European Network on New Sensing Technologies for Air Pollution Control and Environmental Sustainability - *EuNetAir*

COST Action TD1105

3rd International Workshop *EuNetAir* on

New Trends and Challenges for Air Quality Control

University of Latvia - Faculty of Geography and Earth Sciences

Riga, Latvia, 26 - 27 March 2015

KERNEL NETWORKS FOR LEARNING FROM SENSOR DATA

Roman Neruda, P. Vidnerová V. Kůrková

MC Substitute, roman@cs.cas.cz

Institute of Computer Science, Academy of Sciences of the Czech Republic, Prague

Outline

- Kernel networks
  - Theoretical properties
  - Composite and product kernels
- Learning algorithms
  - Beyond parameter setting
  - Evolutionary computing
- Data
- Results
- Discussion
  - Challenges


MODELS

Learning from data

- Problem:

  Given set of data samples:

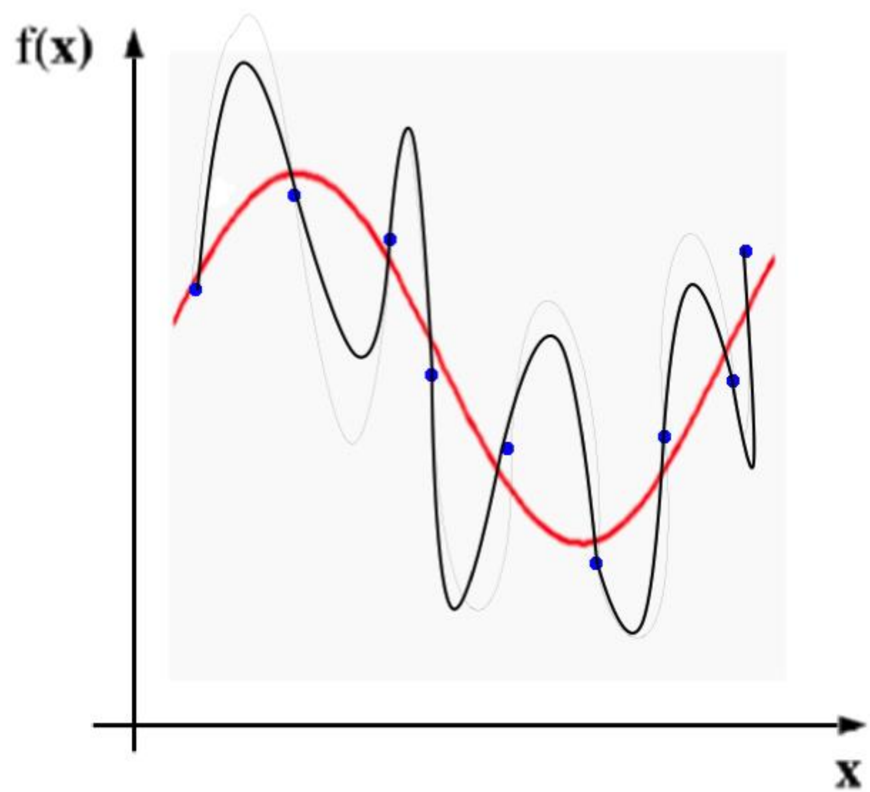
  $\{(\boldsymbol{x}_i, y_i) \; \varepsilon \; R^d x R; \; i=1, \ldots, N\}$

