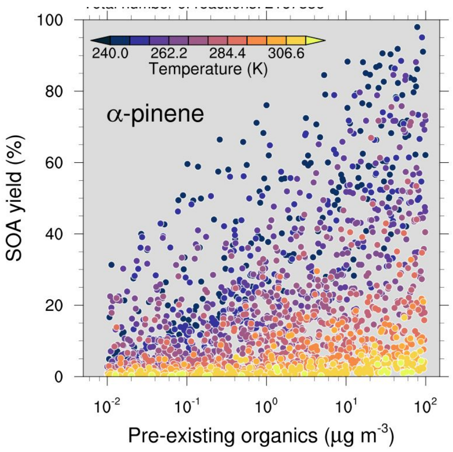
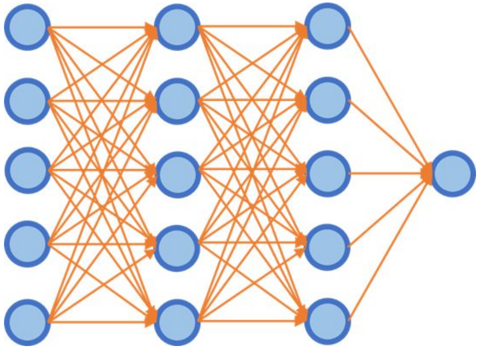
AI4ESS Hackathon: GECKO-A Emulator Challenge

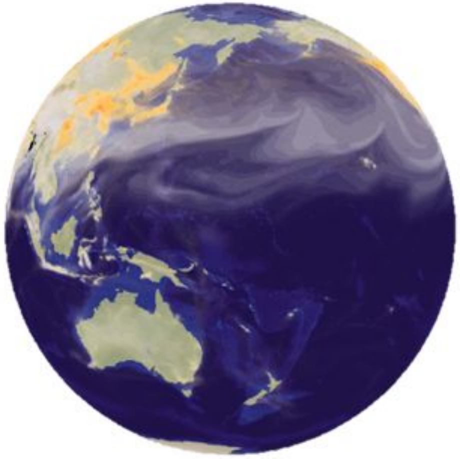
GECKO-A Training Library



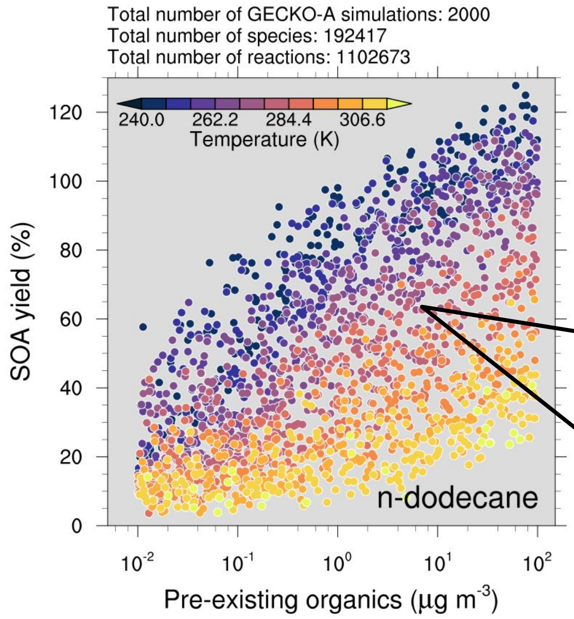
Machine-Learning Emulator

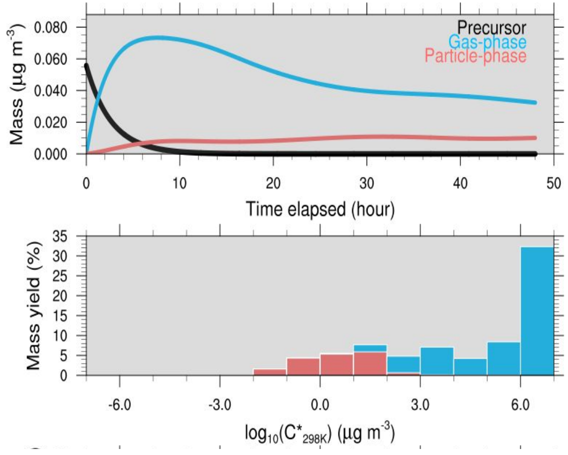


3-D Models



- Many inspiring applications out there: machine-learning emulators using explicit/process-level models, and implementing the trained emulators into large-scale models. Such explicit/process-level models are otherwise too expensive for large-scale models.
- The goal of this project is to train the machine-learning emulator using the “library” generated by the hyper-explicit chemical mechanism, GECKO-A.





1	Time [s]	Precurso	Gas [ug/	Gas [ug/	Gas [ug/	Gas [ug/	Gas [ug/	Gas [ug/	Gas [ug/	Gas [ug/	Gas [ug/
2	1	3.77E-02	0	0	0	0	0	0	0	0	0
3	301.52	3.76E-02	9.61E-26	1.70E-21	2.46E-22	1.69E-20	2.14E-18	3.40E-15	8.12E-15	1.59E-14	1.01E-12
4	602.04	3.76E-02	4.61E-25	8.13E-21	1.06E-19	2.52E-19	1.14E-14	7.65E-15	1.32E-14	7.54E-14	6.72E-12
5	902.56	3.75E-02	1.14E-24	2.00E-20	1.25E-18	5.27E-19	4.08E-14	1.65E-14	3.10E-14	1.18E-13	2.16E-11
6	1203.08	3.74E-02	2.59E-24	4.54E-20	1.06E-18	6.37E-19	3.92E-14	3.97E-14	9.22E-14	1.51E-13	4.87E-11
7	1503.6	3.74E-02	3.98E-24	6.95E-20	9.74E-19	1.02E-18	4.47E-14	6.41E-14	1.28E-13	1.80E-13	8.95E-11
8	1804.12	3.73E-02	5.72E-24	9.97E-20	8.14E-19	1.16E-18	3.95E-14	9.65E-14	1.69E-13	2.20E-13	1.47E-10
9	2104.64	3.72E-02	8.11E-24	1.41E-19	7.54E-19	1.63E-18	4.27E-14	1.42E-13	1.91E-13	2.64E-13	2.22E-10
10	2405.16	3.72E-02	9.83E-24	1.71E-19	6.16E-19	2.02E-18	3.56E-14	1.91E-13	2.30E-13	2.98E-13	3.17E-10
11	2705.68	3.71E-02	1.10E-23	1.91E-19	5.04E-19	2.57E-18	2.98E-14	2.76E-13	2.54E-13	3.47E-13	4.32E-10
12	3006.2	3.70E-02	1.25E-23	2.16E-19	4.59E-19	3.44E-18	2.97E-14	3.87E-13	2.92E-13	4.23E-13	5.70E-10
13	3306.72	3.70E-02	1.46E-23	2.52E-19	4.38E-19	4.58E-18	3.14E-14	5.36E-13	3.32E-13	4.86E-13	7.30E-10
14	3607.24	3.69E-02	1.69E-23	2.89E-19	4.18E-19	5.98E-18	3.22E-14	7.04E-13	3.68E-13	6.26E-13	9.13E-10
15	3907.76	3.68E-02	1.95E-23	3.34E-19	3.50E-19	7.58E-18	2.74E-14	9.25E-13	4.14E-13	6.92E-13	1.12E-09
16	4208.28	3.68E-02	2.23E-23	3.79E-19	2.69E-18	9.67E-18	6.77E-14	1.16E-12	4.73E-13	1.34E-12	1.35E-09
17	4508.8	3.67E-02	2.42E-23	4.11E-19	5.93E-18	1.22E-17	1.27E-13	1.43E-12	5.26E-13	1.38E-12	1.61E-09
18	4809.32	3.66E-02	2.48E-23	4.21E-19	4.95E-18	1.50E-17	1.15E-13	1.75E-12	5.75E-13	1.49E-12	1.89E-09
19	5109.84	3.66E-02	2.58E-23	4.37E-19	4.22E-18	1.84E-17	1.11E-13	2.14E-12	6.40E-13	1.51E-12	2.19E-09
20	5410.36	3.65E-02	2.77E-23	4.67E-19	3.50E-18	2.18E-17	9.44E-14	2.56E-12	6.96E-13	1.55E-12	2.53E-09

Demo: what the data looks like

GECKO-A Library:

- 2000 GECKO-A simulations: in each run, we run GECKO-A under certain condition for 5 days
- 2000 input files (csv).
- Each file contains: (i) mass of precursors; (ii) mass of products in the gas-phase; and (iii) mass of products in the particle-phase. All (i)-(iii) as a function of time.

