use a train-test split (80-20) to train network
    - Observe error measurements and compare
    - Test if network performs well by comparing training data and validation data
    - Validation data is data obtained from RFE
  - Compare performance of neural network and RRE
- Relu activation, 1 hidden layer, neurons (551→ 339→ 111)
- Epochs: 200 (convergence after ~13)
- Hyperparameter optimization:
  - Optuna: Bayesian optimization n_layers, layer size etc.

Structure and possible downsides



- Only 1 hidden layer
    - Very little training
    - Huge risk of overfitting
    - Poor gradient descent
    - Hard to transfer model to other problems
    - Could quickly become complicated with more data

- Why we allow this
    - The equation we are approximating is fairly simple
    - We are not aiming to have a portable model



Data

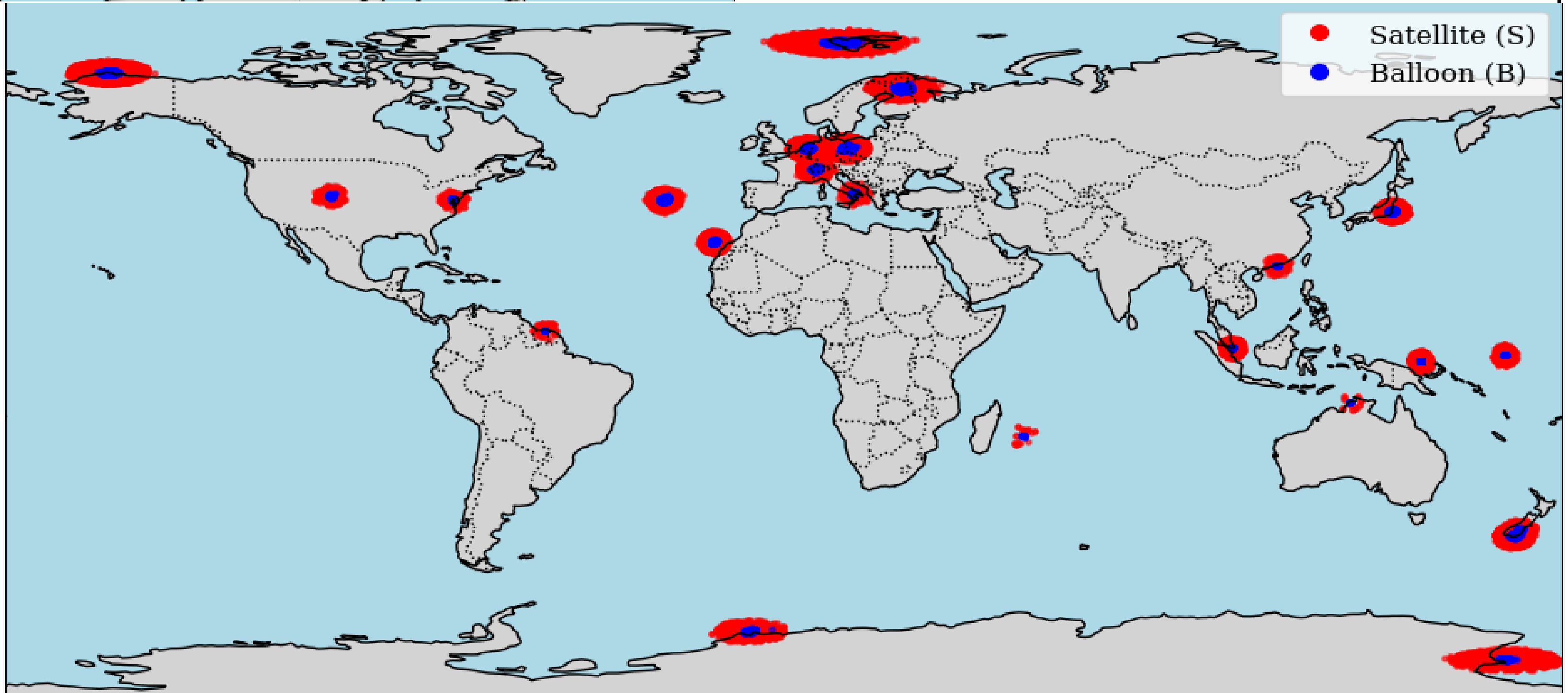
35255 co-located measurements from balloons and satellites

max distance = 300 km, max

max time lag = 3 hours

Vectors

Scalars



Variables:

- Refractivity (247) → (111)
- Refractivity uncertainty (247) → (111)

- Temperature (137) → (93)
- Temperature uncertainty (137) → (93)
- Specific humidity (137) → (93)
- Specific humidity uncertainty (137) → (93)

- Surface pressure
- Surface geopotential height

- Latitude
- Longitude
- Local year
- Local month
- Local day
- Local hour
- Local minute

- Time lag
- Distance

Data treatment for generated refractions

Forward input:
- temp (93)
- temp_sigma (93)
- shum (93)
- shum_sigma (93)
- geop_sfc (1)
- press_sfc (1)
- press_sfc_sigma (1)

FORWARD

Forward output:
- nfrac (111)

