SMART SOLAR POWER

OTT KEKIŠEV

project owner

CHAN WAI TIK
AAP VARE
JEYHUN ABBASOV
TOFIG BAKHSHIYEV

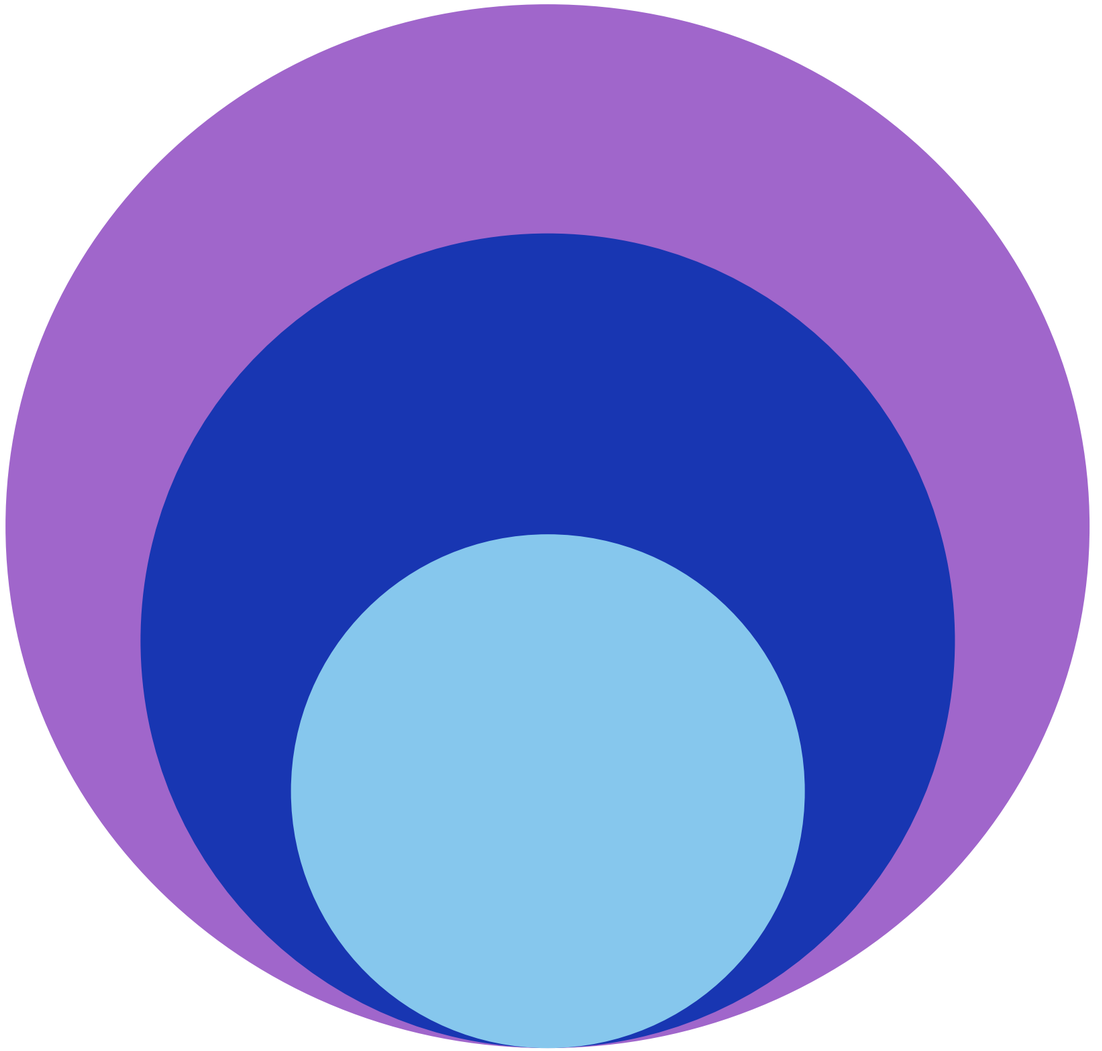
team members

The Problem

- **Time Series Forecast**
  - Tartu
- **Interval Flux Rate on 5min**
  - Model Train on 1 hrs avg
  - Inference on 5mins interval

(Base on the properity of continuous time modelling of birth and death process

- **Washing Machine Scheduling**



Our Approach

1

Data

- Estonia Weather Service [1]
- https://meteo.physic.ut.ee
- From 2019 to 2021

2

Independent Variable

- 1st layer cloud coverage
- Wind direction , speed
- Bearing (base on Tartu)
- Distance(base on Tartu)
- 4 surrounding city
- 3 Timelag (pre 3 hrs)

3

Dependent Variable

Next hours avg Flux (MinMaxTransformed to %)

4

Technique

- Physic Informed Neural Network [2]
- Simple Birth & death process

Training/Validation Results (hour avg)

Epocs
500 epochs - L1loss
converage - 300 epochs
avg validation loss (0.04) ~ 4% in average (max flux 1132)