Metadata

Metadata	Units	Label
Number Experiments	2000	id
Total Timesteps	1440	Time
Timestep Delta	300 seconds	-

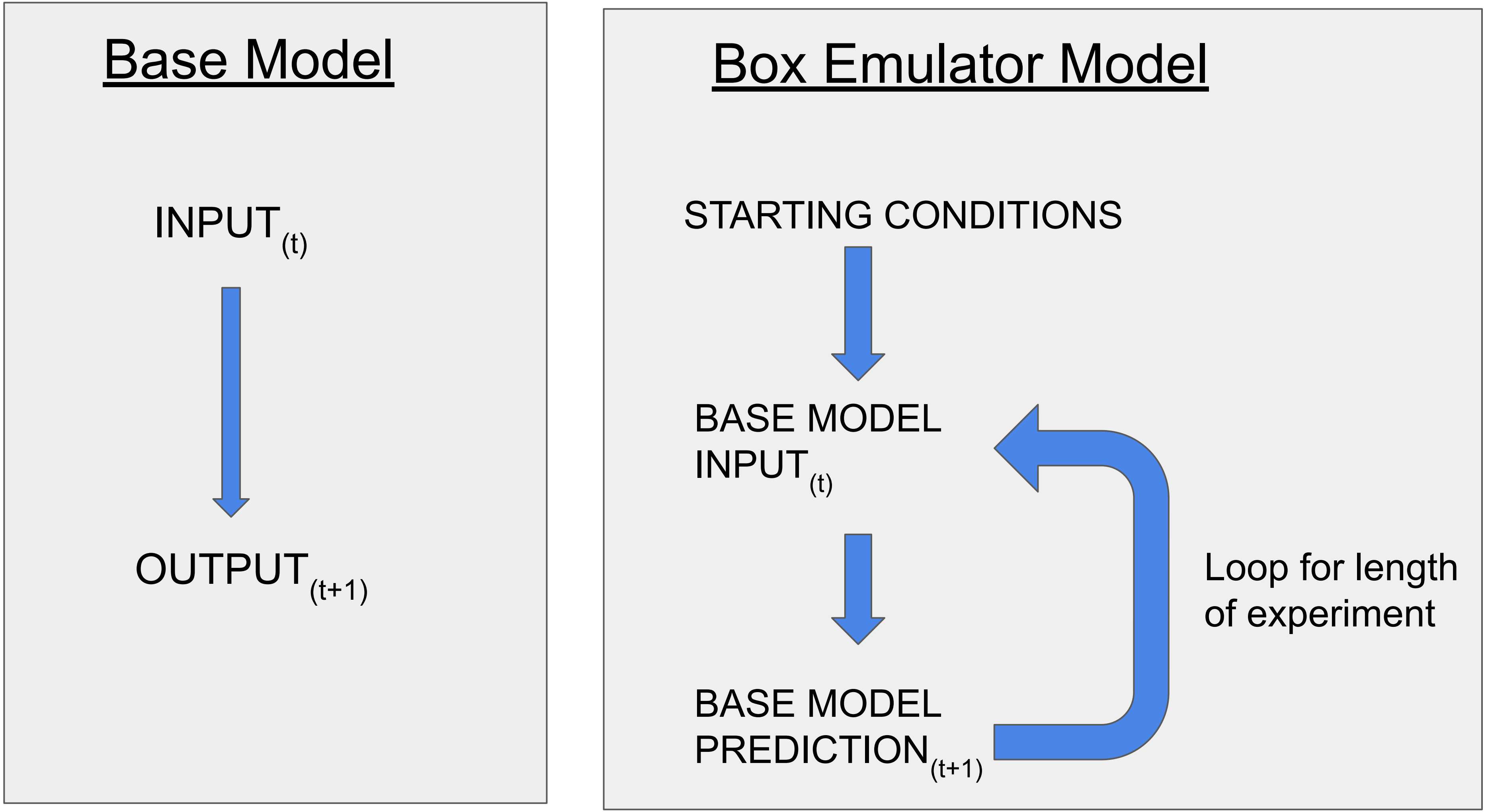
DATA

Potential Input Variables

Variable Name	Units	Type
Precursor	ug/m3	Varies
Gas	ug/m3	Varies
Aerosol	ug/m3	Varies
Temperature	K	Static
Solar Zenith Angle	degree	Static
Pre-existing Aersols	ug/m3	Static
o3	ppb	Static
nox	ppb	Static
oh	10^6 molec/cm3	Static

Potential Output Variables

Variable Name	Units	Type
Precursor (at t+1)	ug/m3	Varies
Gas (at t+1)	ug/m3	Varies
Aerosol (at t+1)	ug/m3	Varies

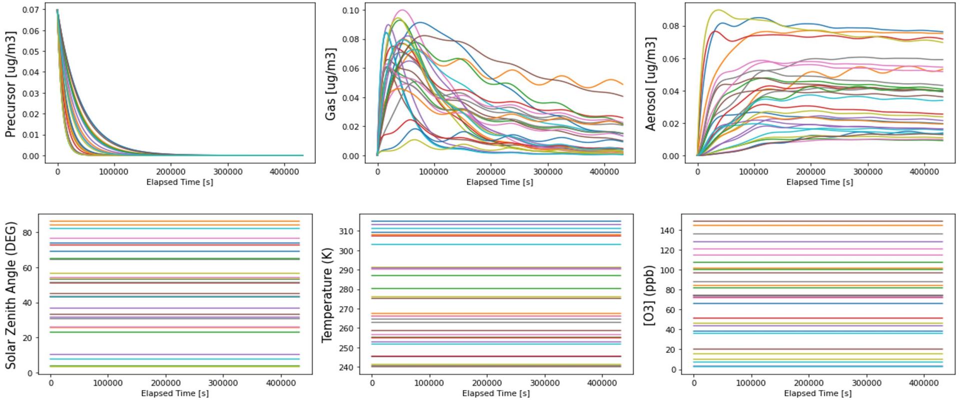

Base Model
INPUT(t)
OUTPUT(t+1)
Box Emulator Model
STARTING CONDITIONS
BASE MODEL INPUT(t)
BASE MODEL PREDICTION(t+1)
Loop for length of experiment

