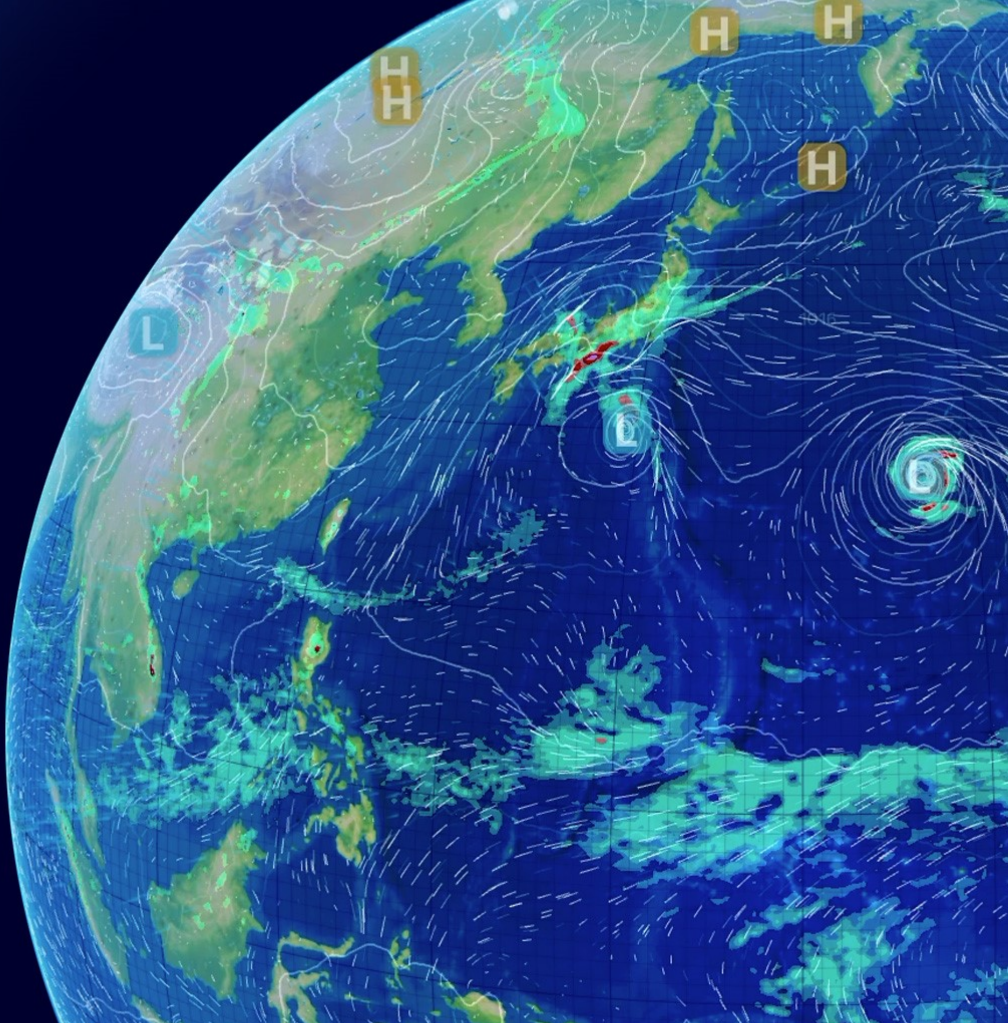
Met Office

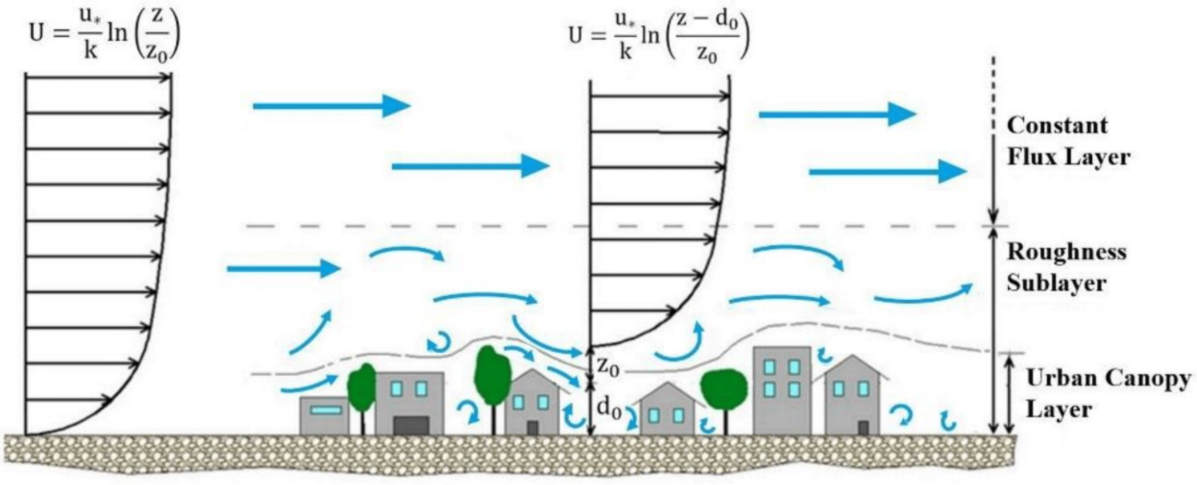
Improving the predictions of the surface fluxes in JULES using Machine Learning