  Recover the unknown function

  $f: R^d \rightarrow R$

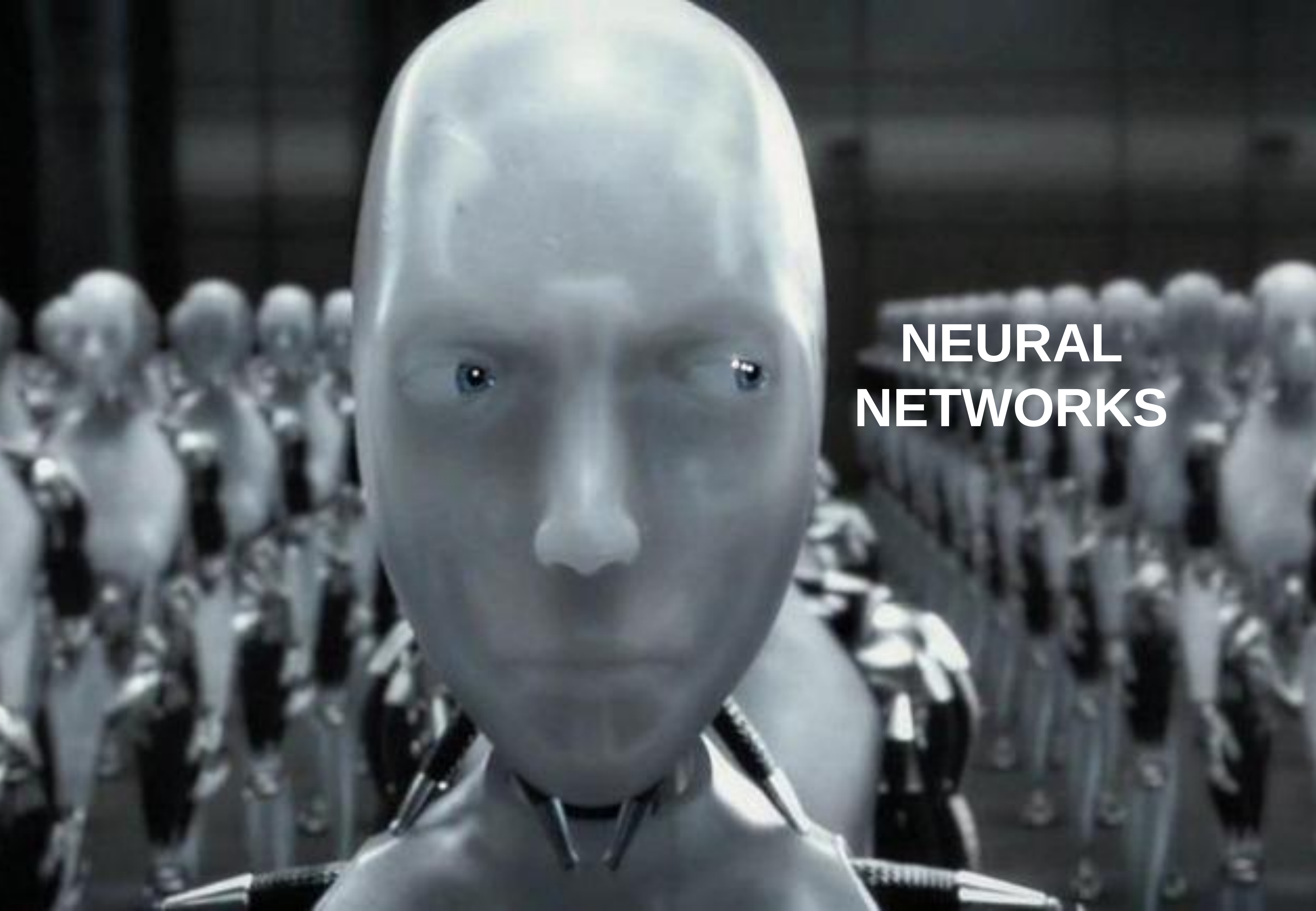
  (or find a best approximation)