	Data
Training	2019
Validation	2020
Testing	2021 until Nov

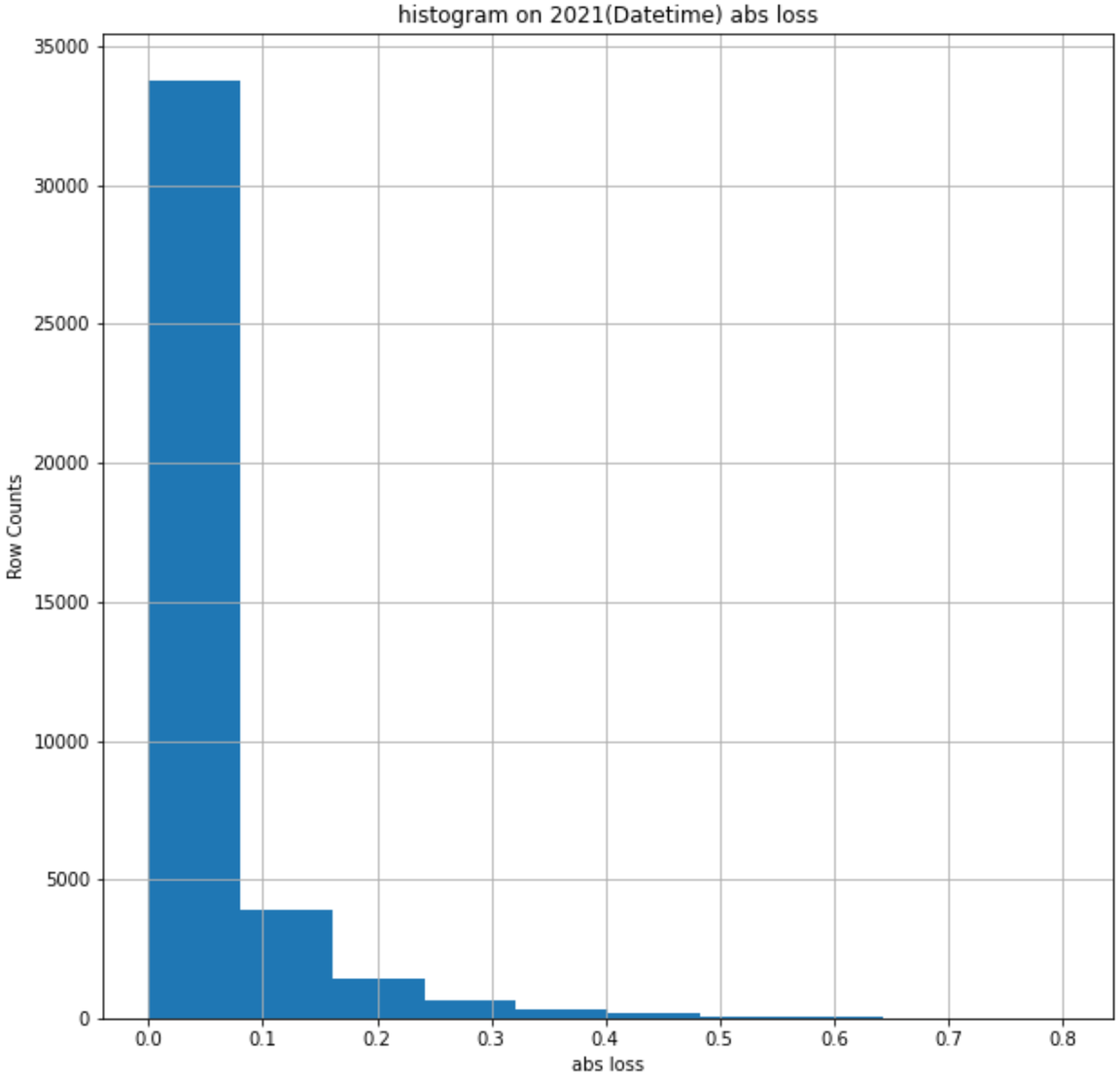
CPINN Loss Plot

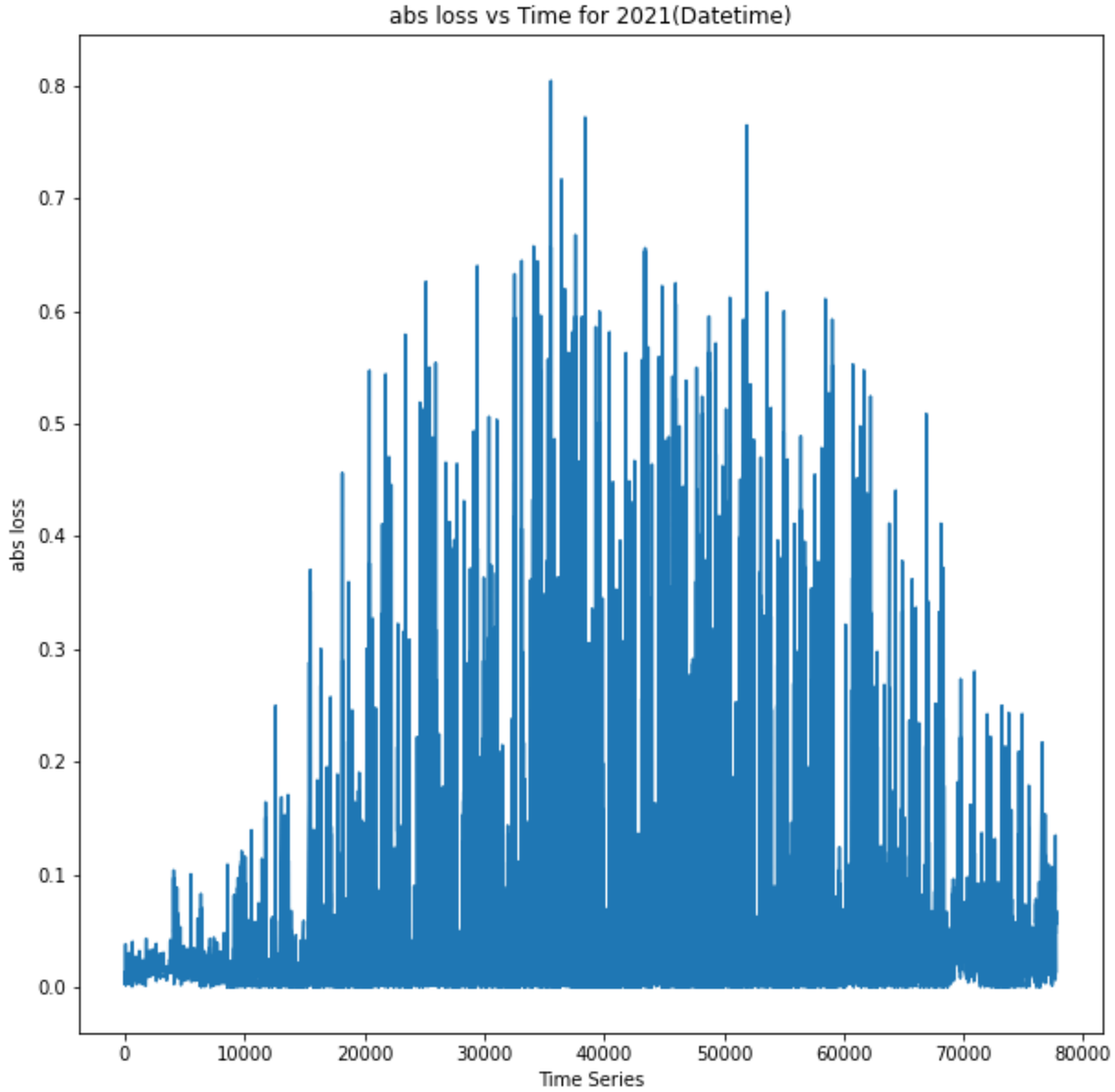
Testing Results on loss (5min Interval -max flux 1132)

All hours(Day+Night)	2019 (77828 rows)	2020 (97837 rows)	2021 (84839)
avg loss	0.024 (2.4%)	0.025 (2.5%)	0.028 (2.8%)
std. deviation	0.05 (5%)	0.05 (5%)	0.057 (5.7%)
median	0.0039 (0.4%)	0.0037 (0.37%)	0.006 (0.6%)
75% quantile	0.028 (2.8%)	0.027 (2.7%)	0.032 (3.2%)
Max	0.735 (73%)	0.8 (80%)	0.8 (80%)

Only Date time	2019	2020	2021
avg loss	0.047 (4.7%)	0.05 (5%)	0.052 (5.2%)
std. deviation	0.0615 (6.15%)	0.069 (6.9%)	0.071 (7.1%)
median	0.029 (3%)	0.029 (3%)	0.030 (3%)
75% quantile	0.056 (5.6%)	0.059 (5.9%)	0.059 (5.9%)
Max	0.735 (73%)	0.8 (80%)	0.8 (80%)

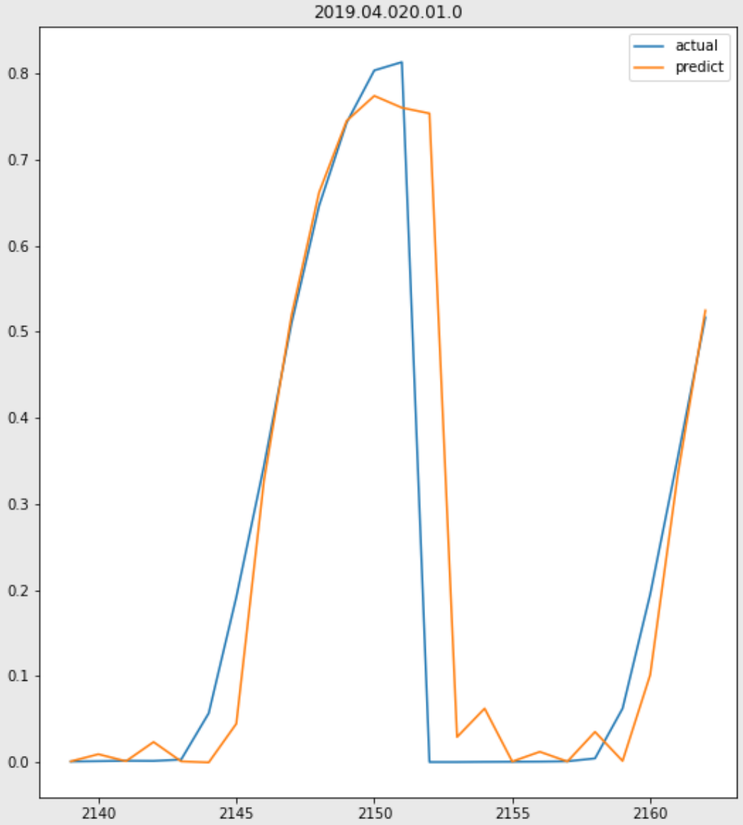
Testing Results on loss (5min Interval -max flux 1132)