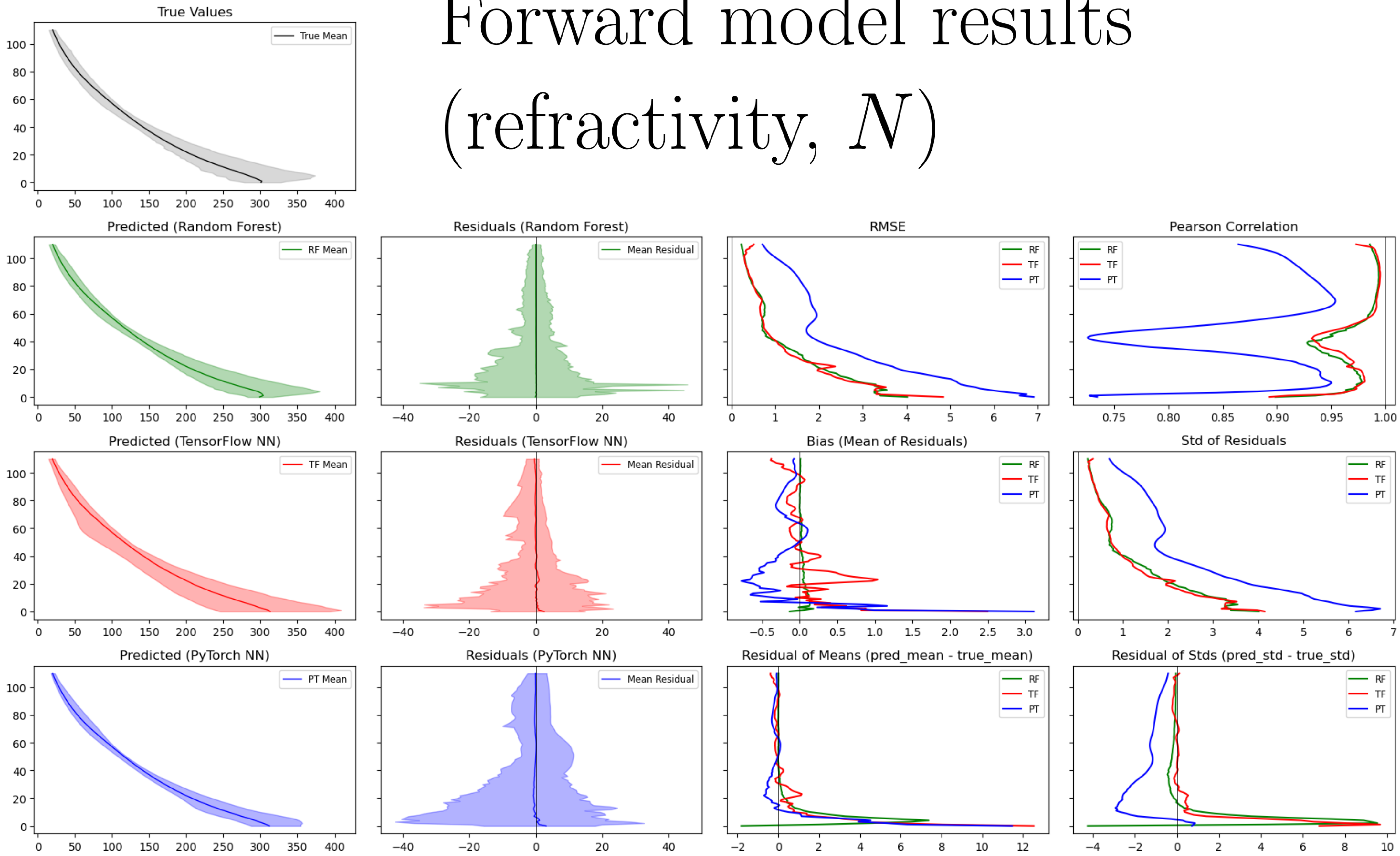
Inverse input:
- nfrac (111)
- Ba_meta (8)
- Sa_meta (8)
- geop_sfc (1)
- press_sfc (1)
- press_sfc_sigma (1)

INVERSE

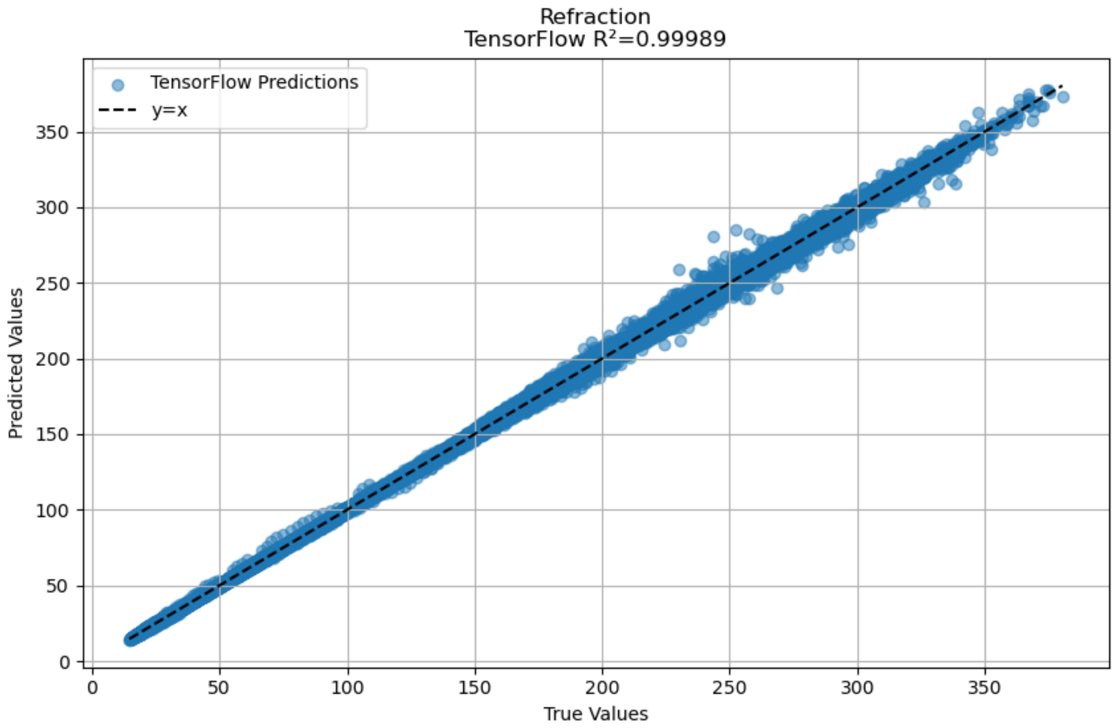
Inverse output:
- shum (93)
- temp (93)

