Preliminary Results of DSM2 ANNs

11/16/2021

Outline

- Dataset Overview
  - Pre-Processing
- ANN Architecture
- Preliminary Results (Fixed or Trainable Convolution)
  - Mean Squared Error (MSE) Performance
  - Input Variable Importance
  - Visualization of ANN Predictions

Dataset Overview

- **Inputs**: 1282 variables (excluding all "gate"-related variables for now)

Sampling Frequency	Number of input variables sampled at this frequency
15-minute	4
Hourly	3
Daily	757
Lower than daily	518

- **Outputs:** 25 monitoring stations, all in 15-min resolution
- **Data range:** 2000/1/1 – 2020/1/1

Dataset Overview – Pre-Processing

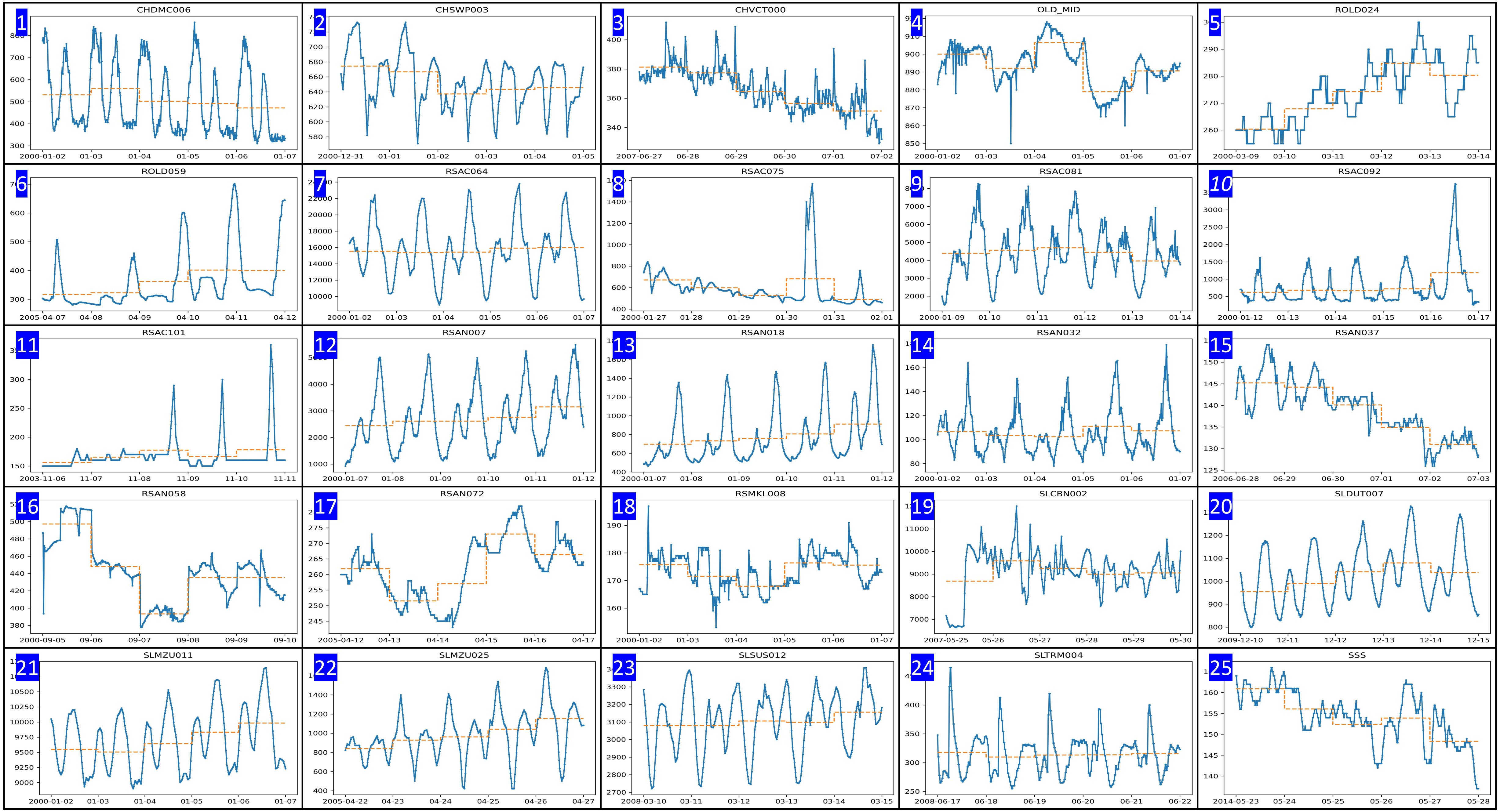
1. Normalization: linearly scale each variable to 0 ~ 1
2. Data synchronization
3. Data Filtering
4. Data Windowing