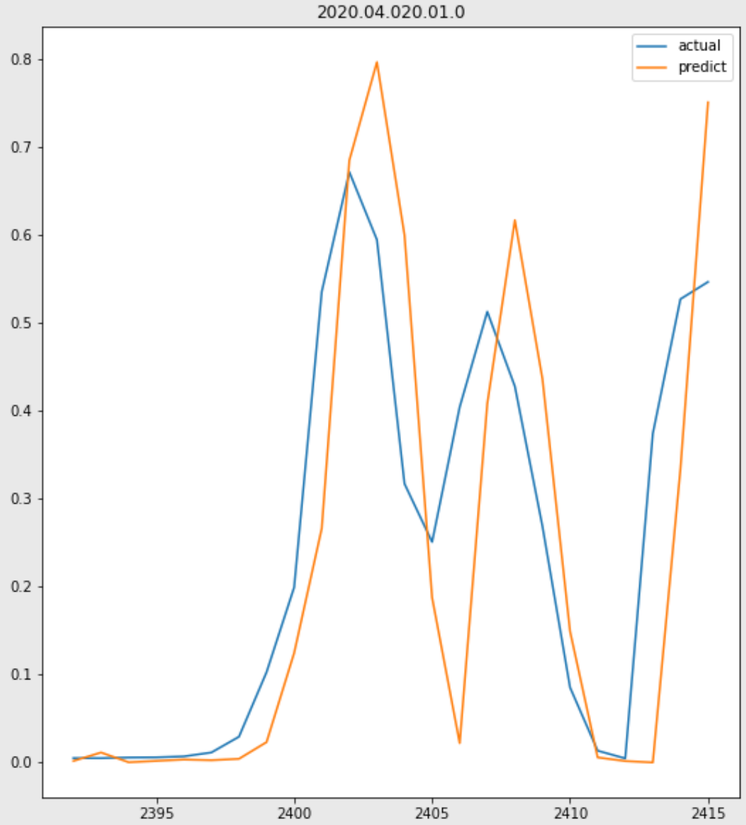


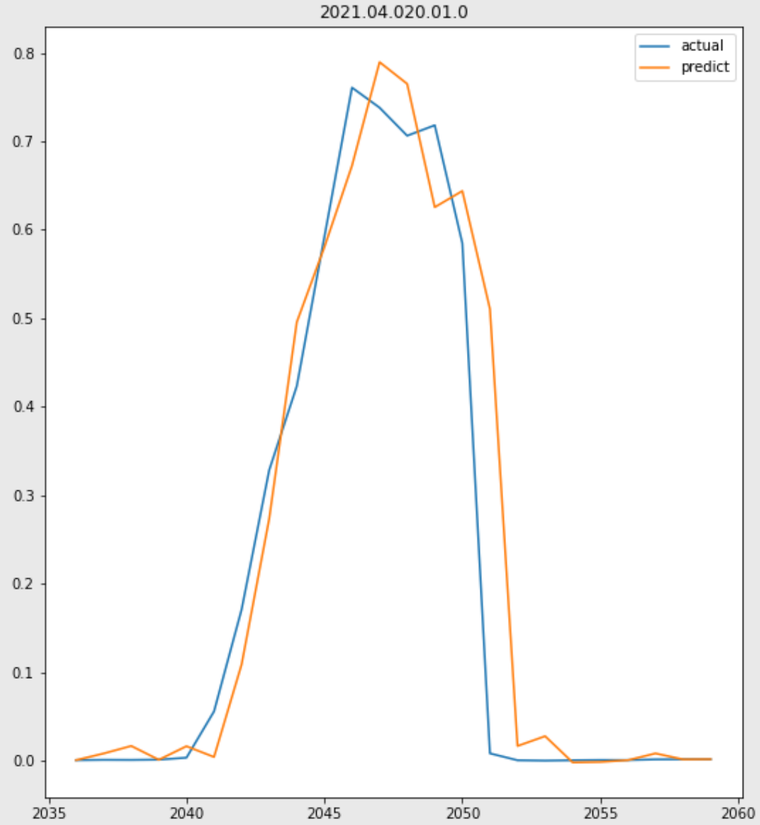


Expectations vs Results (Hours avg)