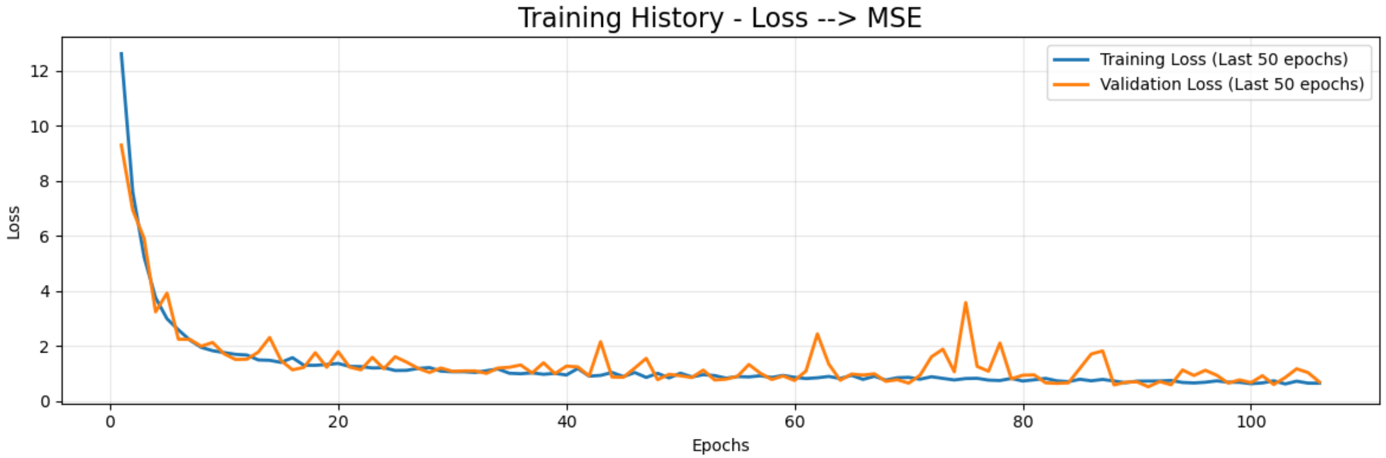
(nfrac (111) is used solely as generated data and as mixed data, where NaNs in measurements are filled with generated data.)

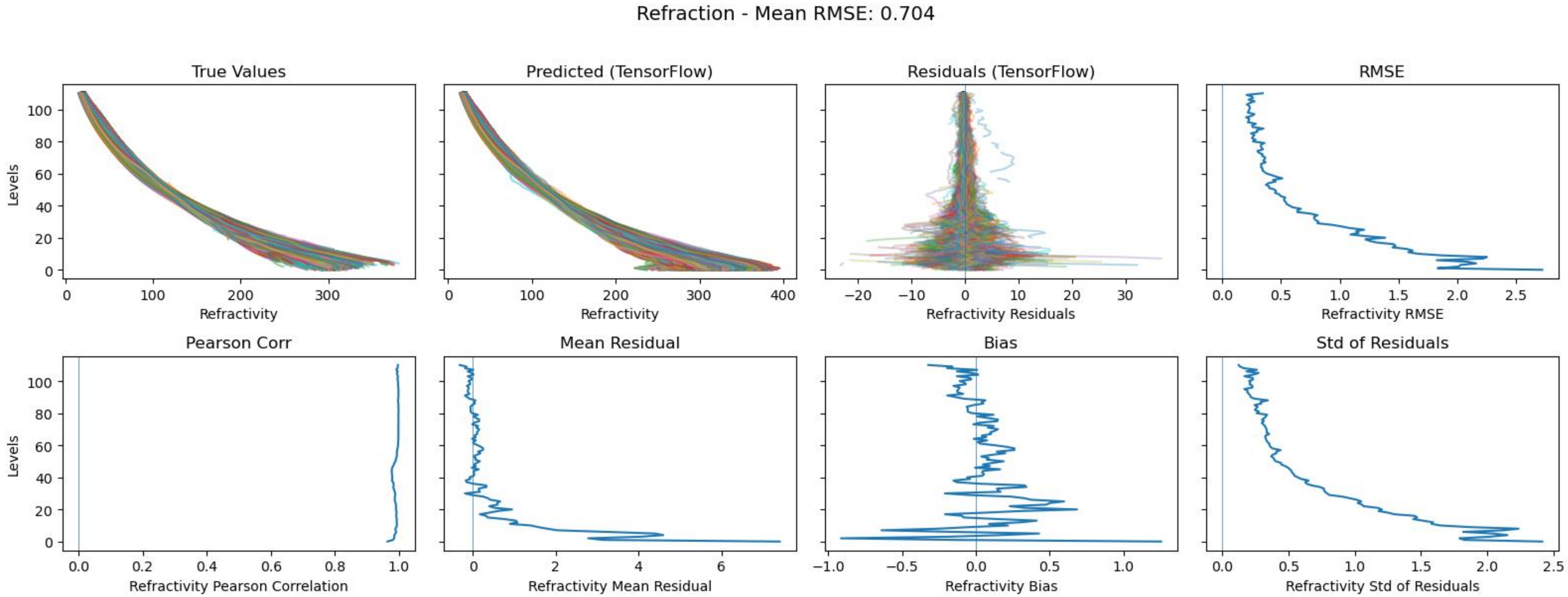
Forward model results (refractivity, N)