Dataset Overview – Pre-Processing

1. Normalization
2. Data synchronization
    - Method 1: Save data in 15-minute resolution, mark missing entries by “-1”
        - Issues: huge dataset size; long training time
    - Method 2: Downsample to daily
        - Issue: need to interpolate daily prediction

Data resolution	Precision	CSV file size	Approx. training time of a 3-layer MLP ANN
15-Minute	16 Decimal digits (float64)	8 GB	> 4 Days
15-Minute	3 Decimal digits	3.8 GB	
Daily	3 Decimal digits	56 MB	< 2 Hours

5-day Salinity Plots – Tidal-like Patterns at Most Stations

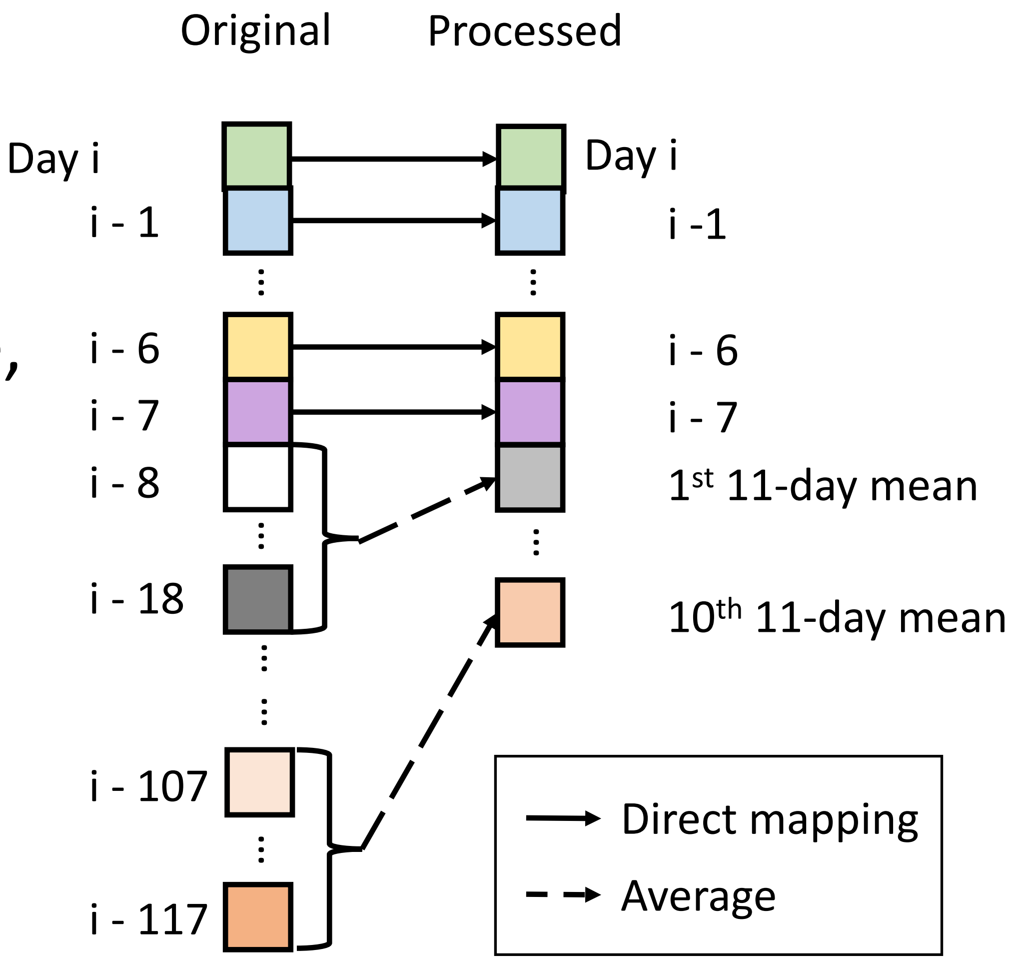


Dataset Overview – Pre-Processing

1. Normalization
2. Data synchronization
3. Data Filtering: eliminate samples without valid salinity values (missing entries or values = -3.4×10^{38}). Use first 80% for training, others for test.
4. Data Windowing