- random selected date
- 24 hours intervals

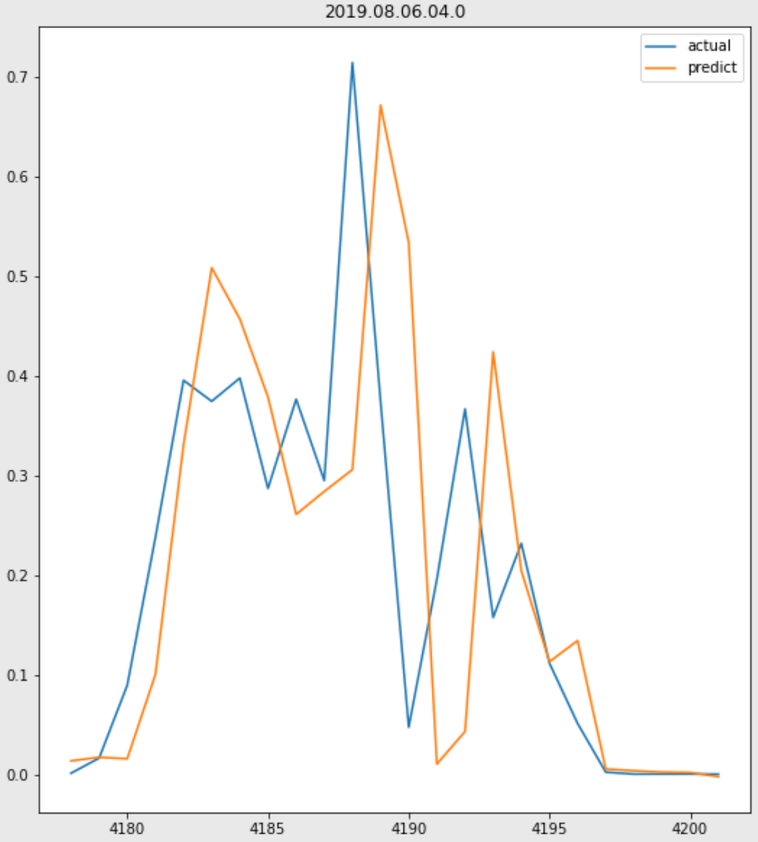


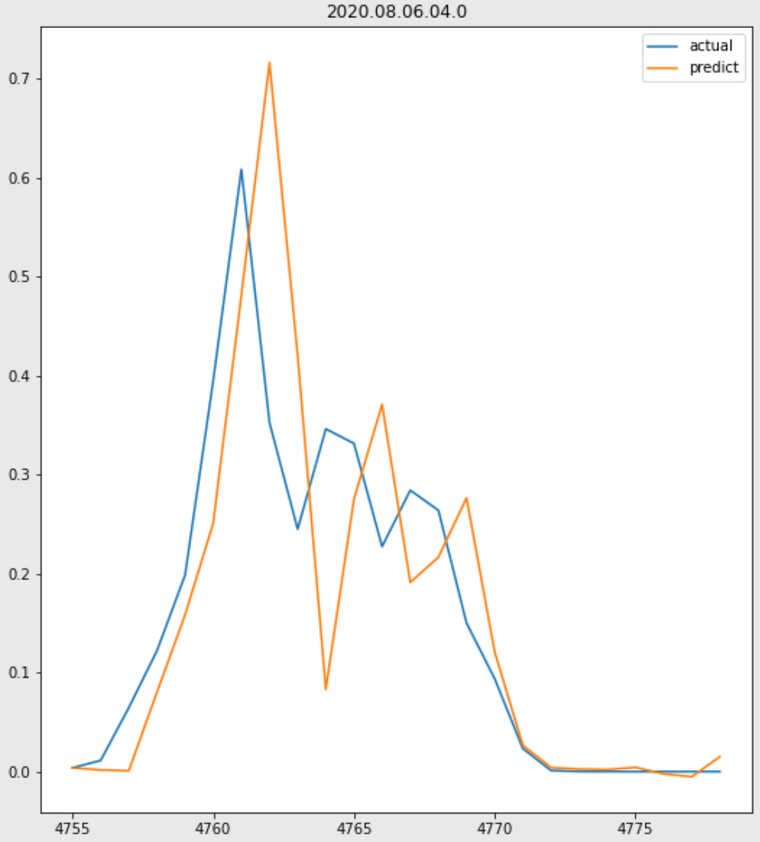


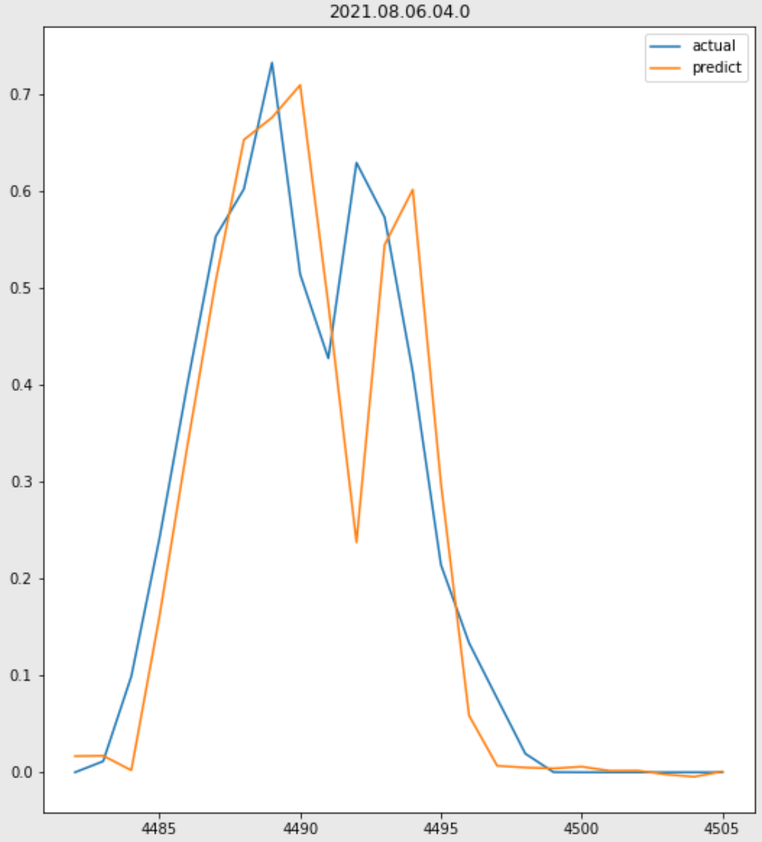


Expectations vs Results (Hours avg)