Team 42: Gecko-A Emulation

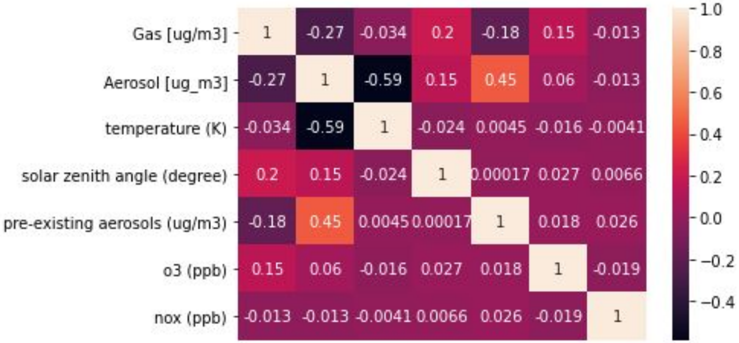
Jeonghoe Kim, Josh Alland, Hemanth SK. Vepuri

- Methods
  a. Linear models (LinearRegression, Ridge, Lasso, ElasticNet), tree-based models (RandomForest, GradientBoosting), and neural network models (DNN, CuDNN-LSTM, LSTM) were tested.
  b. LinearRegression and CuDNN-LSTM show the best performance.
- Data

Time series of features in selected experiments



Correlation Heat map

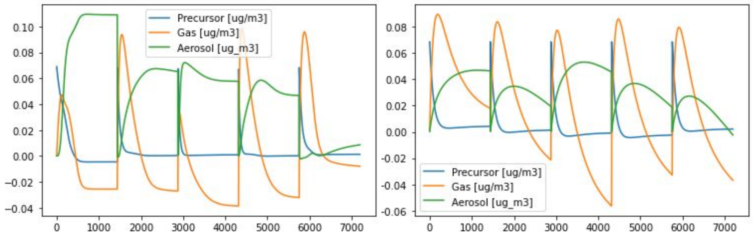


Team 42: Gecko-A Emulation

- Metric Scores (Box Emulator) and Time Series of Concentrations

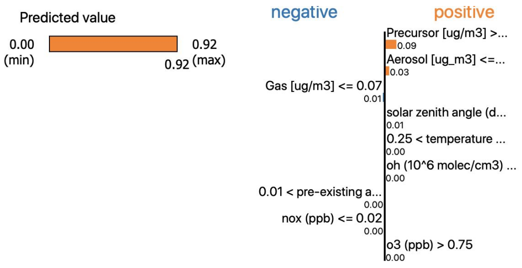
Lin. Reg.	Precursor	Gas	Aerosols
RMSE	0.00500	0.02494	0.01381
R2	0.82047	0.65767	0.74146
Hellenger	0.31740	0.31260	0.35933

Cu-LSTM	Precursor	Gas	Aerosols
RMSE	0.00377	0.03379	0.02400
R2	0.91294	0.57065	0.69842
Hellenger	0.06289	0.49109	0.27553



Lin. Reg. (Left) CuDNN-LSTM (Right)

- Feature Selection by Random Forest



0.1396530182381005

Feature	Value
Precursor [ug/m3]	0.95
Aerosol [ug_m3]	0.01
Gas [ug/m3]	0.04
solar zenith angle (degree)	0.88
temperature (K)	0.42
oh (10^6 molec/cm3)	0.14
pre-existing aerosols (ug/m3)	0.06
nox (ppb)	0.01
o3 (ppb)	0.79

Lessons Learned with AI4ESS and Hackathon

- Machine Learning is a very powerful tool, but precise and sophisticated design of ML model is required. Using ML models without consideration often makes a catastrophically bad prediction.
- Pursuing a “best” accuracy of ML model does not guarantee a successful adoption of ML model to the prediction of certain phenomena.

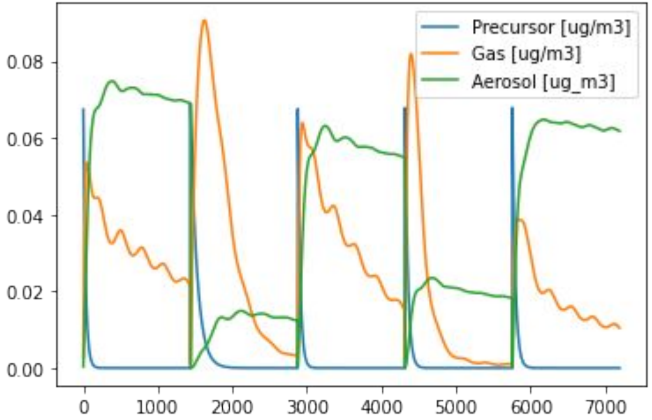
Team 48: GECKO

- Jiaze Wang*, Antonio Lorenzo*, Lee Brent*, Jared Brewer*
- Linear Regression, PCA, Random Forest Tree Regressor, Gradient Boosting Tree Regressor, Fully Connected Neural Network, Res Neural Net(failed to work)

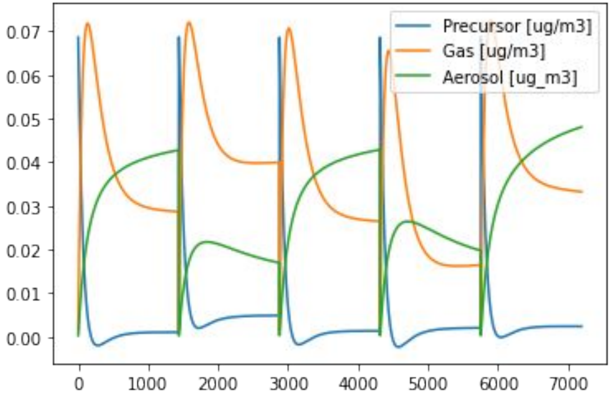
Linear Regression model

Metrics for Box Emulator:
RMSE: Precursor: 0.00375, Gas: 0.01818, Aerosols: 0.02049
R2: Precursor: 0.88393, Gas: 0.55147, Aerosols: 0.72645
Hellenger Distance: Precursor: 0.35322, Gas: 0.22406, Aerosols: 0.54450
<matplotlib.axes._subplots.AxesSubplot at 0x7f8031acddd8>

True_box



Pred_box

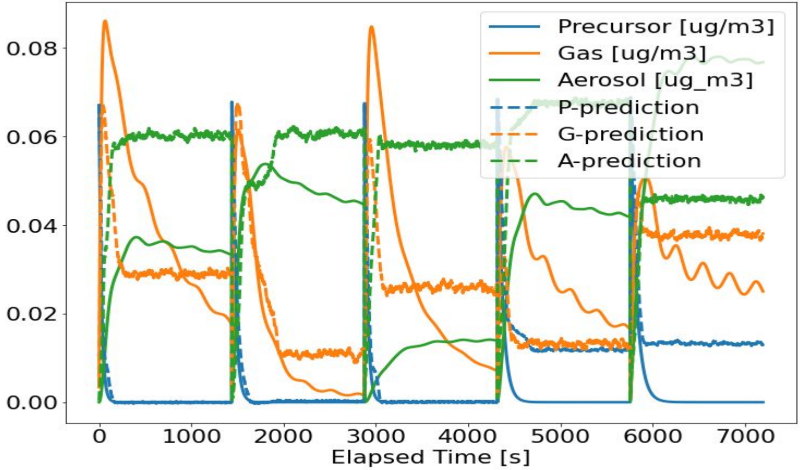


Team 48: GECKO

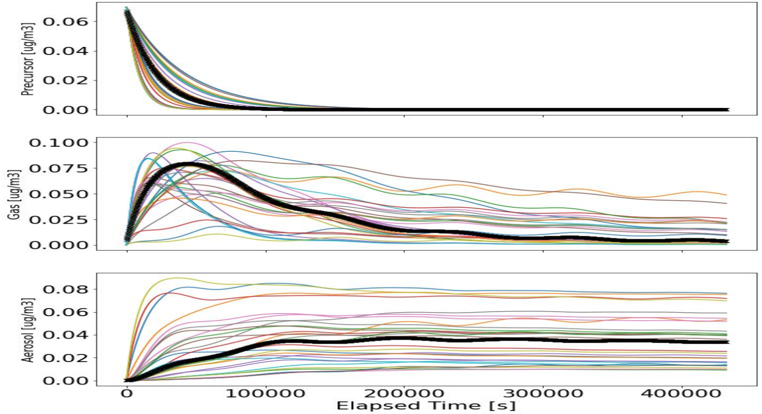
- Metrics of ML and parameter selection doesn’t work well
- RNN doesn’t work
- Lesson learned: general ideas about ML and applications of ML in earth sciences, and basic knowledge on how to do ML in python
- Challenges: Having trouble in parameter selection, metrics and visualization on evaluation ML models with so limited knowledge on ML packages in python