- Supervised learning
  - Regression
  - Classification
  - Prediction
  - …



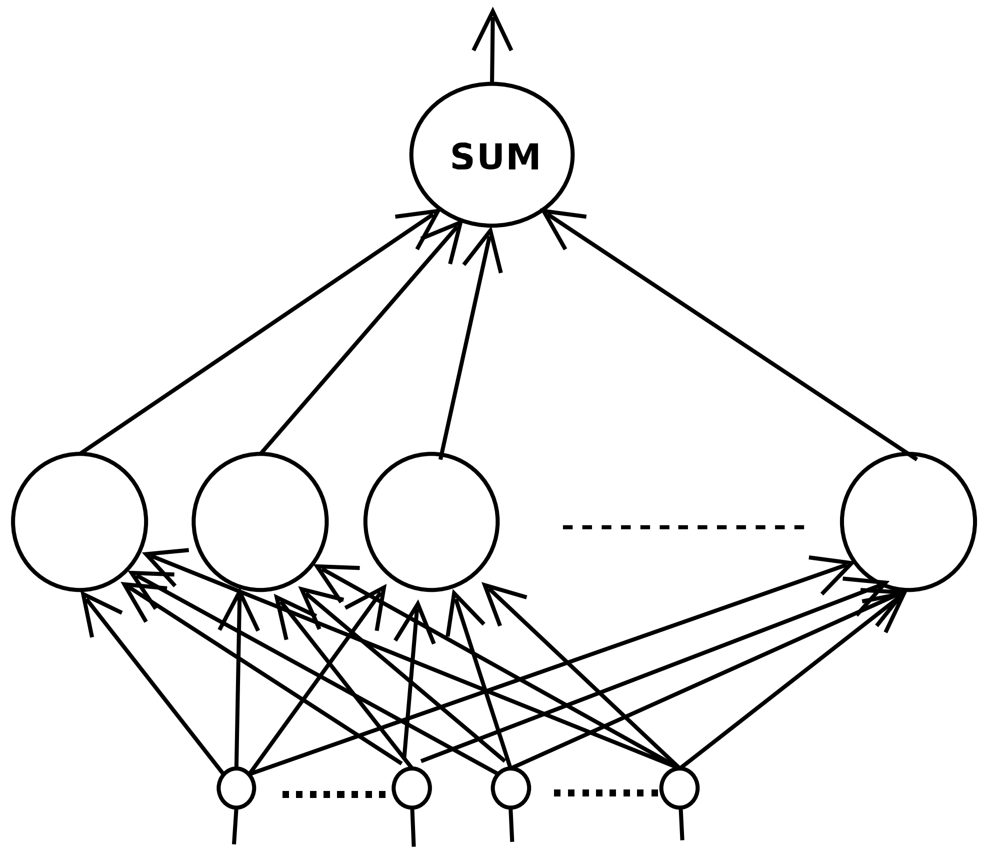
Regularization theory

- Empirical risk minimization:
  - Find a solution f that minimizes $H(f) = \Sigma\, (f(\mathbf{x}_i)-y_i)^2,\ (i=1 \ldots N)$
  - Generally ill-posed problem
  - Choose a solution according to a priori knowledge
    - (what should f look like? – e.g. smooth, small oscilations,)
- Regularization:
  - Add a stabilizer $A(f)$:
  - $H(f) = \Sigma\, (f(\mathbf{x}_i)-y_i)^2 + \gamma\, A(f),\ ,\ (i=1 \ldots N)$
    - $A(f)$ – based on Fourier transform divided by a kernel function
    - $A(f)$ – defined by a norm on reproducing kernel Hilbert space (RKHS)
  - $f(x) = \Sigma\, w_i\, K(\mathbf{x}-\mathbf{x}_i)$, for positive K, where $(\gamma\, \mathbf{I} - \mathbf{K})\, \mathbf{w} = \mathbf{y}$