- random selected Month,date,hour
- 24 hours intervals

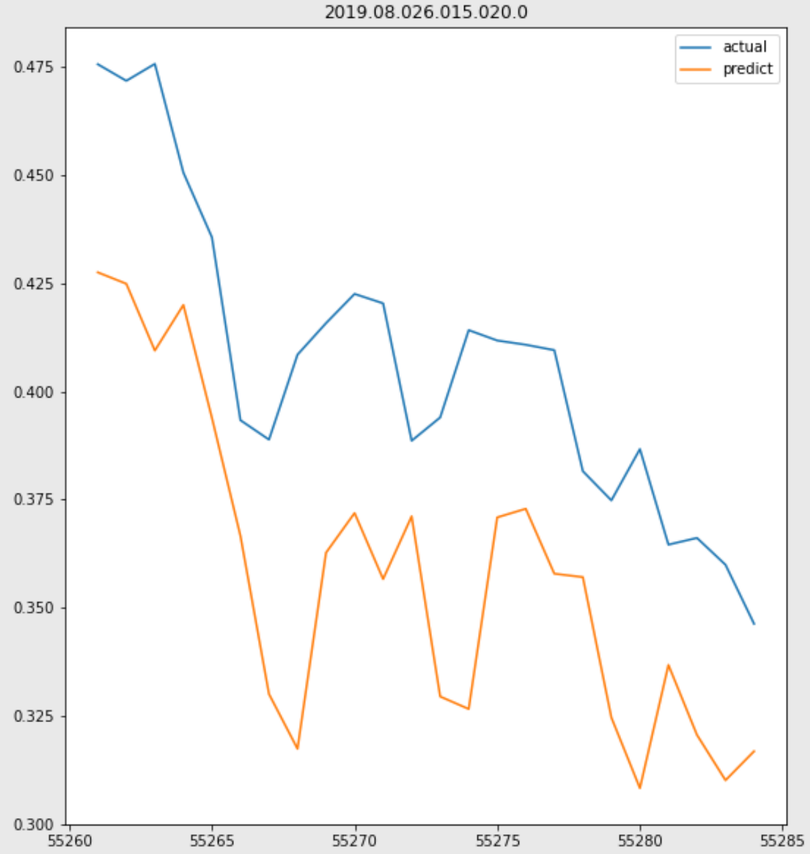


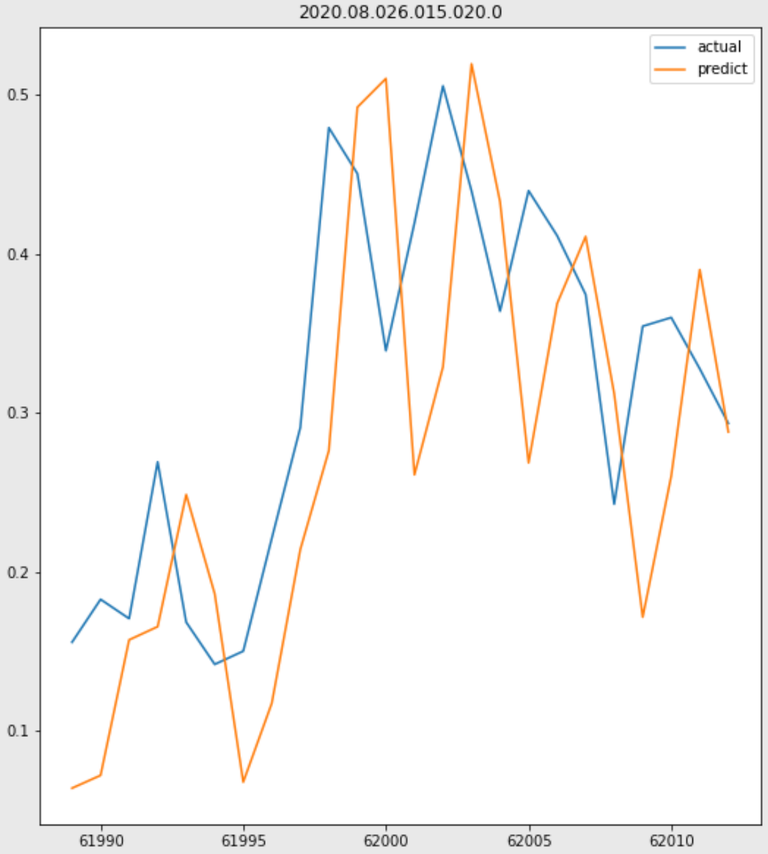


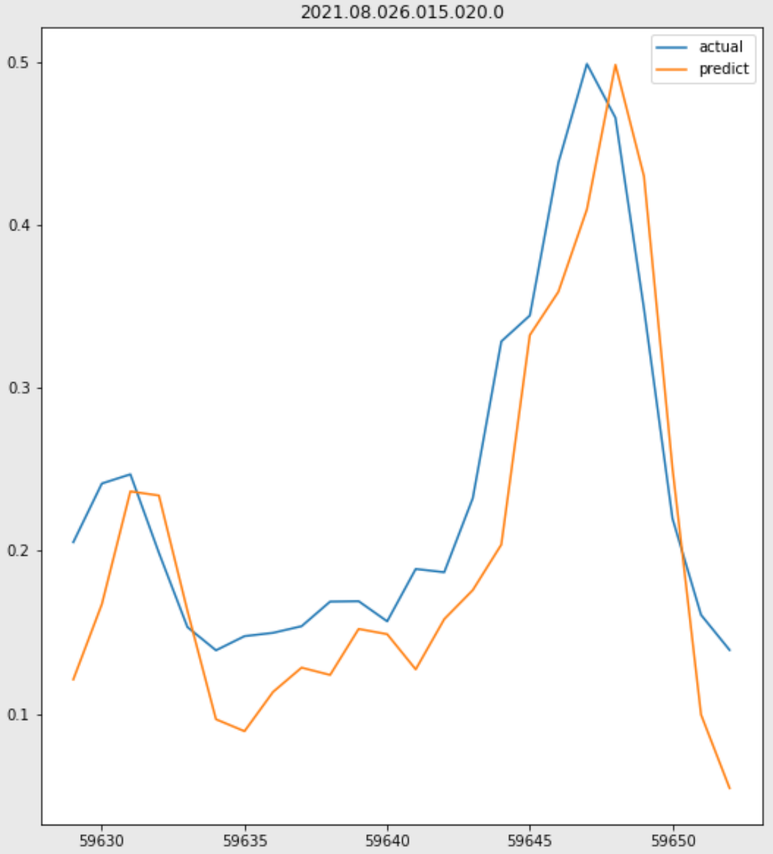


Expectations vs Results (5mins)