Helena Reid*, Heather Rumbold

*Machine learning, stochastic physics and ensembles team



Monin-Obukhov Similarity Theory



- Provides understanding of the turbulent flows in the atmospheric surface layer
- Describe how atmospheric variables deviate from neutral conditions
- Express turbulent fluxes and gradients as functions of stability parameter z/L (ratio of buoyancy to shear driven turbulence)
- Logarithmic wind profile in neutral conditions

Met Office

Monin-Obukhov similarity theory

$$
\begin{aligned}
\frac{\partial T}{\partial z} + \frac{g}{c_P} &= -\frac{H_0}{c_P \rho_0 v_*} \frac{\phi_h(z/L)}{kz} \\
\frac{\partial q}{\partial z} &= -\frac{E_0}{\rho_0 v_*} \frac{\phi_h(z/L)}{kz} \\
\frac{\partial \mathbf{v}}{\partial z} &= \frac{\tau_0}{\rho_0 v_*} \frac{\phi_m(z/L)}{kz},
\end{aligned}
$$

- Vertical gradients of model variables in surface layer related to surface fluxes via **stability functions, φ,** functions of stability (is buoyancy enhancing/suppressing turbulence?)
- We integrate these equations to get our surface fluxes as a function of model variables and the integrated versions of these stability functions Φ

Met Office

Lots of things depend on U*, which depends on Φ

$$u_* = \frac{\kappa U}{ln\left(\frac{z_h}{z_0}\right) - \varphi\left(\frac{z_h}{L}\right)}$$

- Why not make phi a Neural Network (NN), train on obs, and throw it loads of inputs? Thus saying Φ ≡ ΦNN(LAI, LWdown, Precip, Psurf, Qair, Swdown, Tair, Wind, z0)

Project description

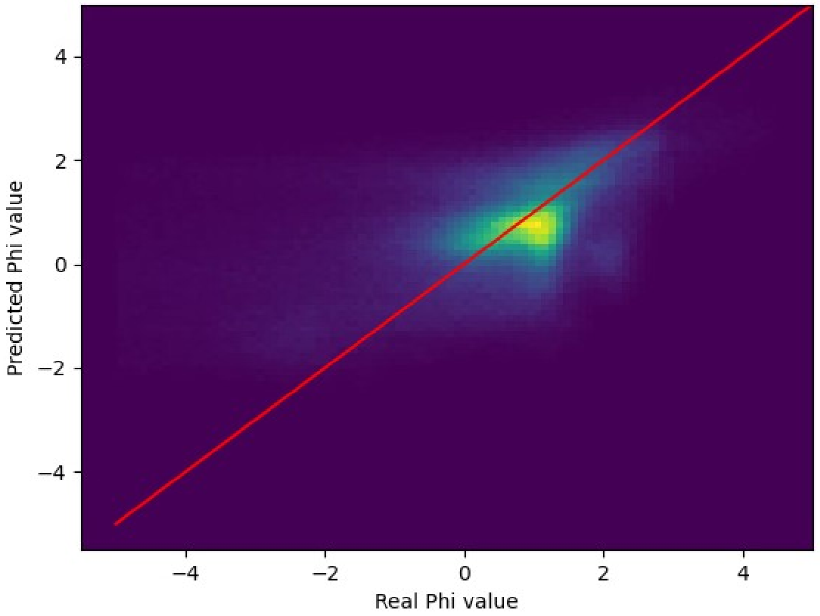
- Use observations of meteorological variables to compute the stability function for momentum (and later heat/moisture)
- Use these inputs paired with the corresponding stability function value to plug into a neural network and output a ML stability function.
- Compute the loss function for the stability function and use this to calculate a ML derived flux to compare back to the physical derived values.
- Train and validate on a fixed number of sites and test on the remainder.