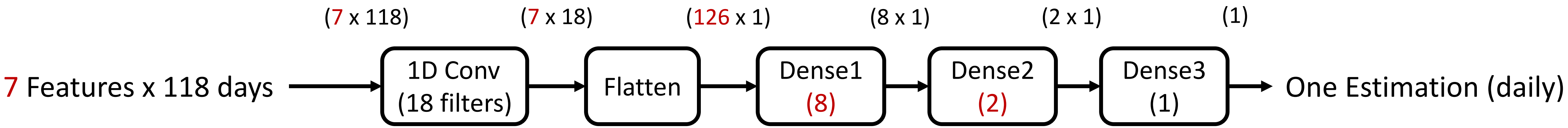
Dataset Overview – Pre-Processing

1. Normalization
2. Data synchronization
3. Data Filtering
4. Data Windowing: of each input variable, we use **118 daily values** to generate **18 features** as ANN inputs



ANN Architecture

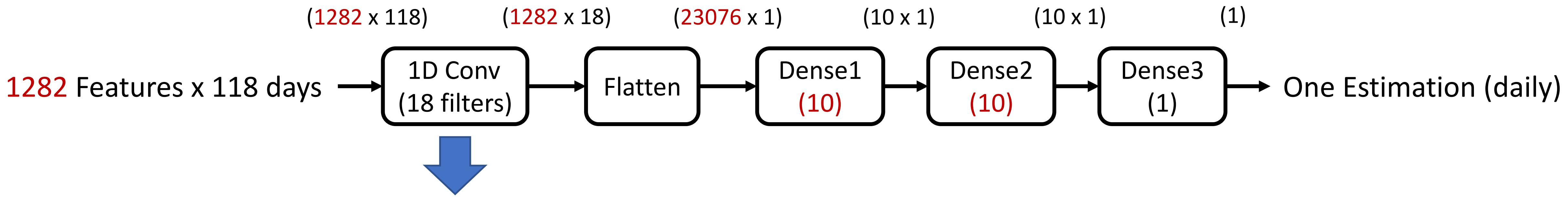
Single-task learning (STL) architecture from previous phase[1]:



[1] Qi, S., et al. *Journal of Water Resources Planning and Management 2021*.

ANN Architecture

Current STL architecture:



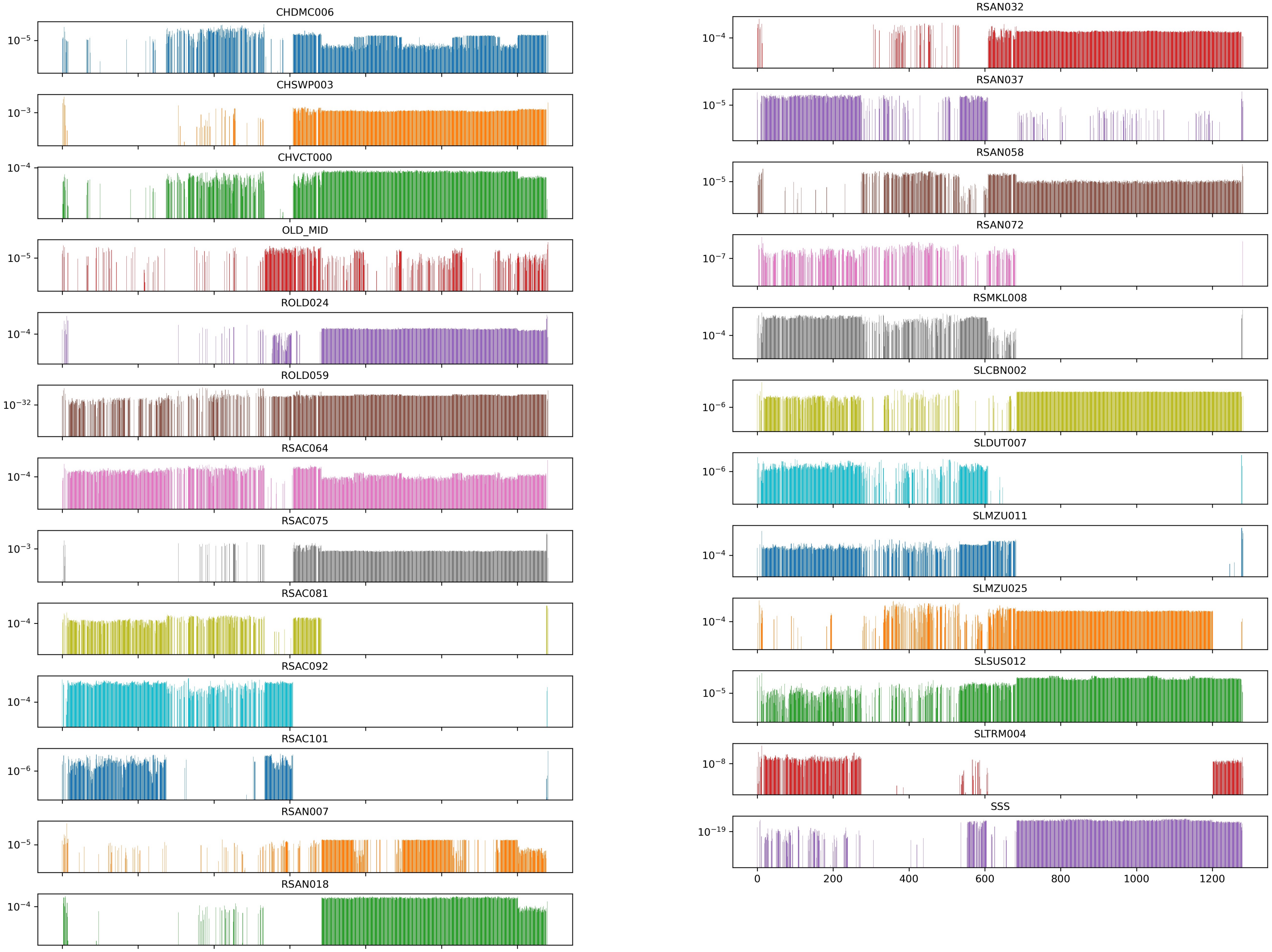
- Fixed: performing pre-defined windowing
- Trained: start from pre-defined windowing, but weightings can change