Forward model results

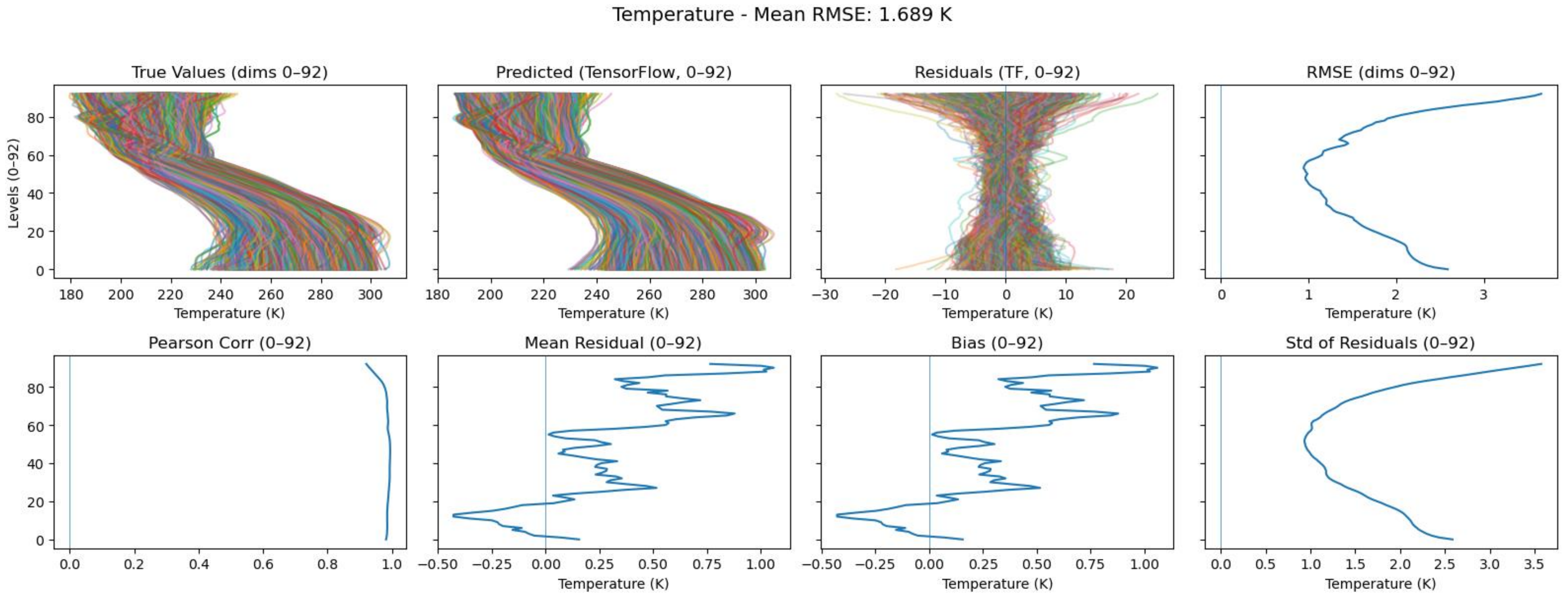


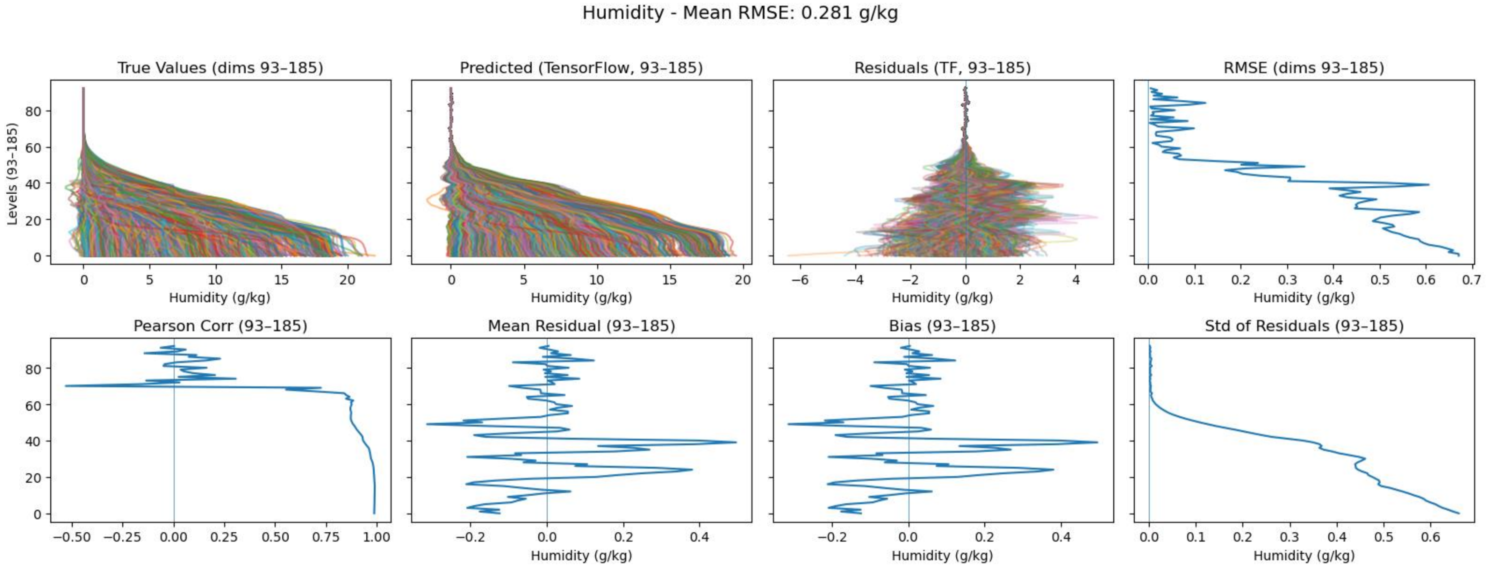


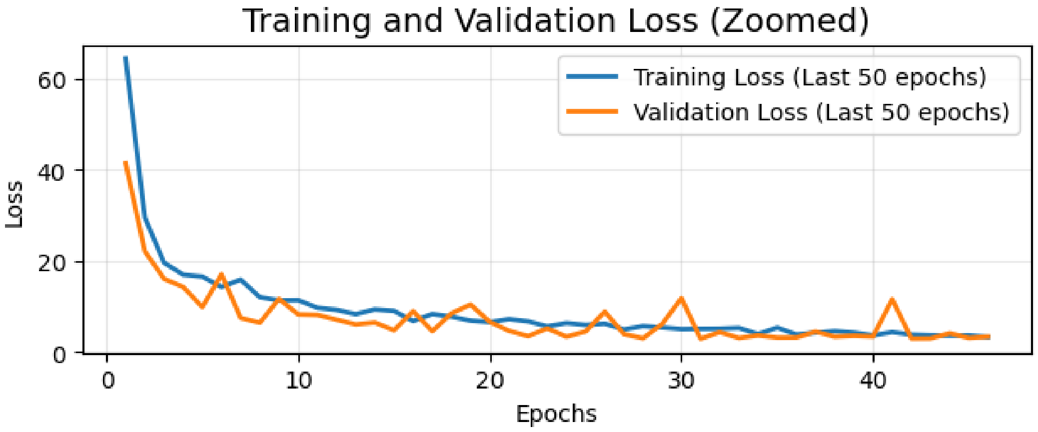


Inverse model results