NEURAL
NETWORKS

Regularization networks

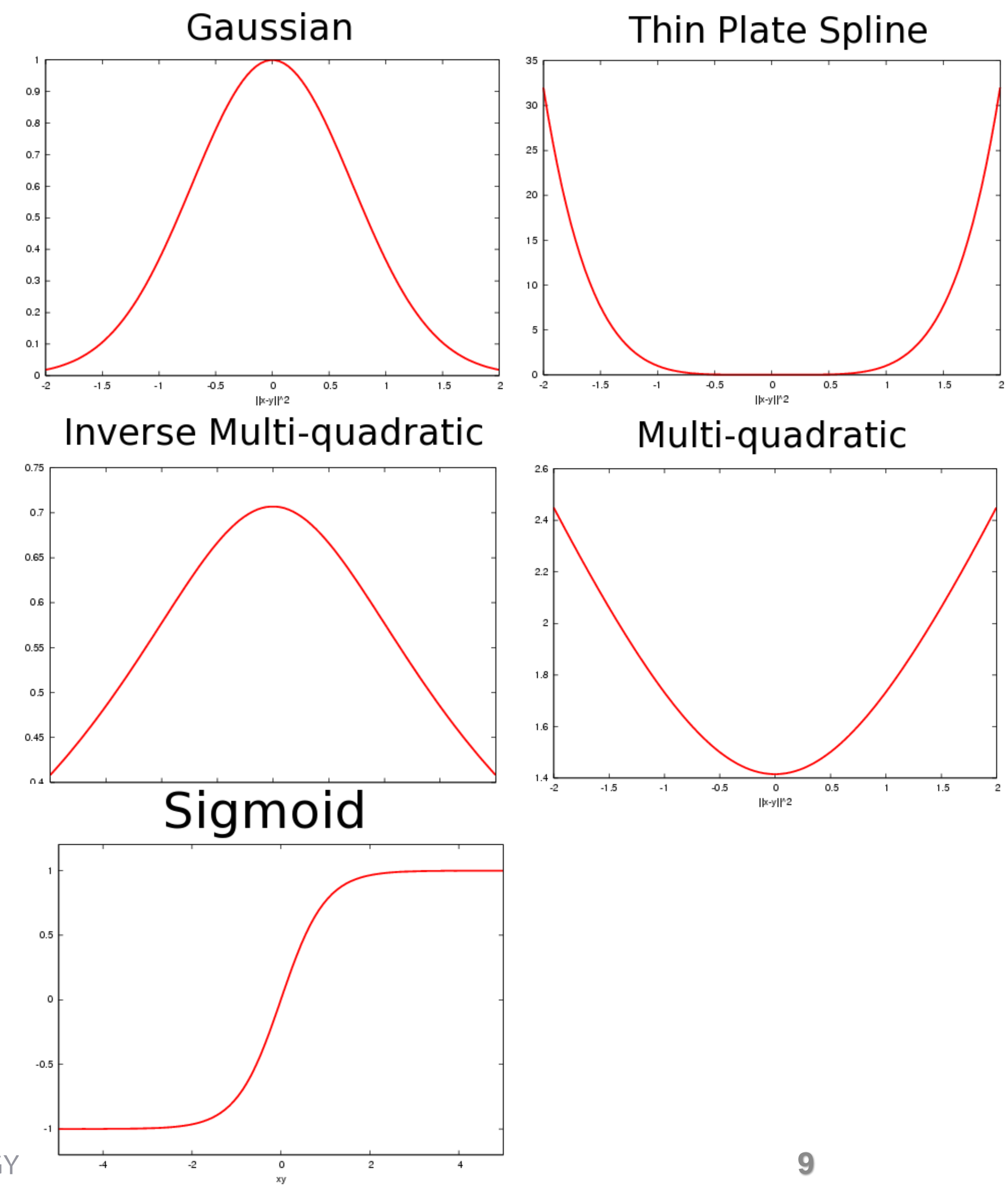
- f can be represented by a ***feed-forward network*** with one hidden layer of units computing K
- Function K is called ***basis*** or ***kernel*** function
- choice of K represents our knowledge or assumption about the problem
- choice of K is crucial for the generalization performance of the network



Regularization networks

- Basic Algorithm:
  - 1. set the centers of kernel functions to the data points
  - 2. compute the output weights by solving linear system

    $$(\gamma \boldsymbol{I} + \boldsymbol{K})\, \boldsymbol{w} = \boldsymbol{y}$$
- Pros:
  - easy, fast
- Cons:
  - choice of γ and K (and its parameters) is crucial for performance
  - … and it is data dependent: (no-free-lunch theorem)