Preliminary Results – **Fixed** Conv. Layer, MSE on Test Set

Number of valid samples: number of samples after eliminating missing salinity measurements or those days with salinity = -3.4×10^{38}
Test MSE: mean squared error computed on the test split (last 20% of valid samples). Salinity values are linearly normalized to 0 ~ 1

Station	Number of valid samples	Test MSE	Station	Number of valid samples	Test MSE
CHDMC006	7189	0.025	RSAN032	7223	0.0051
CHSWP003	6781	0.029	RSAN037	4069	0.015
CHVCT000	4536	0.029	RSAN058	6999	0.033
OLD_MID	7280	0.066	RSAN072	5381	0.066
ROLD024	7083	**0.039**	RSMKL008	7295	0.0097
ROLD059	5382	0.043	SLCBN002	4581	**0.017**
RSAC064	7118	0.027	SLDUT007	3654	**0.012**
RSAC075	7247	**0.025**	SLMZU011	7063	**0.029**
RSAC081	7232	0.016	SLMZU025	5290	0.035
RSAC092	7238	0.0041	SLSUS012	4316	0.031
RSAC101	5917	0.00020	SLTRM004	4191	**0.0042**
RSAN007	6745	0.022	SSS	700	0.0051
RSAN018	7300	0.0051	/	/	/