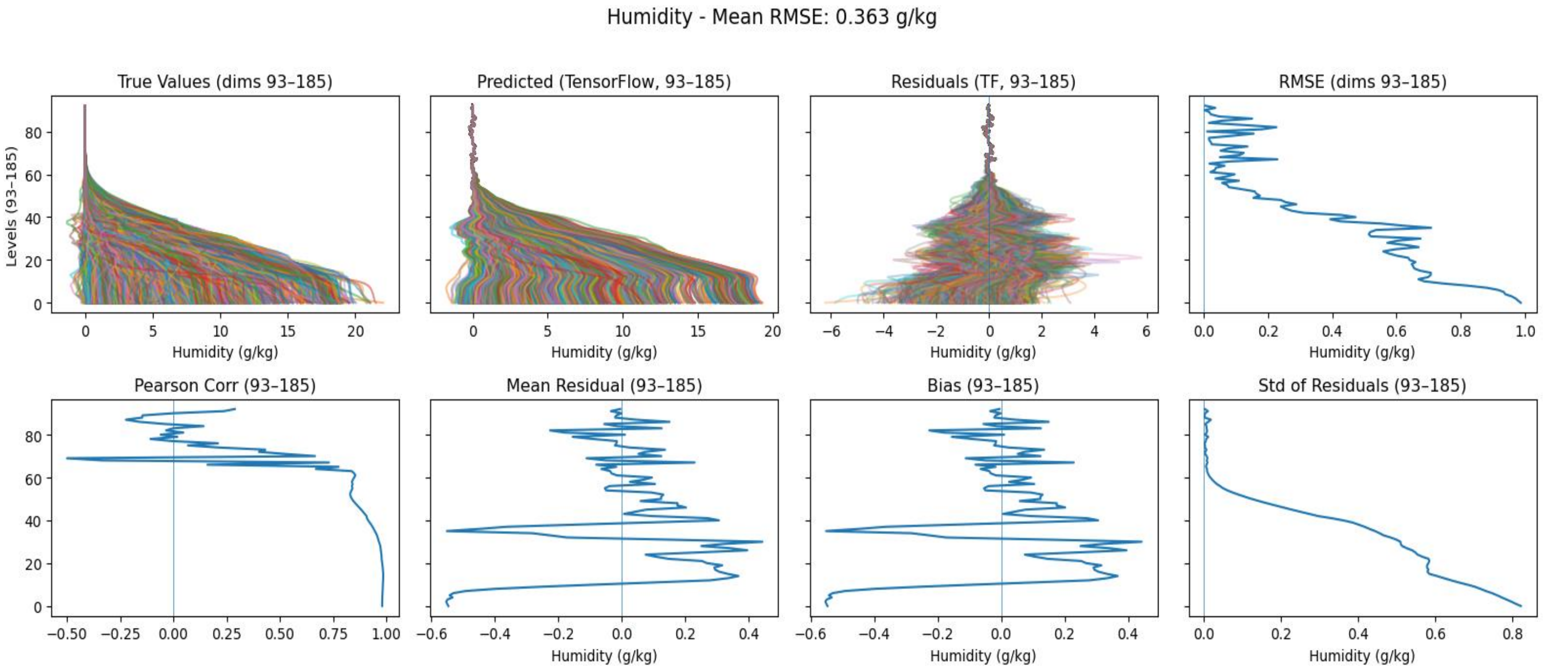
Refraction data is generated



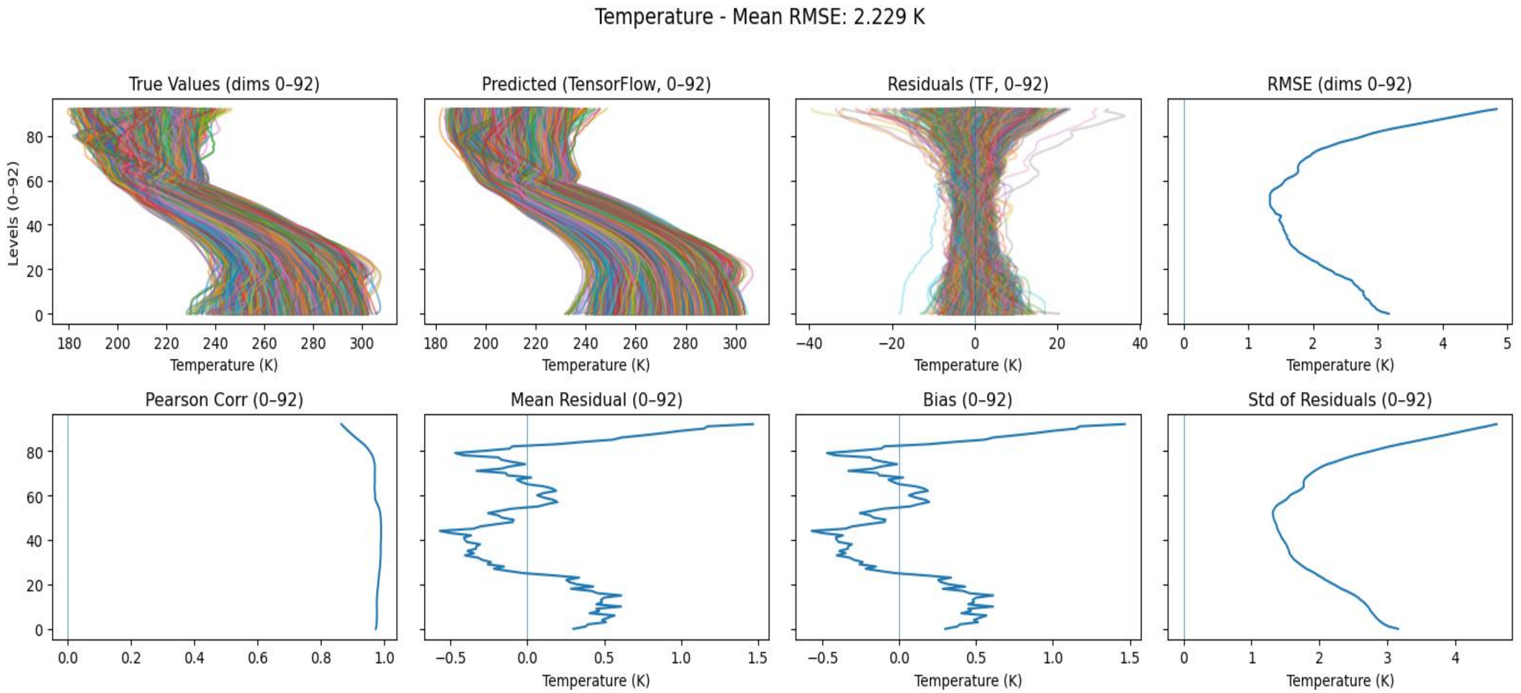




Inverse model results 2



Refraction data is a mix of measurements and generated data