Team 23: GECKO (Iyasu Eibedingil, Ales Kuchar)

- Summary of methods tried
  - Added gaussian noise helped to improved densely connected NN performance in terms of Box Emulator, however, still not able to capture autocorrelation of outputs variables



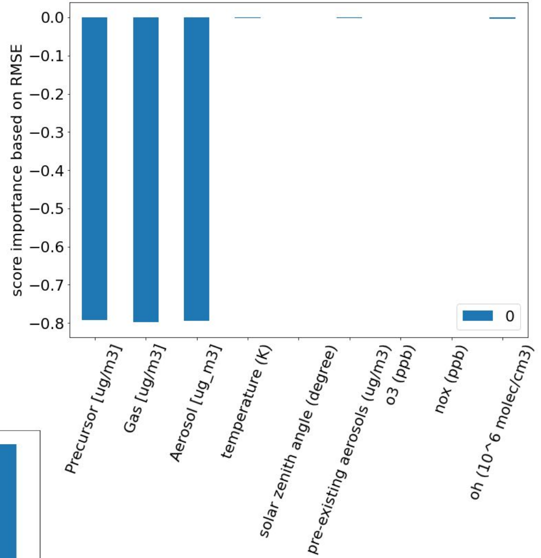
  - Autoregression model was tested (see black line on top of NN may solve the issue above => motivation for LSTM architecture



Team 23: GECKO (Iyasu Eibedingil, Ales Kuchar)

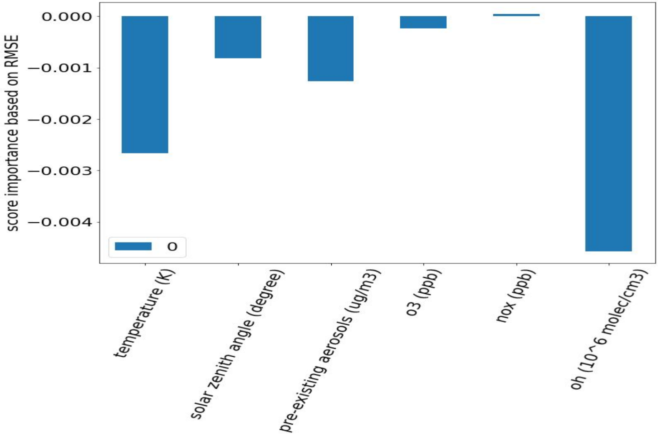
Interpretation of the ML model

- Score importance using RMSE shows that our output variables at t_0 are highly important for t_0+1
- Otherwise temperature and OH seems to be most important



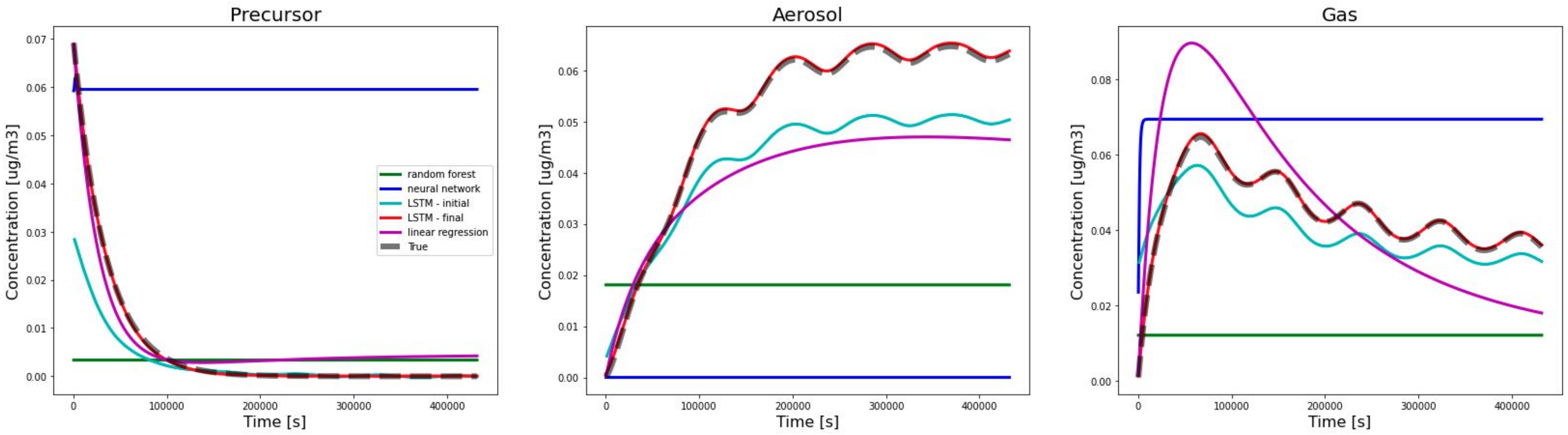
Challenges

- Lack of time/workforce
- Jupyterlab issues

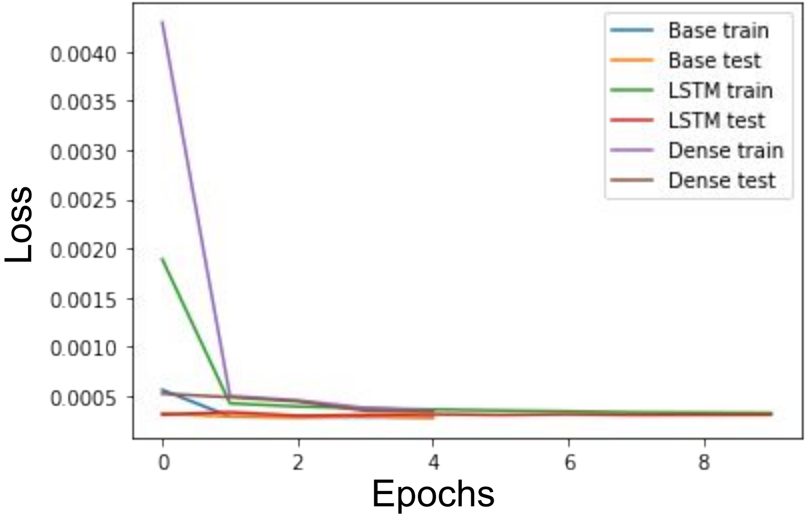


Team 10: GECKO

- Team Members: Devon Dunmire, Errami Larbi, Jean Lim, Luke Thompson
- Methods tried: Linear Regression, Random Forest, Dense Neural Network. LSTM