Preliminary Results – **Fixed** Conv. Layer, Input Variable Importance



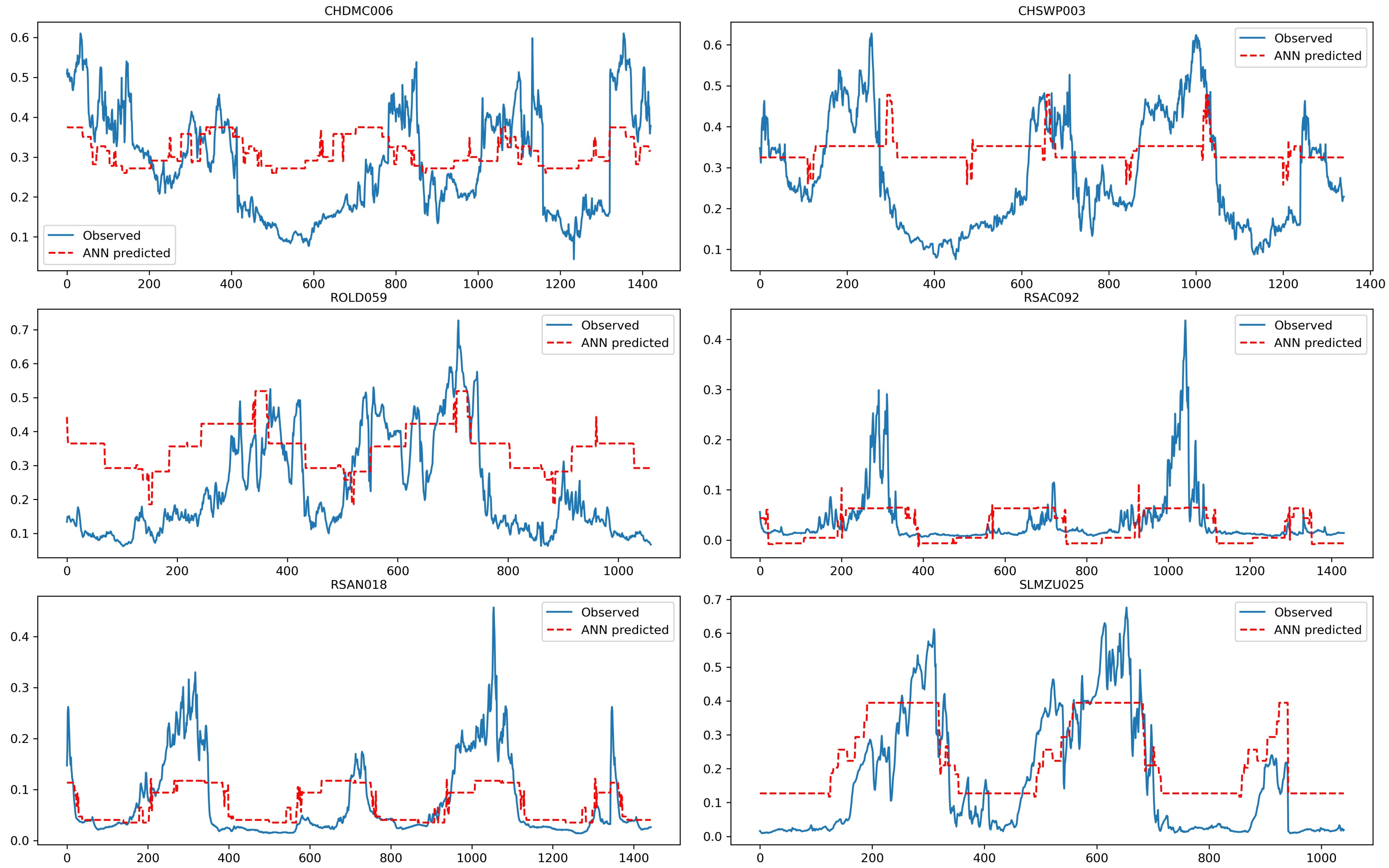
Importance of input variable x_i:

$$\frac{\partial\, output}{\partial x_i},\ 1 \le i \le 1282$$

Observations:

1. Some input variables contribute clearly more than others
2. Some stations' salinity predictions rely on a similar set of input variables.
   -> They can be grouped together to train a multi-task learning ANN
3. For some stations (ROLD059, RSAN072, SSS), the scale of input feature importance is too small to affect the output.

Preliminary Results – **Fixed** Conv. Layer, Observations vs. ANN Predictions



Observations:

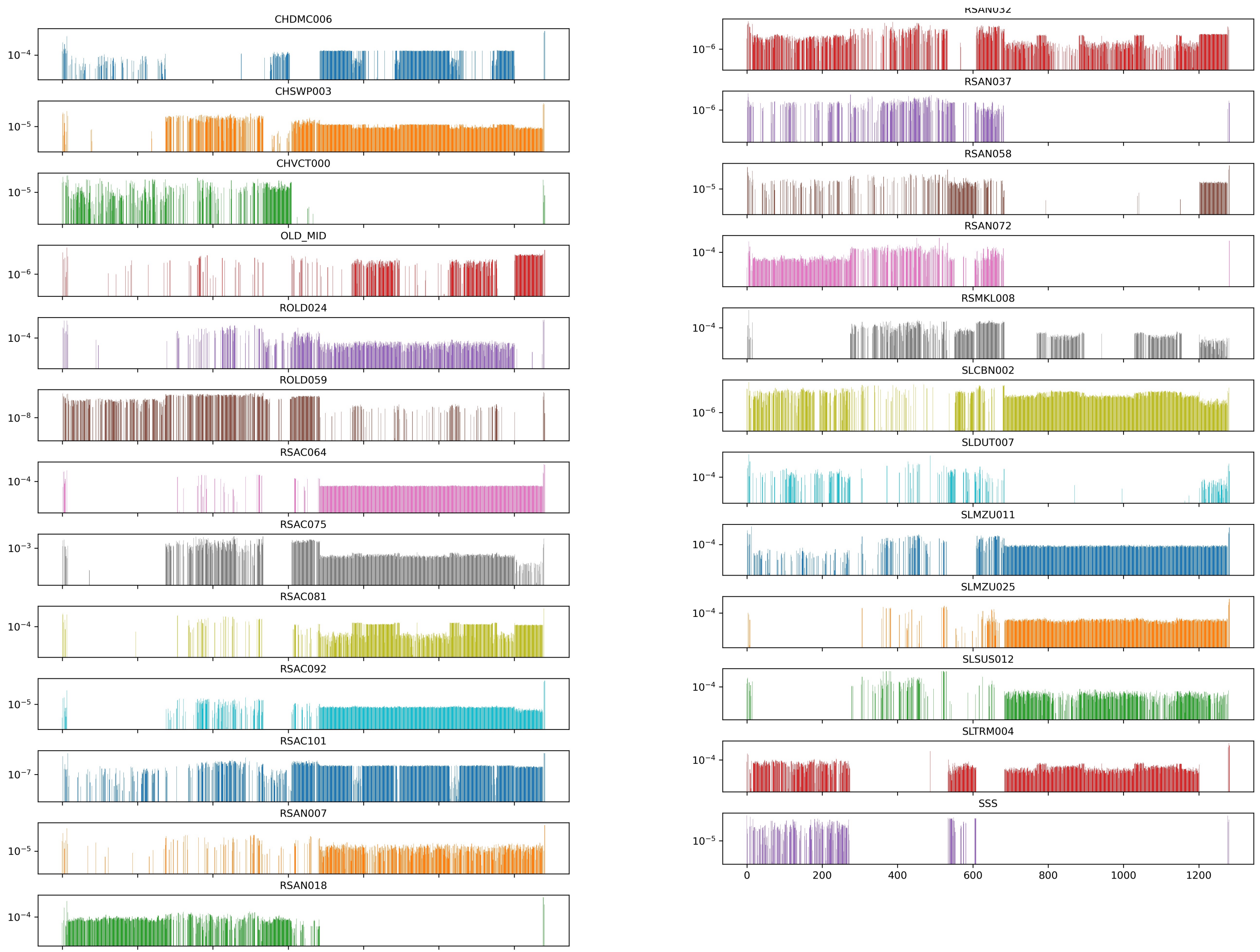
1. ANN predictions can only capture a general trend in salinity
2. Some parts of ANN predictions look like step functions, because the Sigmoid activation function easily saturates (intermediate layers generates binary output)

Preliminary Results – **Trainable** Conv. Layer, MSE on Test Set

Station	Number of valid samples	Test MSE	Station	Number of valid samples	Test MSE
CHDMC006	7189	**0.024**	RSAN032	7223	**0.0048**
CHSWP003	6781	**0.027**	RSAN037	4069	**0.014**
CHVCT000	4536	**0.026**	RSAN058	6999	**0.030**
OLD_MID	7280	**0.065**	RSAN072	5381	**0.063**
ROLD024	7083	**0.036**	RSMKL008	7295	**0.0087**
ROLD059	5382	0.044	SLCBN002	4581	0.019
RSAC064	7118	**0.025**	SLDUT007	3654	0.013
RSAC075	7247	0.026	SLMZU011	7063	0.032
RSAC081	7232	**0.013**	SLMZU025	5290	**0.033**
RSAC092	7238	**0.0040**	SLSUS012	4316	**0.027**
RSAC101	5917	0.0002	SLTRM004	4191	0.0044
RSAN007	6745	0.022	SSS	700	**0.0040**
RSAN018	7300	0.0051	/	/	/

Observations:
Training the convolutional layer reduces MSE for most stations.