- random selected Month,date,hours,mins
- 2 hours intervals

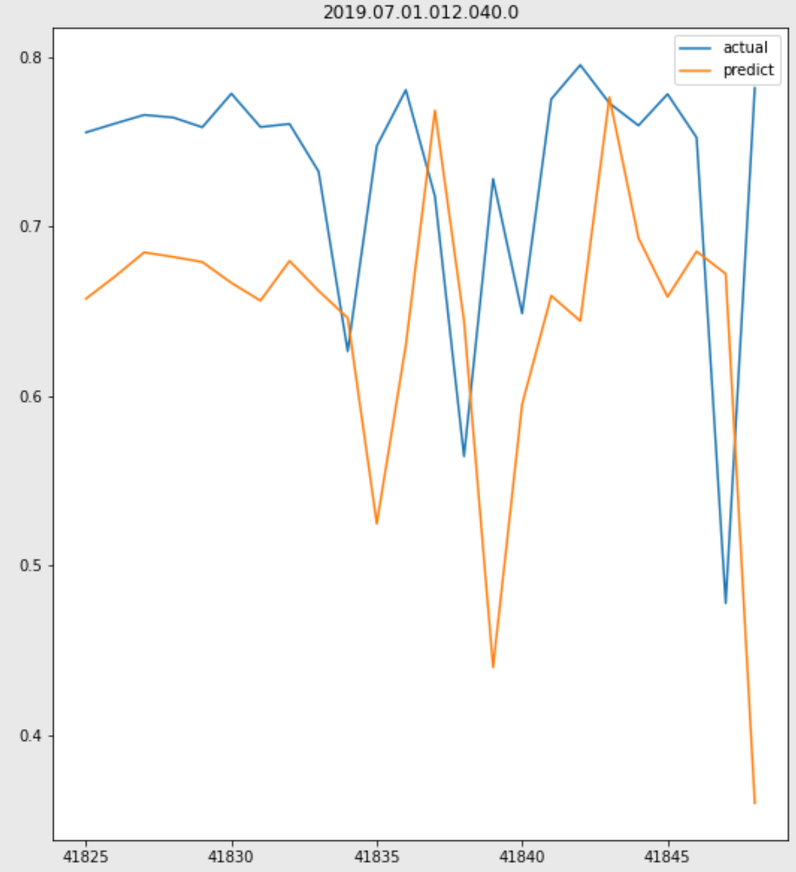


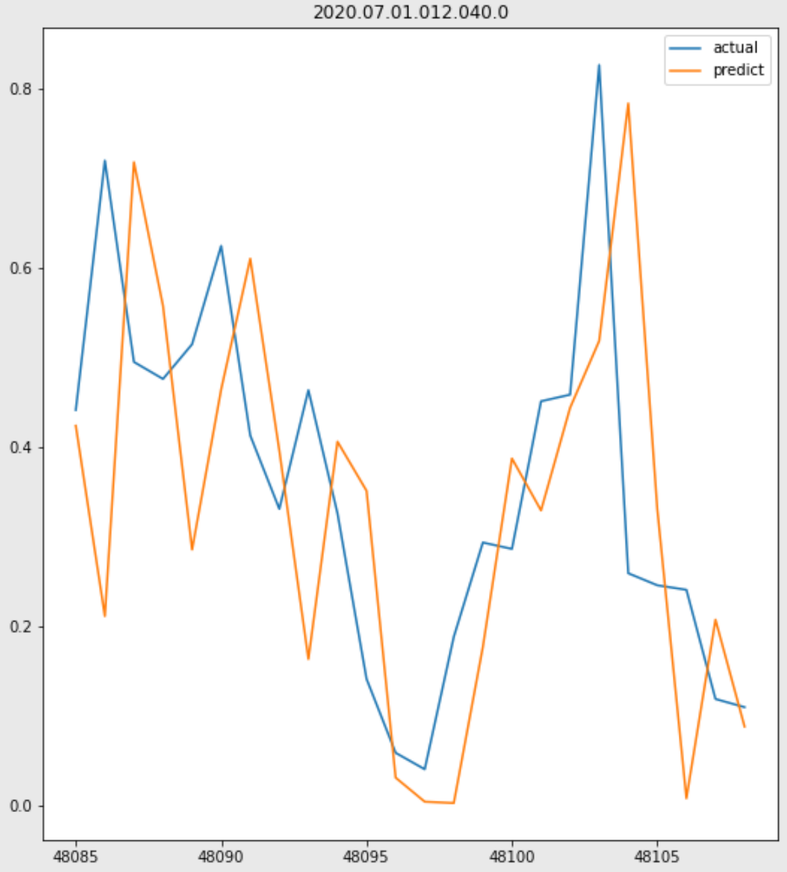


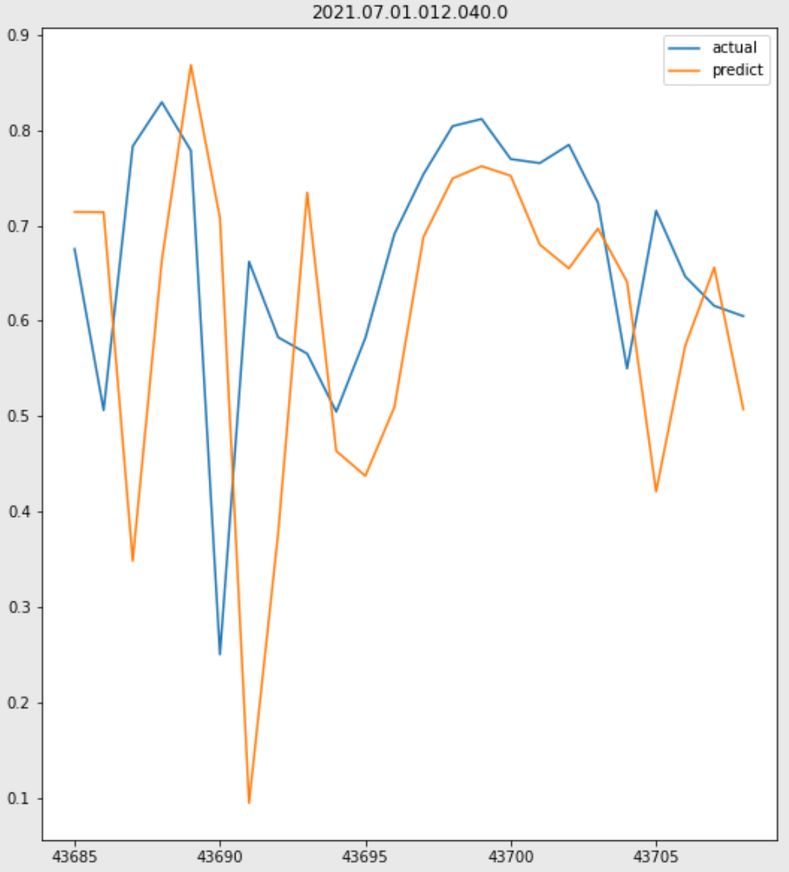


Expectations vs Results (5mins)

- random selected Month,date,hours,mins
- 2 hours intervals







We've learned

Soft Skills

- Out of box thinking
- Make a comporise from different opinions
- How to interact with people from different cultures

Technical Skills

- Pytorch
- Data Mining
- Basic Idea on Time Series Forecast
- Physic Informed Neural Network [2]
- Data Explorartion and Transformation (pandas)

References

[1] Estonia Weather Service

https://www.ilmateenistus.ee/?lang=en
https://meteo.physic.ut.ee/

[2] Physic Informed Neural Network

https://benmoseley.blog/my-research/so-what-is-a-physics-informed-neural-network/
https://github.com/jayroxis/PINNs/blob/master/Burgers%20Equation/Burgers%20Inference%20(PyTorch).ipynb
https://maziarraissi.github.io/PINNs/
Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations
Physics Informed Deep Learning (Part II)Data-driven Discovery of Nonlinear Partial Differential Equations

[3] Smart Solar Power Project Source Code

https://github.com/tik65536/2021UTML_Solar

Reference2

hourly flux std. deviation (base on 5mins data) 2019

```
]: 1 stat[('Irradiation flux',   'std')].describe()
```

```
]: count    8453.000000
   mean       23.878684
   std        42.418104
   min         0.000000
   25%         0.270648
   50%         3.291532
   75%        31.456848
   max       347.276330
   Name: (Irradiation flux, std), dtype: float64
```

Correlation of cloud coverage and flux

```
]: 1 pd.DataFrame(np.vstack((resultI1,resultA2)).T).corr() #Monthly Correlation (avg 2019)
```

]:

	0	1
0	1.000	-0.889
1	-0.889	1.000

```
]: 1 pd.DataFrame(np.vstack((resultI2,resultA1)).T).corr() #Daily Correlation (avg 2019)
```

]:

	0	1
0	1.000000	-0.456883
1	-0.456883	1.000000

```
]: 1 pd.DataFrame(np.vstack((resultI3,resultA3)).T).corr() # Yearly Correlation (avg 2019)
```

]:

	0	1
0	1.000000	-0.895039
1	-0.895039	1.000000

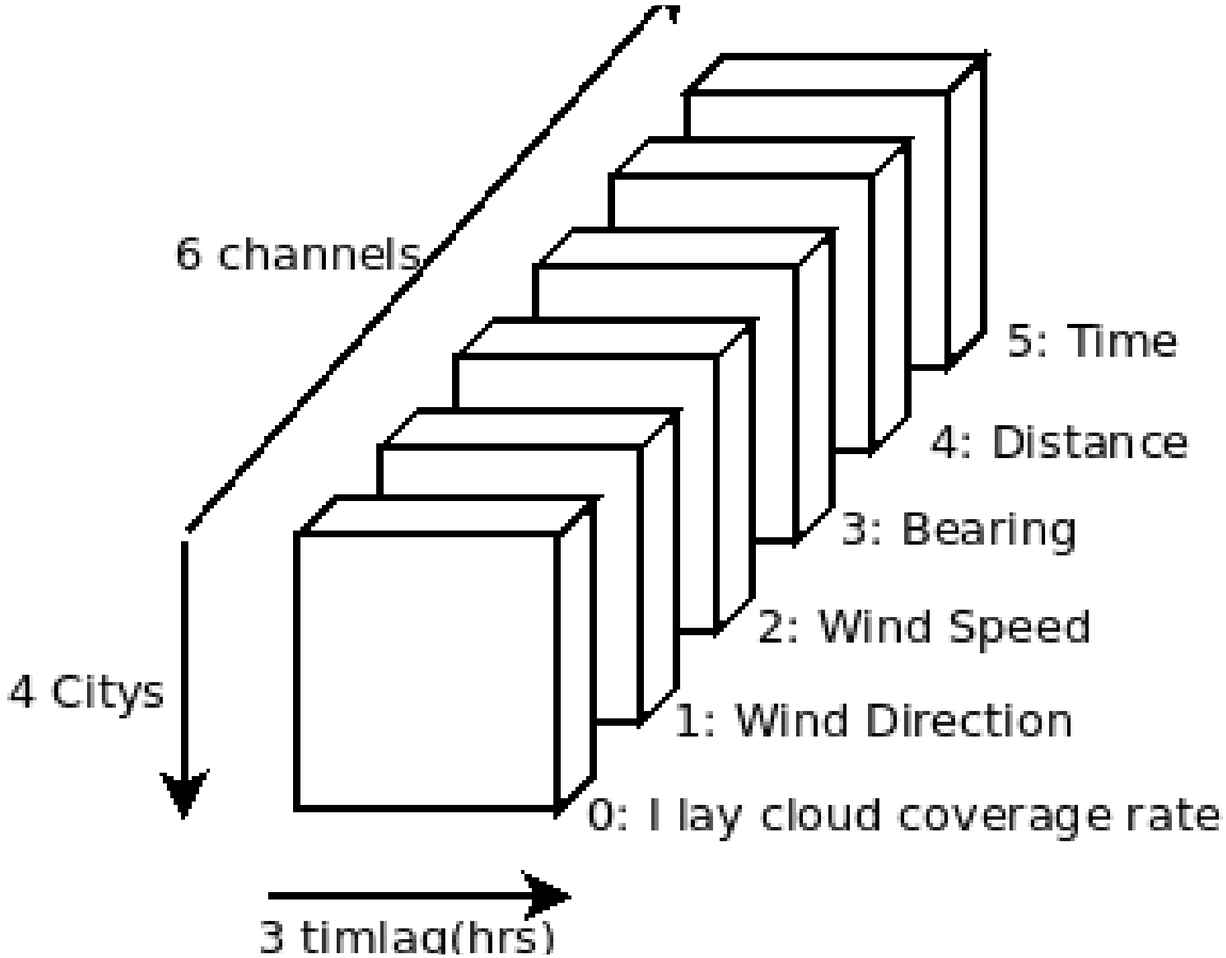
Reference3

Correlation of cloud coverage between Tartu, Turi,Voru,Vaike,Vilsandi

```
1 data2019[['I_layer_amount','I_layer_amount_Turi','I_layer_amount_Voru','I_layer_amount_Vaike','I_layer_amount_Vilsandi']].corr()
```