Data treatment on observed data

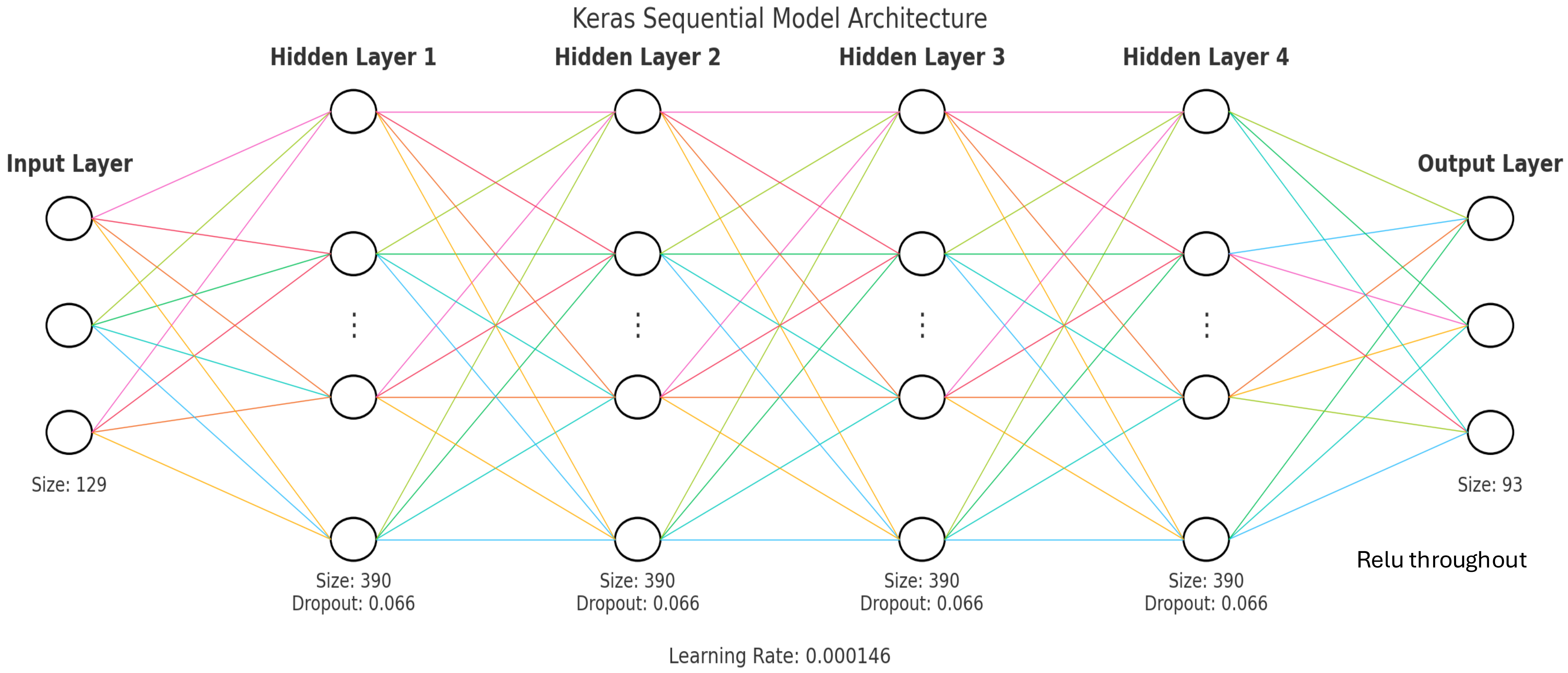
X = input data:
- Satellite measurements of refraction.
- Satellite & Balloon location and timestamp, time lag, distance etc.