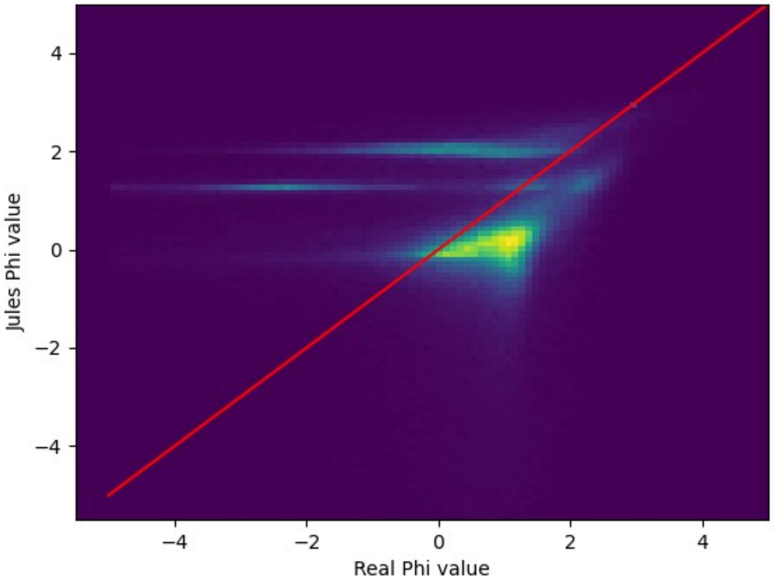
- **Datasets required** - Single point flux tower time series data for 170 sites globally Observed meteorological variables and surface fluxes.

- **Data processing** - Apply quality control mask across all variables with a qc flag. Compute the stability function based on the observations for training and test sites. Scale and normalise the inputs

- **Apply ML workflows** (Multilayer Perceptron MLP using scikit-learn and keras)

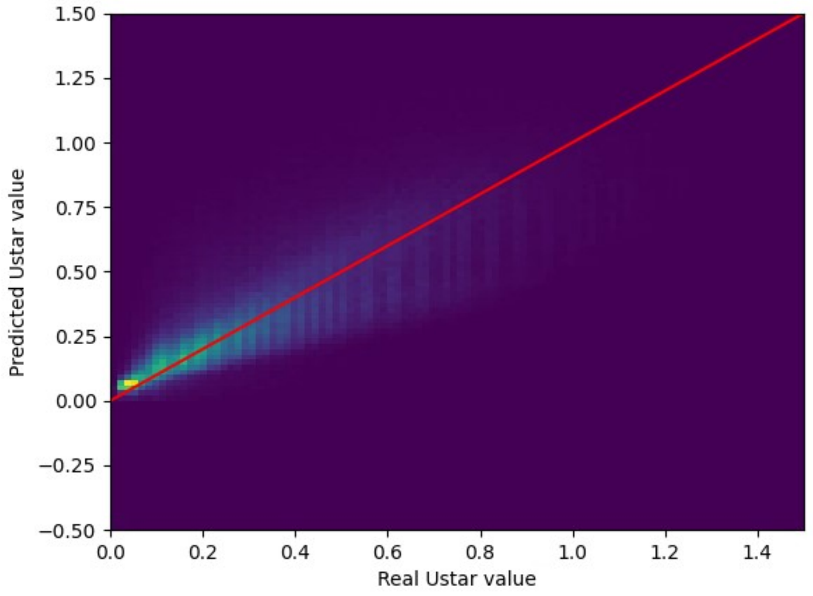
Predictions of the (momentum) stability function