	I_layer_amount	I_layer_amount_Turi	I_layer_amount_Voru	I_layer_amount_Vaike	I_layer_amount_Vilsandi
I_layer_amount	1.000000	0.605705	0.659917	0.603170	0.538266
I_layer_amount_Turi	0.605705	1.000000	0.526393	0.660447	0.596965
I_layer_amount_Voru	0.659917	0.526393	1.000000	0.528558	0.486318
I_layer_amount_Vaike	0.603170	0.660447	0.528558	1.000000	0.541491
I_layer_amount_Vilsandi	0.538266	0.596965	0.486318	0.541491	1.000000

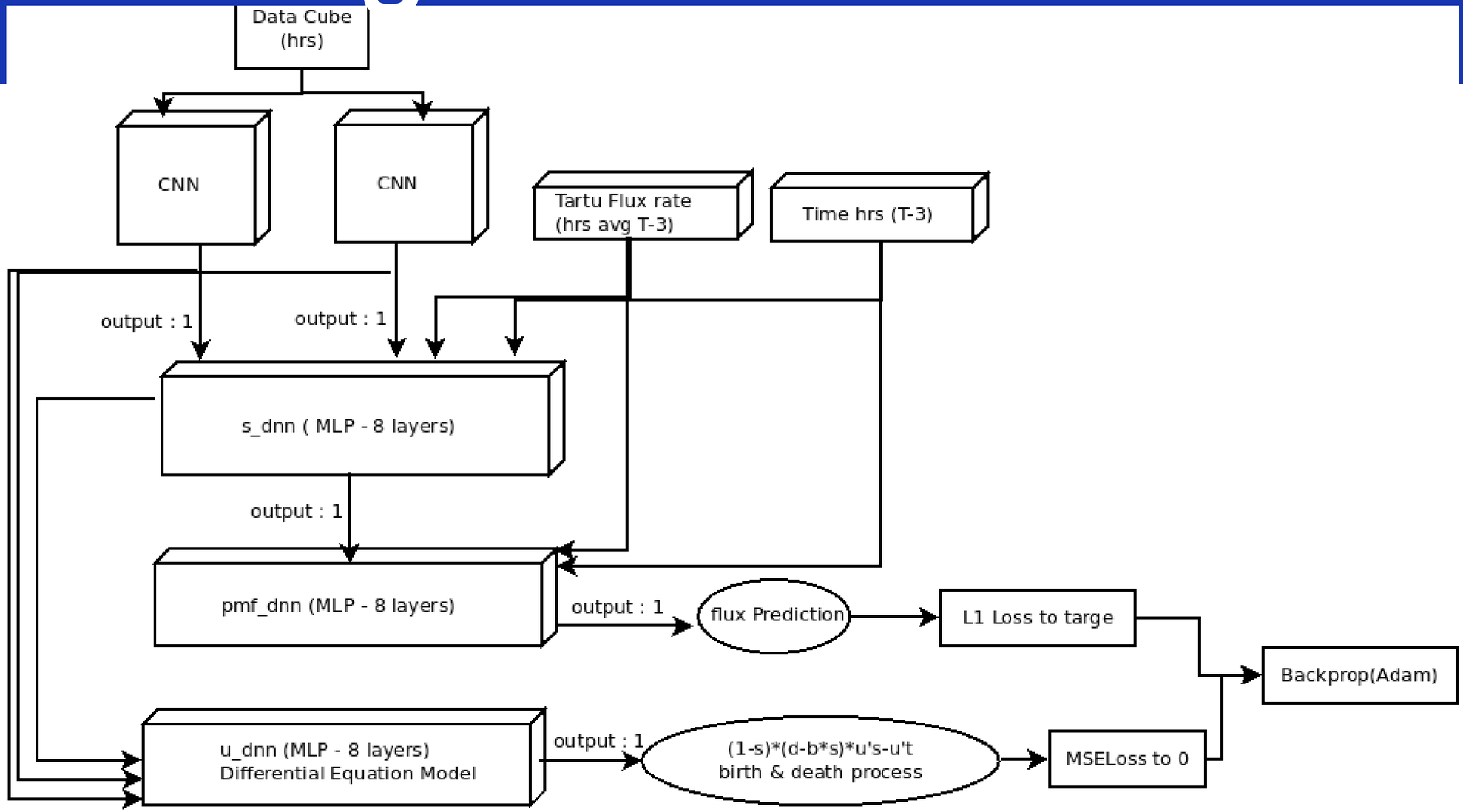
CPINN model (Data Cube)



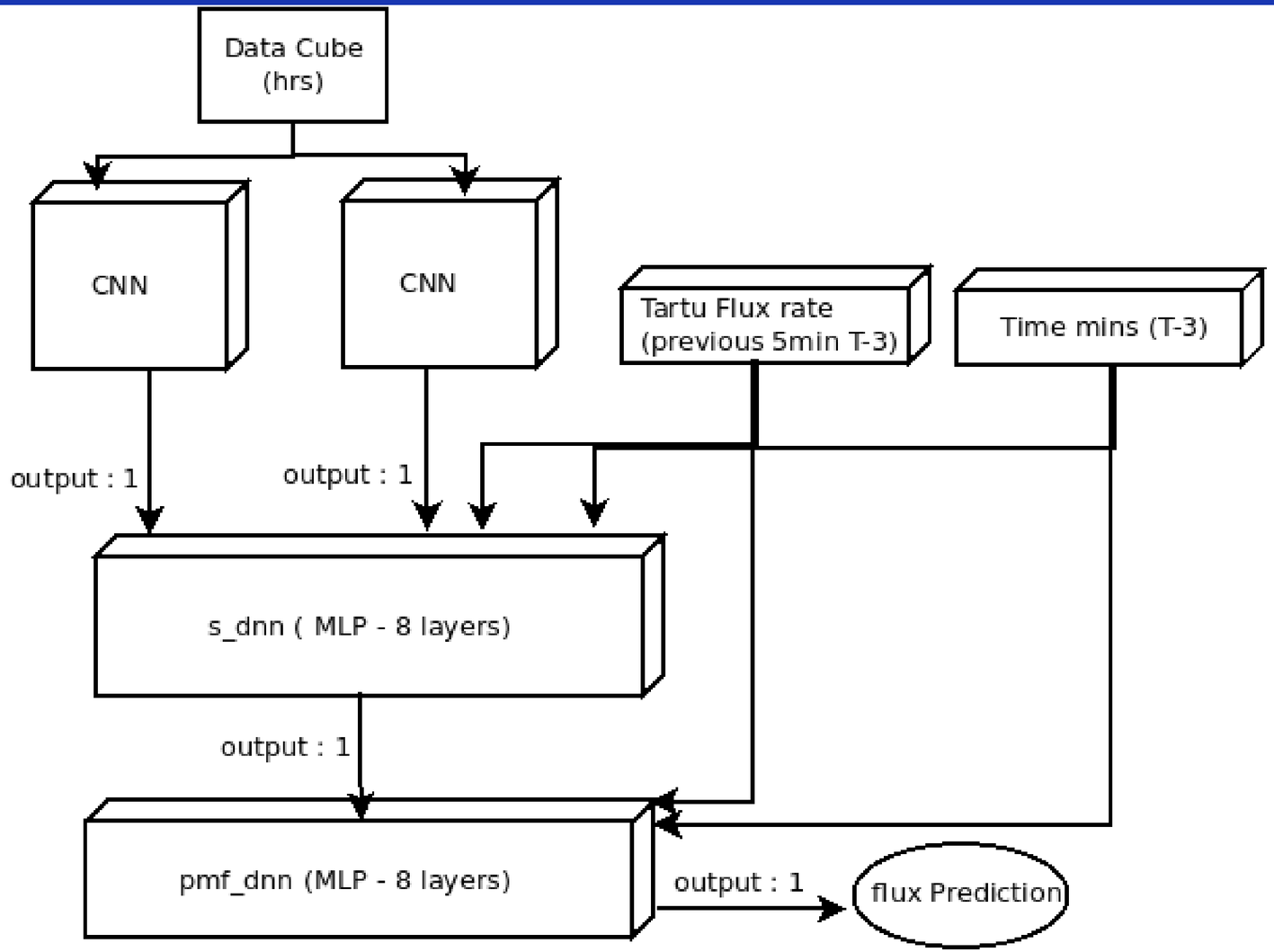
Time Encoding (Hrs):
Time = Month-1 + hrs*(1/23)