Treatment:
- Rows and cols consisting of only NaNs are removed (likewise for output rows.)
- Remaining NaNs are interpolated using the mean for the given level. (not ideal given geographical differences)

y = output data:
- Balloon measurements of temperature (& humidity)

Treatment:
- Same as for input data.

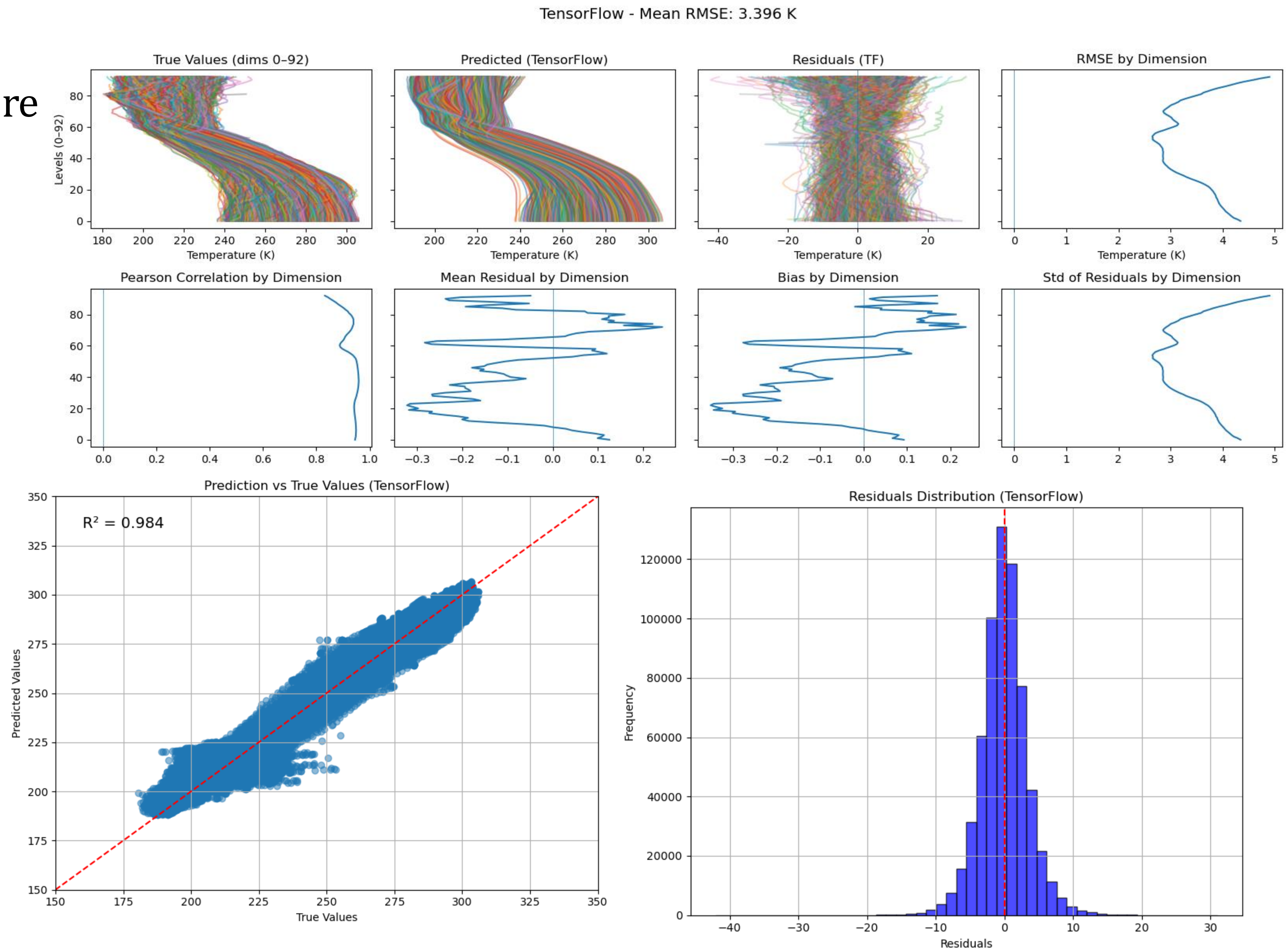
Not ideal:
- ~17% of all refractions were NaNs.
- Up to 3 hour time lag.
- 300 km distance.
- Rain/cloud? I.e. unknowns…