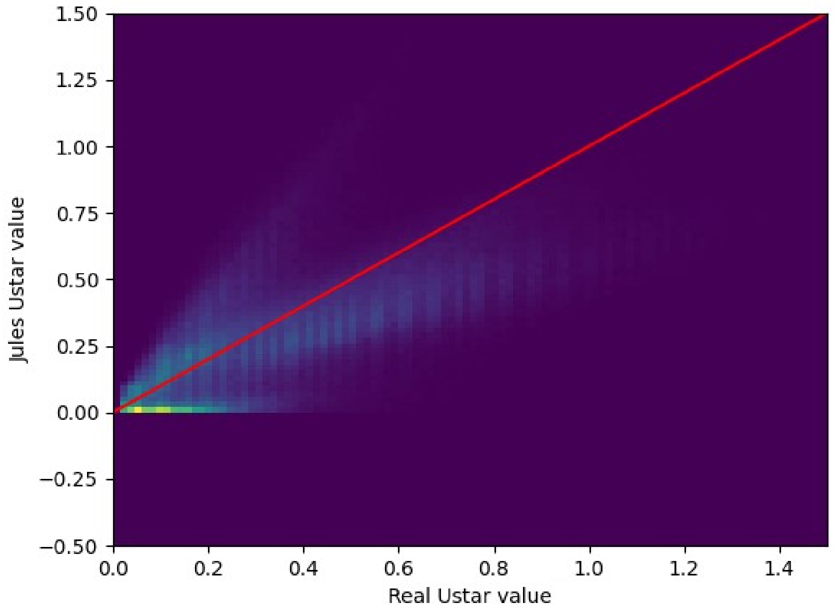




Met Office

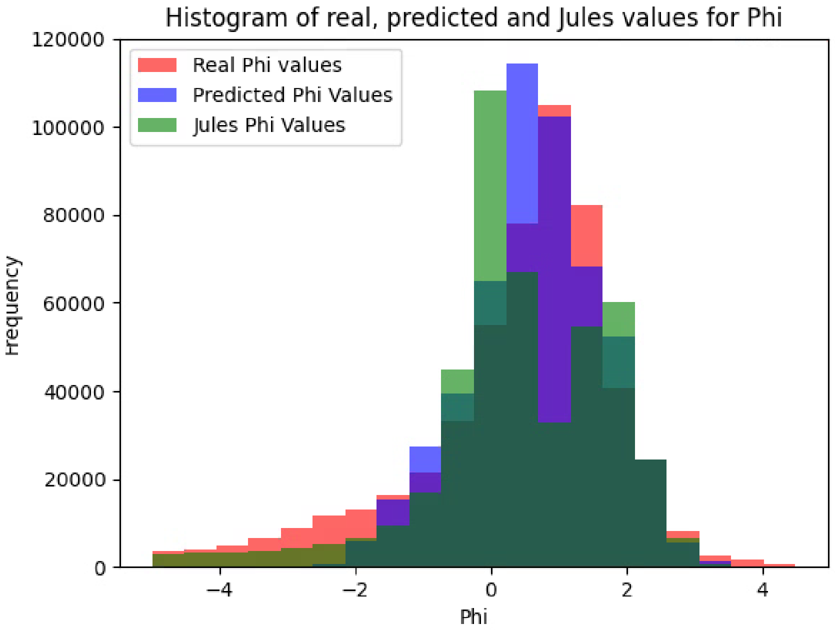
Predictions of U*

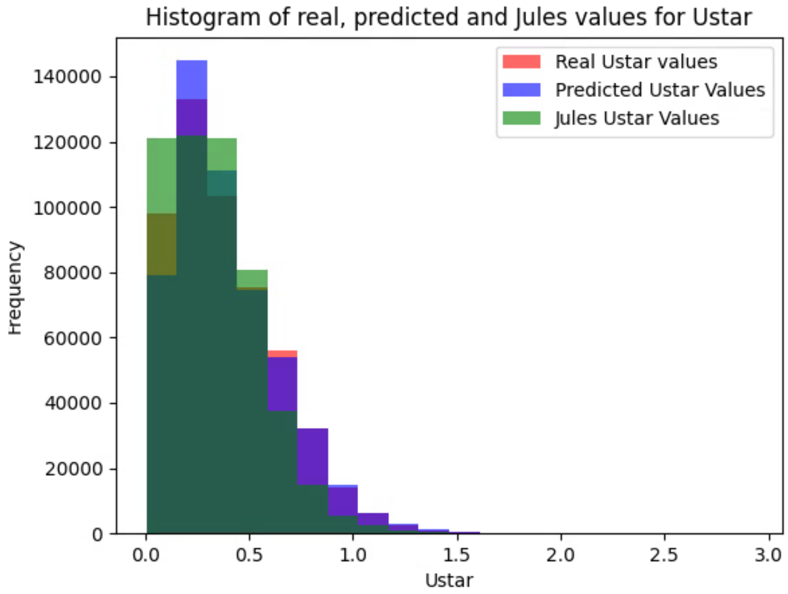




Met Office

Histograms of Φ and U*





Met Office

Stats

RMSE for Ustar between real and predicted values: 0.174
RMSE for Ustar between real and Jules values: 0.213
Mean for Ustar in test data: 0.389
Mean for Ustar in predicted data: 0.397
Mean for Ustar in Jules data: 0.322
Standard deviation for Ustar in test data: 0.258
Standard deviation for Ustar in predicted data: 0.258
Standard deviation for Ustar in Jules data: 0.229
Kolmogorov-Smirnov test for Ustar between real and predicted values:
KstestResult(statistic=0.0366, pvalue=1.16e-304, statistic_location=0.160, statistic_sign=1)
Kolmogorov-Smirnov test for Ustar between real and Jules values:
KstestResult(statistic=0.1001, pvalue=0.0, statistic_location=0.0800, statistic_sign=-1)