Time Encoding (mins):
Time = Month-1 + mins*(1/23/60)

CPINN model (Network Structure) Training



CPINN model (Network Structure) Predict (5mins)



CPINN model (Network Summary)

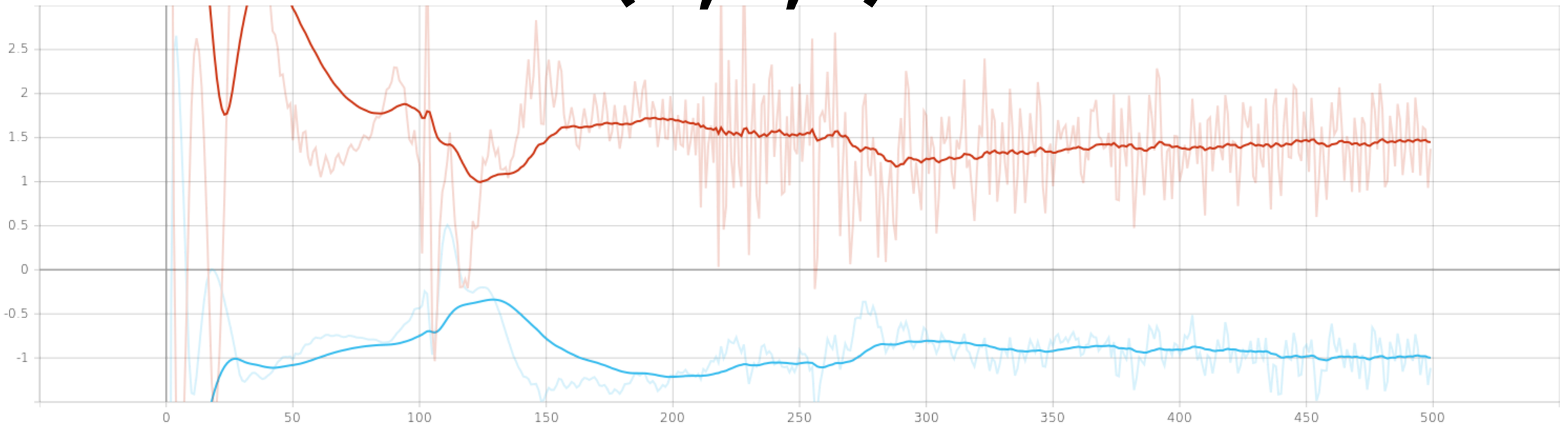
CNN Summary

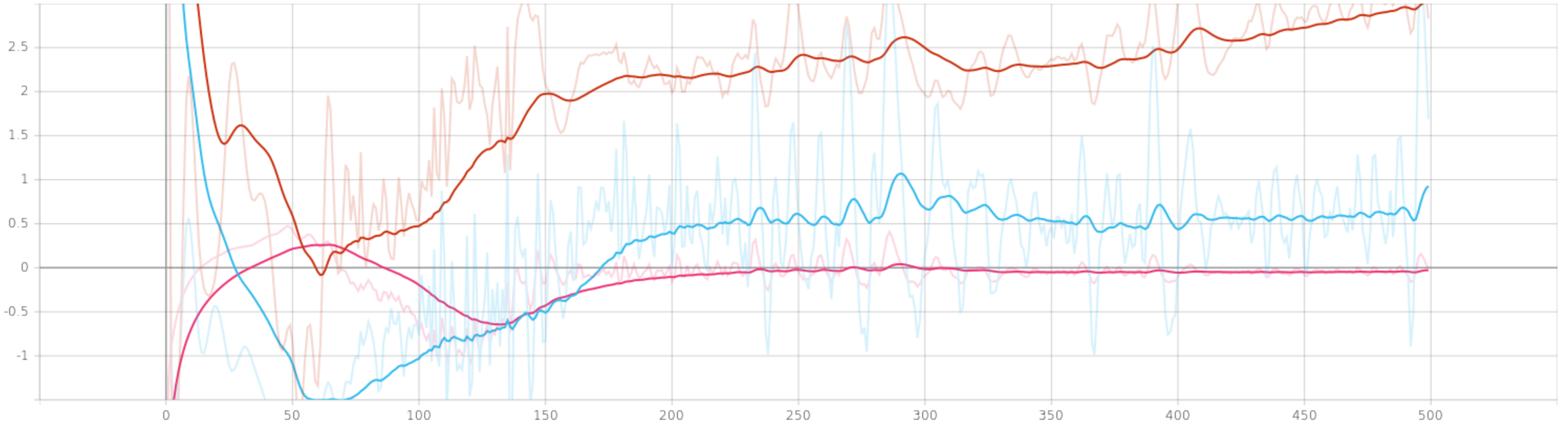
```
Training Set birth data len 7067
Validation Set birth data len 7885
CDNN(
  (activation): ReLU()
  (dnn1): Sequential(
    (0): Conv2d(6, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU()
    (4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): ReLU()
    (6): Flatten(start_dim=1, end_dim=-1)
    (7): Linear(in_features=1536, out_features=1024, bias=True)
    (8): ReLU()
    (9): Linear(in_features=1024, out_features=256, bias=True)
    (10): ReLU()
    (11): Linear(in_features=256, out_features=1, bias=True)
  )
)
```

MLP Summary

```
Training Set birth data len 7067
Validation Set birth data len 7885
DNN(
  (layers): Sequential(
    (layer_0): Linear(in_features=17, out_features=40, bias=True)
    (activation_0): ReLU()
    (layer_1): Linear(in_features=40, out_features=40, bias=True)
    (activation_1): ReLU()
    (layer_2): Linear(in_features=40, out_features=40, bias=True)
    (activation_2): ReLU()
    (layer_3): Linear(in_features=40, out_features=40, bias=True)
    (activation_3): ReLU()
    (layer_4): Linear(in_features=40, out_features=40, bias=True)
    (activation_4): ReLU()
    (layer_5): Linear(in_features=40, out_features=40, bias=True)
    (activation_5): ReLU()
    (layer_6): Linear(in_features=40, out_features=40, bias=True)
    (activation_6): ReLU()
    (layer_7): Linear(in_features=40, out_features=40, bias=True)
    (activation_7): ReLU()
    (layer_8): Linear(in_features=40, out_features=1, bias=True)
  )
)
```

CPINN model (B,D,S)





CPINN model (B,D,S)