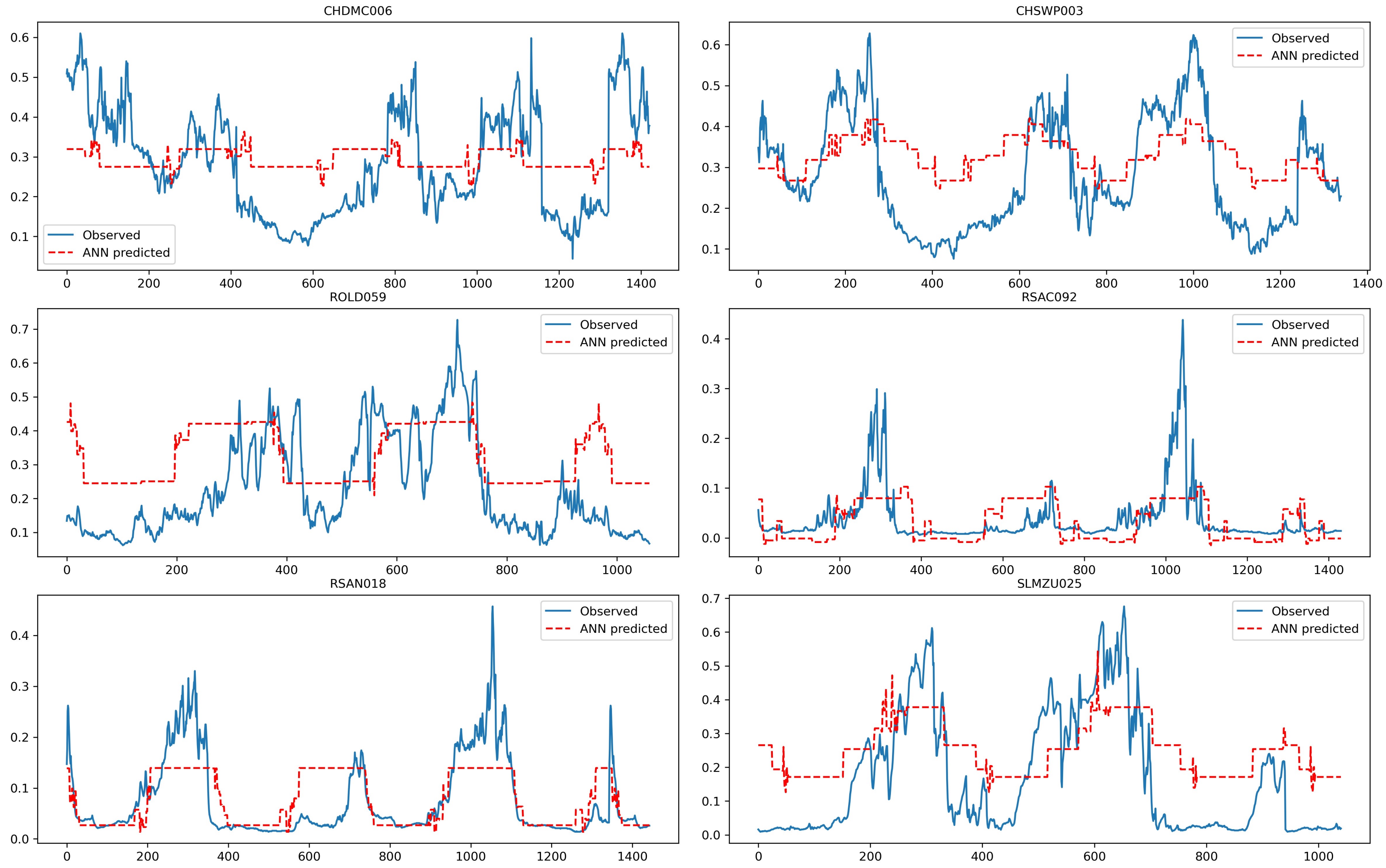
Preliminary Results – **Trainable** Conv. Layer, Input Variable Importance



Observations:

1. The scales of input feature importance are normal (no small values).
2. Some stations input importance plots are very different from “fixed conv layer” case

Preliminary Results – **Trainable** Conv. Layer, Observations vs. ANN Predictions



Observations:

1. Compared with "fixed conv layer" case, there are more variations in ANN predictions. But still not close enough to targets.

Plans

- Solve the Sigmoid saturation issue:
  1. Replace activation functions;
  2. Reduce number of filters in convolutional layer (different data windowing method?);
  3. Reduce number of input features…
  4. Modify architecture
- Group stations by input variable importance, reduce number of input features and train multi-task learning ANNs;
- Try recurrent neural networks.