Team 10: GECKO




```
Metrics for base model:
RMSE: Precursor: 0.00023, Gas: 0.00019, Aerosols: 0.00022
R2: Precursor: 0.99972, Gas: 0.99994, Aerosols: 0.99993
Hellenger Distance: Precursor: 0.00003, Gas: 0.00002,
Aerosols: 0.00568

Metrics for LSTM:
RMSE: Precursor: 0.00035, Gas: 0.00051, Aerosols: 0.00079
R2: Precursor: 0.99949, Gas: 0.99972, Aerosols: 0.99961
Hellenger Distance: Precursor: 0.00024, Gas: 0.00013,
Aerosols: 0.00236
```

Challenges:

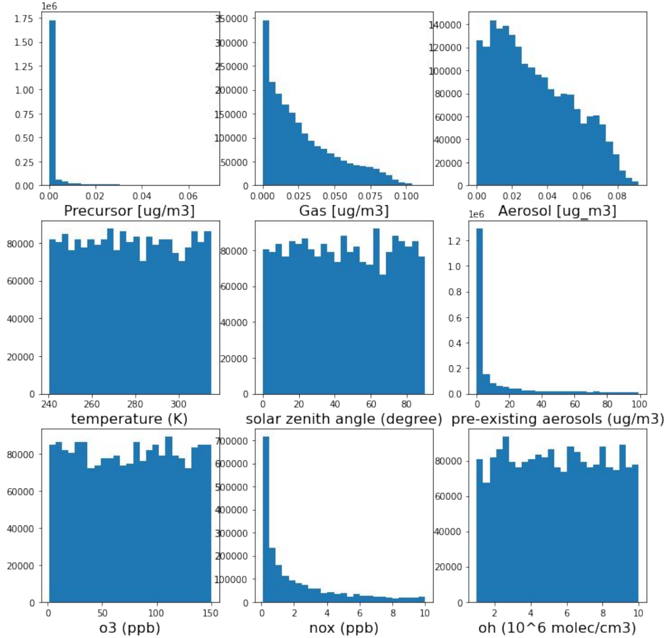
- Our best model did not outperform base model
- Interpretation of LSTM model

Team 17: GECKO

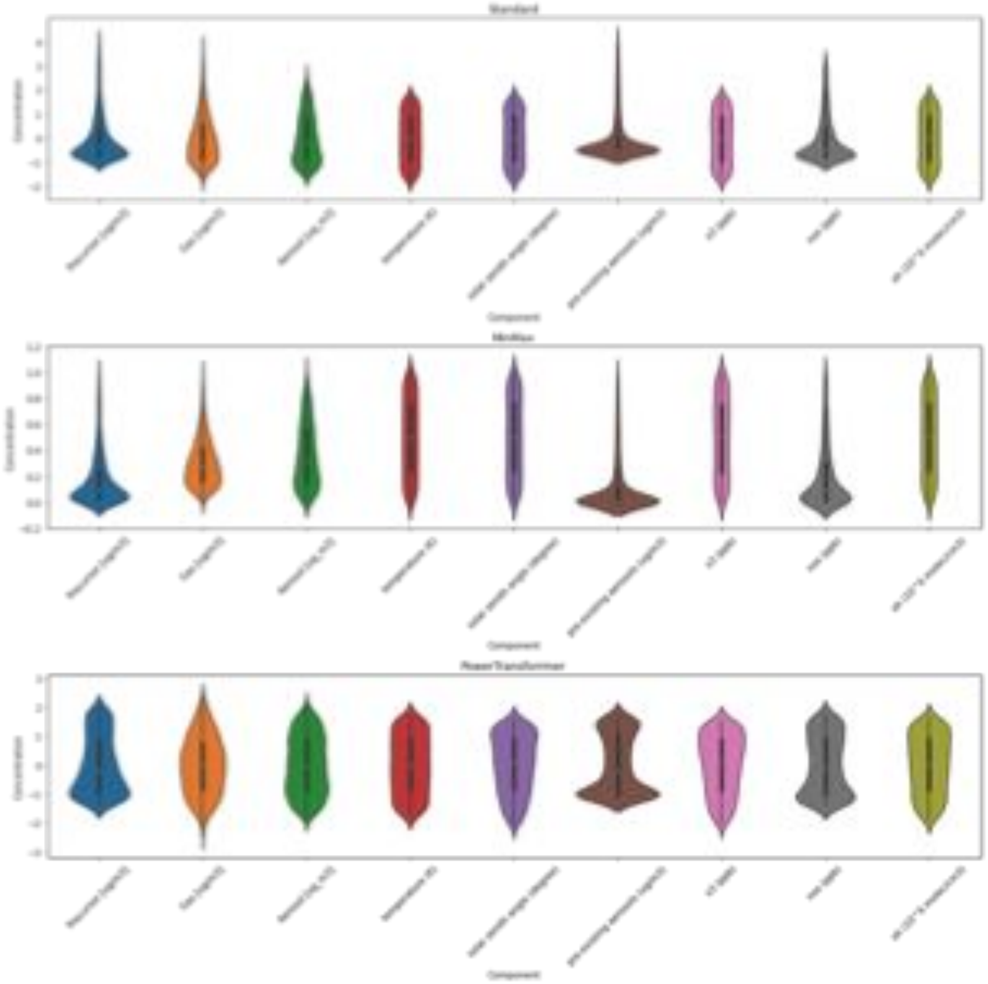
Bowen Fang, Jonathan Eliashiv, Shuting Zhai, Esther Lee, Fernando Campo*, and Raghavendra S. Mupparthy*

Summary of methods tried: Linear, Random Forest Regressor, DNN, simple RNN, LSTM RNN

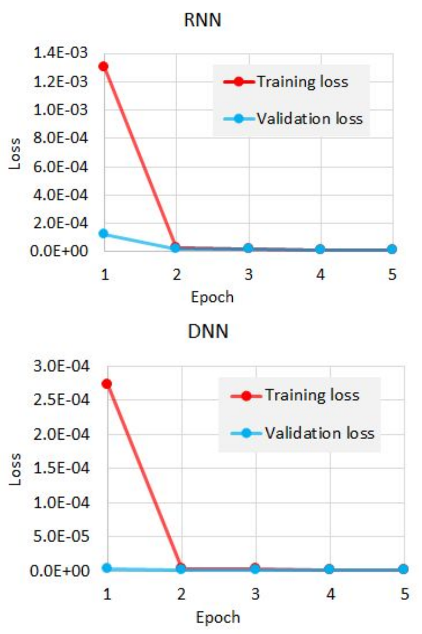
Visualization of training input data

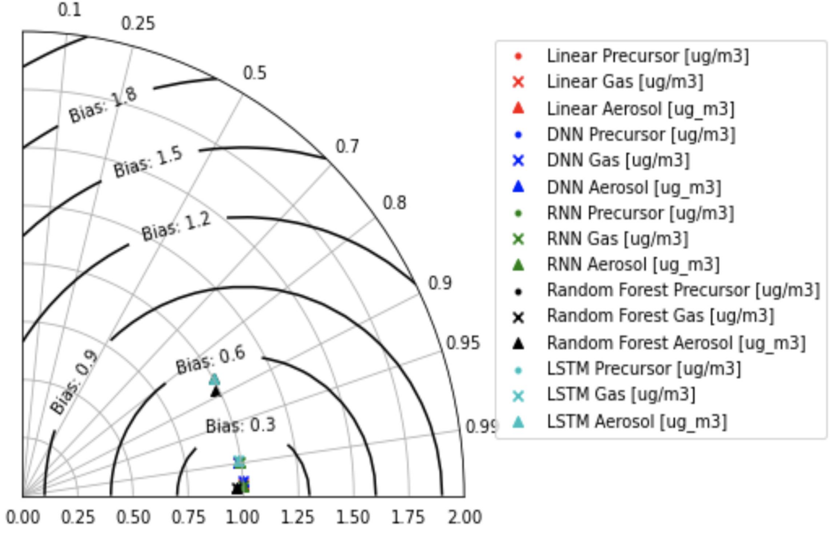


Transformed data (Standard, MinMax, Power)