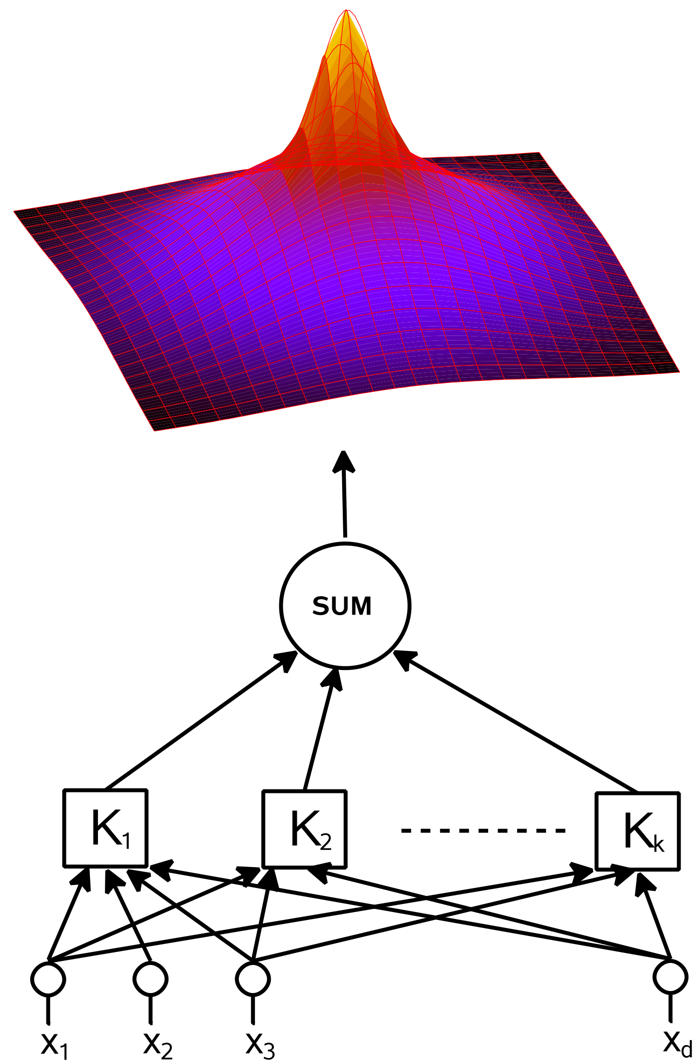


Sum (combination) kernels

- Aronszajn theory: RKHS is closed w.r.t. linear combination

$$K(x,y) = \alpha\, K_1(x,y) + \beta\, K_2(x,y)$$

- Each unit is composed of a two linearly combined kernels
- More parameters to set
- Possiblity to combine different kernel function
- Possibility to retain detailed approximation while having good generalization

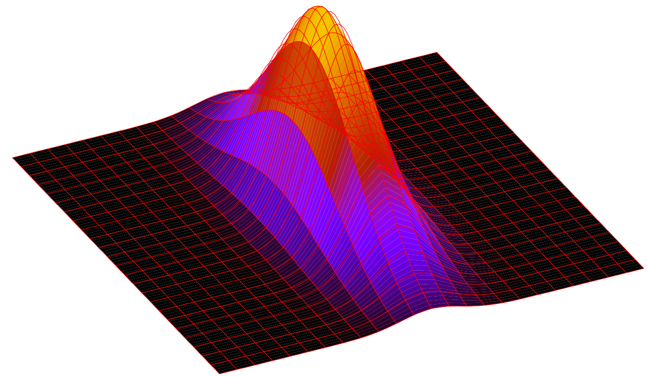


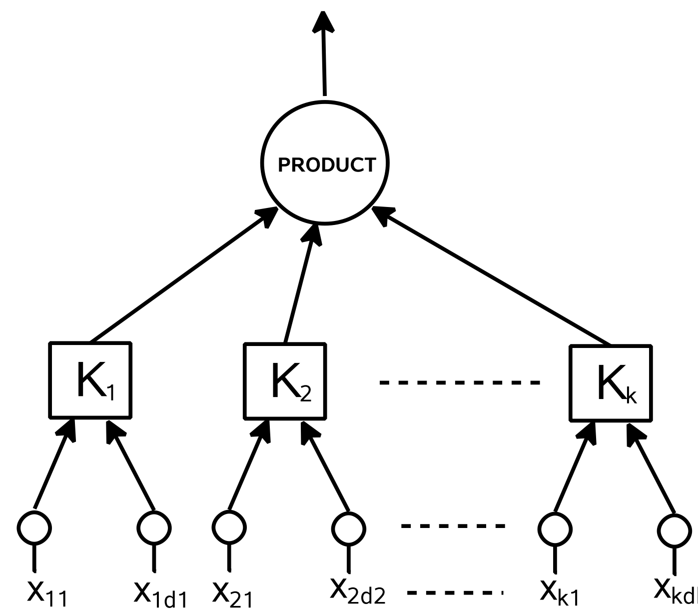
Product kernels

- Aronzsajn: product of two RKHS is in RKHS

$K(x_1x_2,y_1y_2) = K_1\,(x_1,y_1).K_2\,(x_2,y_2)$

- Each unit has two kernel functions operating on different subsets of inputs
- Heterogenous data:
  - Different properties of attributes
  - Processed by different kernels
- Even more parameters (input split)