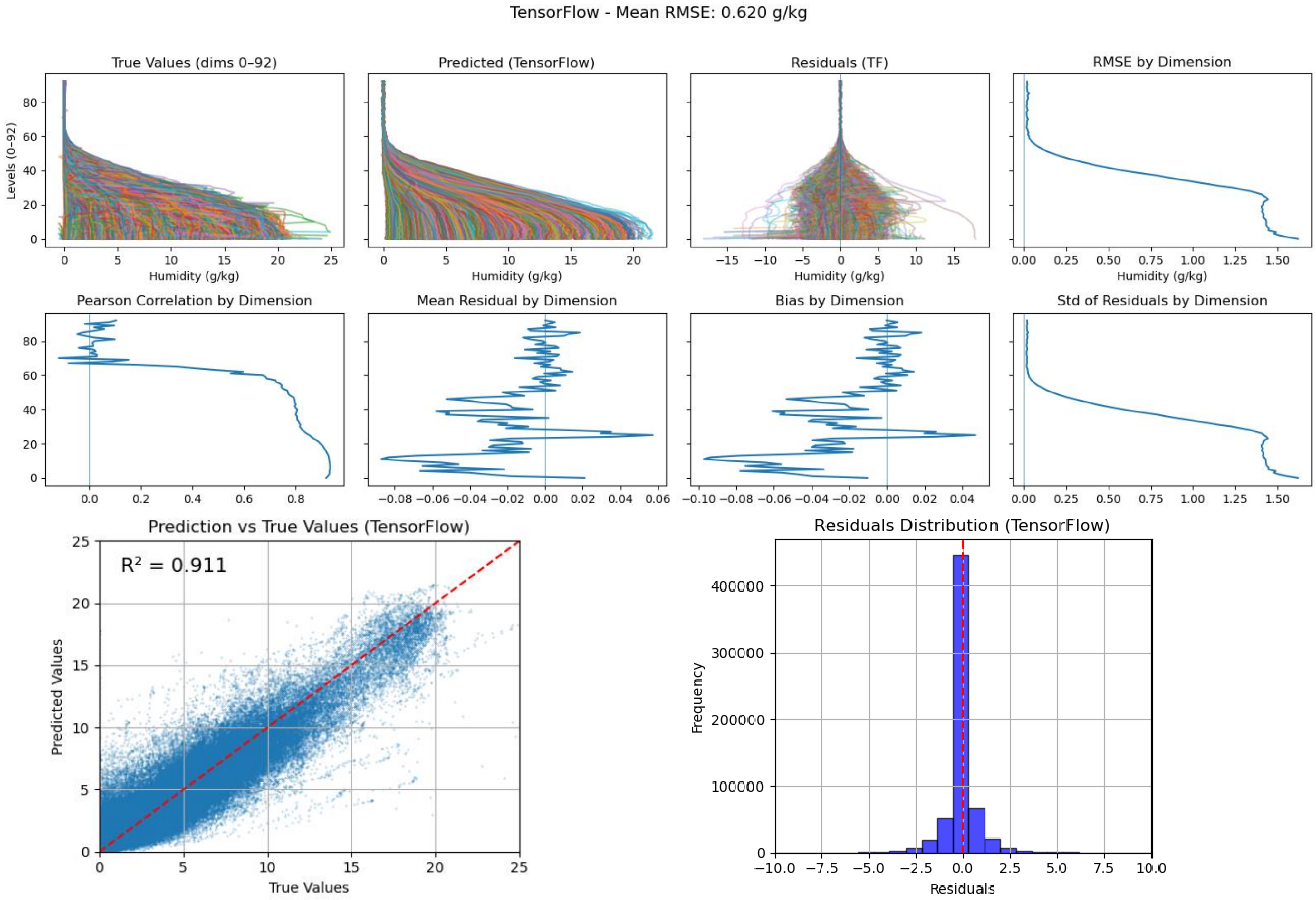
Relu throughout

Optuna optimizes the hyper parameters

Inverse model predicting temperature using observations



Humidity



TensorFlow - Mean RMSE: 0.620 g/kg
True Values (dims 0–92)
Predicted (TensorFlow)
Residuals (TF)
RMSE by Dimension
Levels (0–92)
Humidity (g/kg)
Pearson Correlation by Dimension
Mean Residual by Dimension
Bias by Dimension
Std of Residuals by Dimension
Prediction vs True Values (TensorFlow)
$R^2 = 0.911$
Predicted Values
True Values
Residuals Distribution (TensorFlow)
Frequency
Residuals

Conclusion

- The *forward* NN model can recreate the radio refractivity equation effectively
- Inverse: Uncertainty in balloon height, measurement time and distribution makes approximations and predictions very difficult.
- High error $\rightarrow$ makes it unusable as valid temperature & humidity inputs for further use.

Future work

- Feature importance (e.g. SHAP)
- Use various DL and NN models to predicts specific parameters from balloons, and use a mix of the results
- Run obtained parameters through various feature ranking algorithms
- Obtain an acceptable error measurement to decide if balloon predictions are correct.