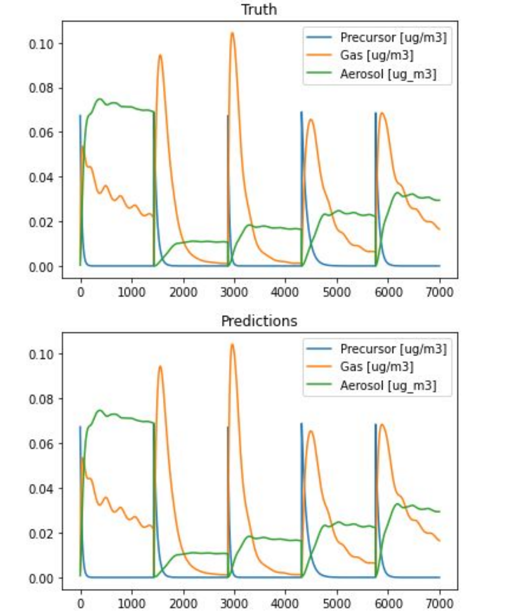


Team 17: GECKO





All of the models we tried produce similar result



The model prediction shows good agreement with validation data

Lessons learned from Hackathon:

- Fanciest tools are not always the best
- Data preparation (pipeline scaling) is really important
- Even with non-Gaussian transformation(MinMax transform), the result was good
- Precision is as important as accuracy. You can’t improve one without the other (RMSE, MAE)

Challenges: spin-up time

(useful plots / charts)

```
====================================
Precursor [ug/m3]
====================================
```

Model Type	MAE	RMSE	R	Hellinger Distance	STDVAR	Truth STDVAR
Linear Val	0.005001	0.013297	0.001411	0.000012	0.009640	0.009633
DNN Val	0.004986	0.013287	0.001939	0.000001	0.009637	0.009633
RNN Val	0.004987	0.013288	0.001937	0.000002	0.009643	0.009633

```
====================================
Gas [ug/m3]
====================================
```

Model Type	MAE	RMSE	R	Hellinger Distance	STDVAR	Truth STDVAR
Linear Val	0.025075	0.033541	-0.130512	0.000003	0.024380	0.024381
DNN Val	0.025074	0.033576	-0.131674	0.000087	0.024375	0.024381
RNN Val	0.025062	0.033531	-0.131790	0.000012	0.024343	0.024381

```
====================================
Aerosol [ug_m3]
====================================
```

Model Type	MAE	RMSE	R	Hellinger Distance	STDVAR	Truth STDVAR
Linear Val	0.026123	0.031397	-0.146268	0.000008	0.022070	0.022071
DNN Val	0.026124	0.031397	-0.146489	0.000049	0.022068	0.022071
RNN Val	0.026117	0.031381	-0.147867	0.000046	0.022032	0.022071

```
====================================
Precursor [ug/m3]
====================================
```

Model Type	MAE	RMSE	R	Hellinger Distance	STDVAR	Truth STDVAR
Linear Train	0.005334	0.013560	-0.019108	0.000066	0.009846	0.009633
DNN Train	0.005329	0.013556	-0.018503	0.000063	0.009892	0.009633
RNN Train	0.005325	0.013558	-0.019011	0.000046	0.009895	0.009633

```
====================================
Gas [ug/m3]
====================================
```

Model Type	MAE	RMSE	R	Hellinger Distance	STDVAR	Truth STDVAR
Linear Train	0.026378	0.034242	0.011044	0.000458	0.024136	0.024381
DNN Train	0.026379	0.034234	0.012315	0.000491	0.024122	0.024381
RNN Train	0.026359	0.034213	0.012262	0.000530	0.024094	0.024381

```
====================================
Aerosol [ug_m3]
====================================
```

Model Type	MAE	RMSE	R	Hellinger Distance	STDVAR	Truth STDVAR
Linear Train	0.024658	0.030237	0.137119	0.008610	0.022109	0.022071
DNN Train	0.024657	0.030238	0.137591	0.008664	0.022096	0.022071
RNN Train	0.024642	0.030212	0.138055	0.008701	0.022061	0.022071

Members: Zhenyang, Dinara, Diana, and Jahangir*

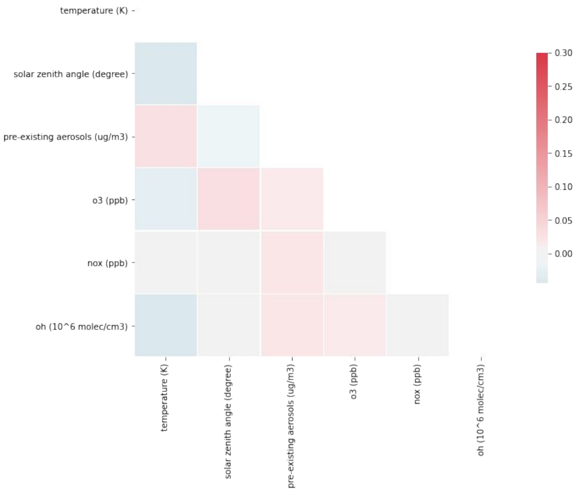
Team 4: GECKO

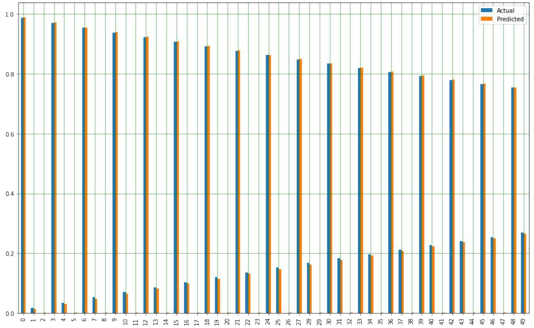
A visualization of your results scores on the problem

Any other cool visualization of results or interpretation of the ML model

Lessons learned/challenges: the main problem was to change dimensionality to perform CNN or LSTM

We were unable to set the box emulator to predict the whole time series (something that need more time for understanding)





Team 4: GECKO

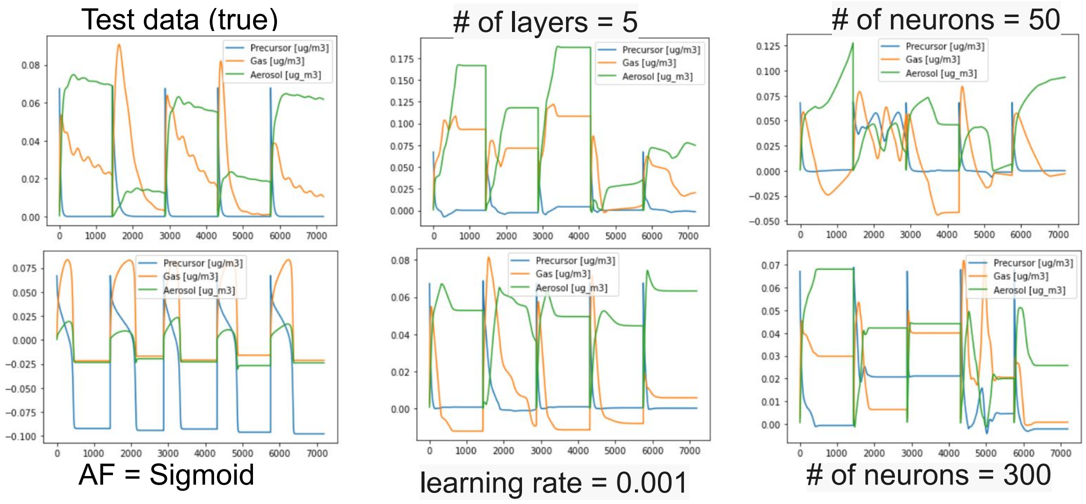
Members: Zhenyang, Dinara, Diana, and Jahangir*

ML methods we've tried during the hackathon:

- Standard and gaussian pdf scaler
- Linear regression and random forest
- PCA (inapplicable though)
- **DNN with different hyperparameter settings**
- LSTM

Default hyperparameters:
# of layers = 2; # of neurons = 100;
AF = relu; learning rate = 0.0001

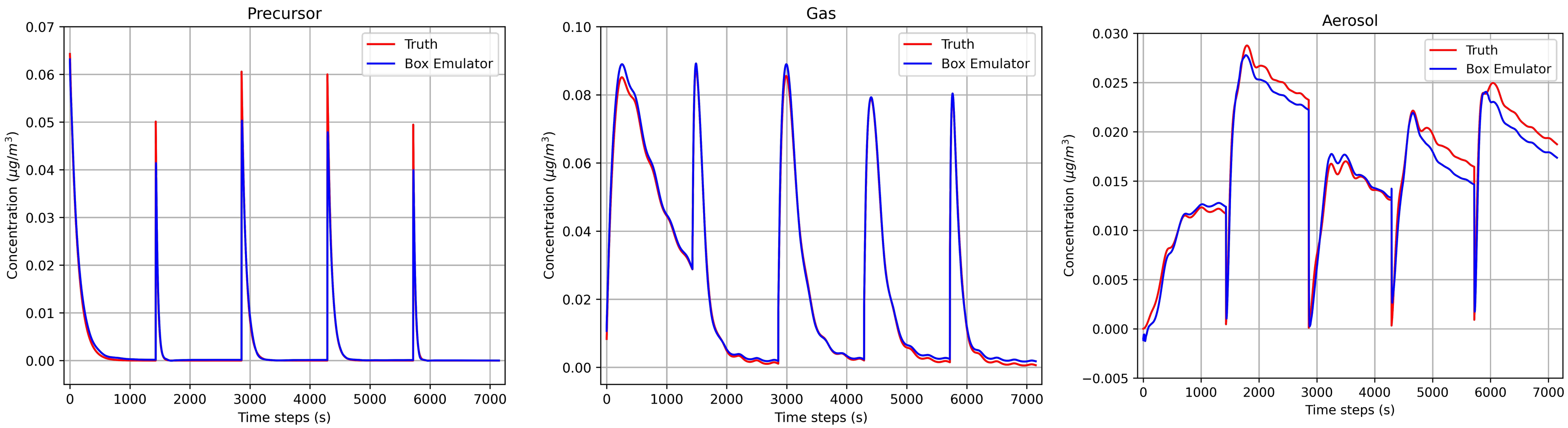
Metrics are shown in the table:
Using LR=0.001 or 5 layers would increase the model score.



Metrics	Default	AF=Sigmoid	**LR=0.001**	50 neurons	300 neurons	**5 layers**
RMSE	0.00657 0.03469 0.04207	0.07531 0.03756 0.06179	**0.00234 0.02533 0.02199**	0.01989 0.03149 0.01892	0.01314 0.01944 0.02083	**0.00281 0.04946 0.07283**
R^2	0.61943 0.03804 0.12799	0.20408 0.35389 0.00600	**0.95694 0.54768 0.30909**	0.17610 0.27722 0.66895	0.39215 0.23105 0.32559	**0.90952 0.19655 0.27384**
H.D.	0.32591 0.26609 0.42771	0.65970 0.53043 0.67728	**0.20106 0.36304 0.32899**	0.21663 0.30813 0.31965	0.38367 0.24422 0.49995	**0.22452 0.35384 0.42799**

Team 33: GECKO

- Team Members: Ethan Kyzivat, Weiming Hu, Hauke Schulz, Chen-Kuang (Kevin) Yang
- Summary of methods tried
  - Random forest (RF)
  - Densely Neural Network (DNN)
  - Long Short-term Memory (LSTM): we decided to use LSTM because it is well-known for time-series prediction
- Data preprocessing
  - Standardization: sklearn “StandardScaler()”
  - Base data: 2,000 experiments (1,440 time-steps per experiment) from GECKO
  - Training/Validation/Testing: 1,400/200/200 experiments
  - Input training data (3-D): [samples, time-steps, features] = [1435*1400, 5, 9]
  - In an essence: we want to use the 9 features from the 5 previous time-steps to inform the information of the next time-step (prediction)



Training the LSTM: **multivariate and one-step prediction**

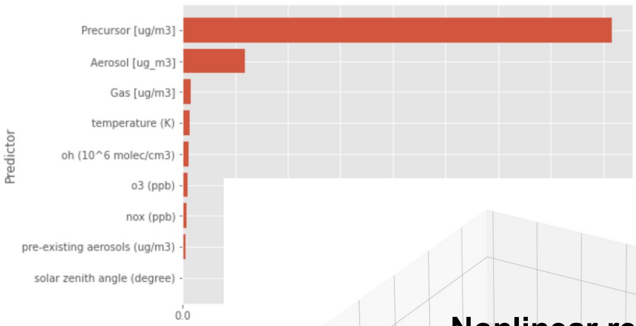
- Hyperparameters
  - Architecture: 64 neurons, ReLU, dropout = 0.2 (prevent overfitting)
  - Training: loss function = MSE, optimizer = Adam, epoch = 5, batch size = 1024, no shuffle on the data
- Evaluation (the graphs above)
  - LSTM + Box Emulator Model
  - Showing the testing result of 5 experiments

Team 14: Gecko

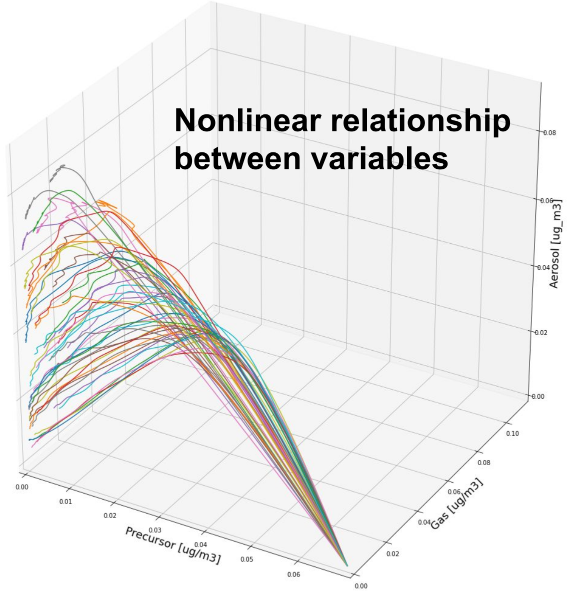
Members: Glenn Liu, Yiluan Song, Laurette Hamlin

- Performed:
  - Exploration: PCA, linear regression, Random Forest, gradient boosting
  - Neural networks: DCNN, SimpleRNN, LSTM
    - Tested sensitivity to various hyperparameters
- Significant:
  - Found and fixed the time lag bug in prepare_data
  - Wrote new data preparation, NN, and box emulators to be compatible with time series analysis
  - Wrote functions for the complete workflow for easy model tuning, comparison and visualization
- Difficulties: Learning Python on the fly!

Detecting relative importance of predictors using RF



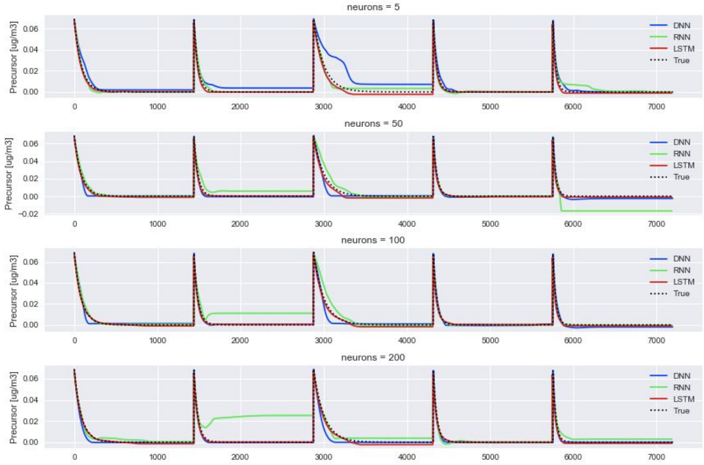
Nonlinear relationship between variables



Team 14: Gecko

Testing hyperparameters in DNN, RNN, and LSTM

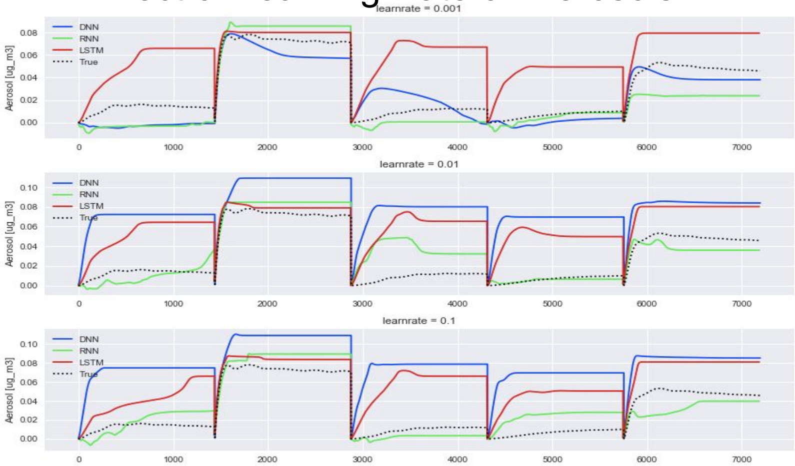
Number of Neurons on Precursor



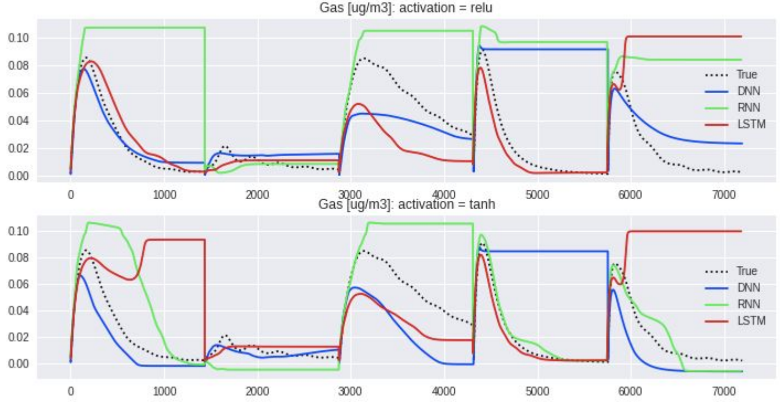
The precursor variable seemed to be less sensitive to choice of hyperparameters.

RNN was the most sensitive to hyperparameters; LSTM was the least sensitive to hyperparameters.

Effect of Learning Rate on Aerosols



Effect of Activation on Gas

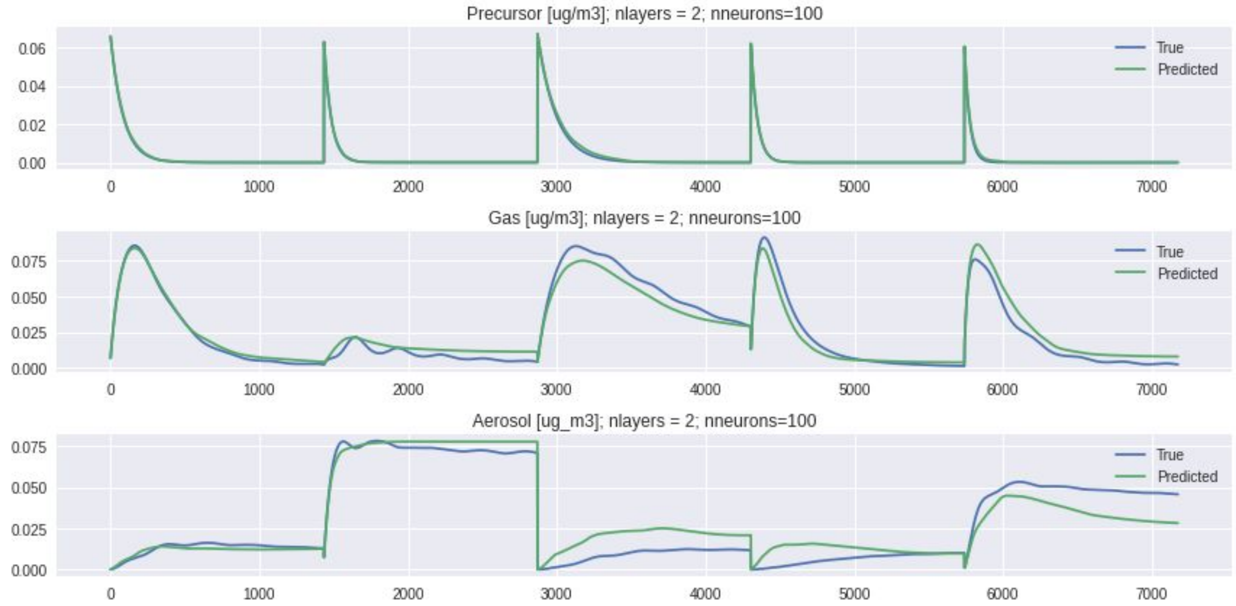


Team 14: Gecko

We could do better given more time and more computational power!

Best model so far:

- Predictors: all 9 input variables at t-1, t-2, t-3, t-4, t-5
- ML method: LSTM
- Architecture: 1 input layer (9 neurons) + 2 hidden layers (100 neurons each) + 1 output layer (3 neurons)
- Activation: “relu”
- Learning rate: 0.001



Model Type	Metric	Variable		
Baseline LSTM		Precursor	Gas	Aerosols
	RMSE	0.00003	0.00012	0.00007
	R^2	0.99999	0.99998	0.99999
	Hellenger Distance	0	0.00002	0.00002
		Precursor	Gas	Aerosols
Box Emulator	RMSE	0.00049	0.00574	0.00873
	R^2	0.99822	0.96352	0.89409
	Hellenger Distance	0.00032	0.0603	0.27265

A Conceptual Note on LSTMs

“I grew up in France where I embraced the language and became fluent in _______ “

Summary

- Results on the base model do not always translate directly to the box emulator.

- Data preparation for RNN/LSTM is not easy!

- LSTM with 5 look-back timesteps seems to be adequate solution to this problem! (a next step would be to see if this model would perform well varying environmental factors)

- **Excellent work everyone!**