Simple learning

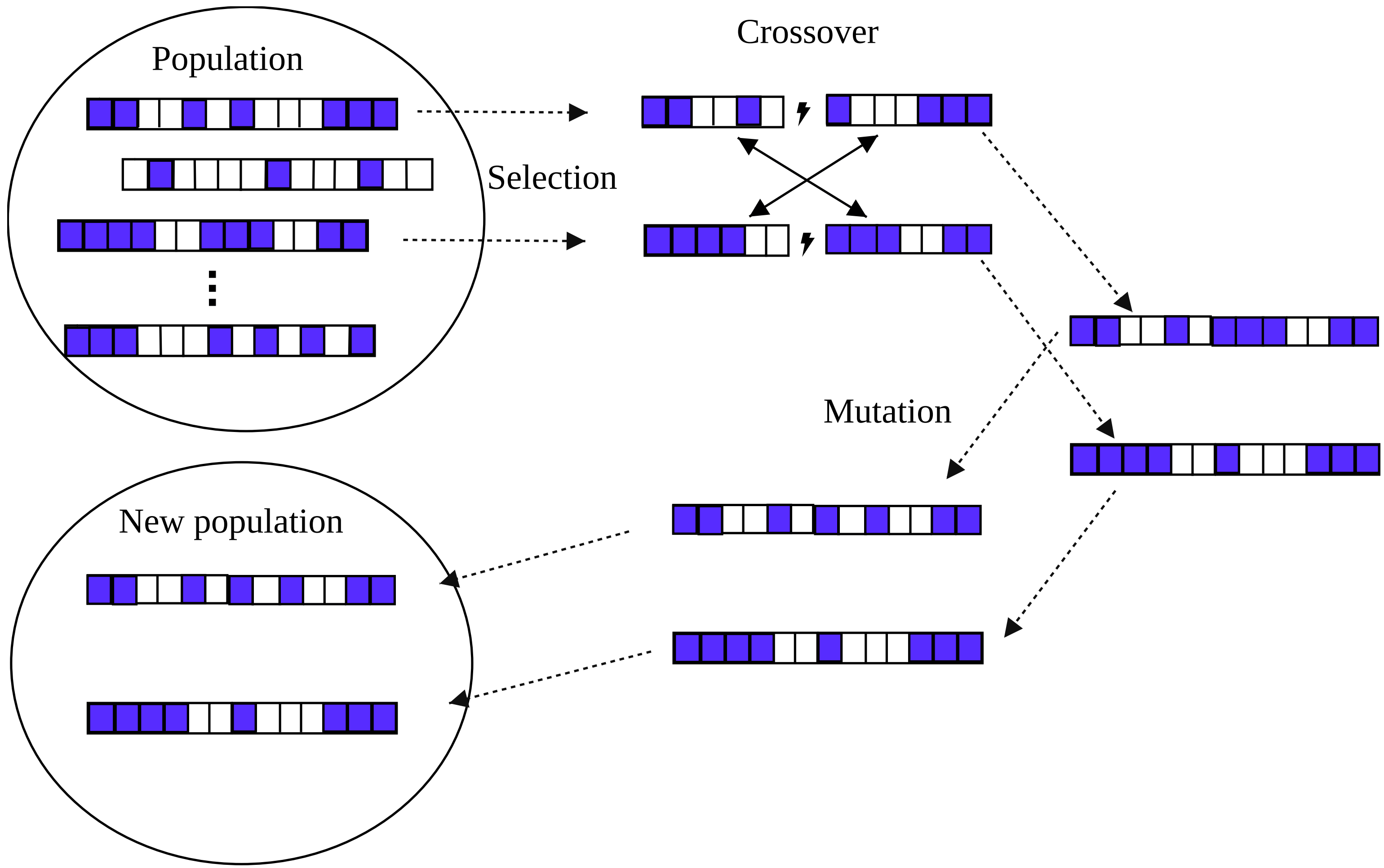
1. Expert knows/miracle (statistics) happens:
   - Type of kernel function, pairs for combinations and products
   - The regularization parameter γ
   - For combination kernels, the parameters of combination
   - For product kernels, the input split into two subsets
2. Then, solve a linear system

OR

1. Use tailored search algorithms to set the metaparameters
   - Such as evolutionary algorithm
2. Combine it with the linear part of the algorithm


EVOLUTIONARY
ALGORITHMS

Evolutionary learning



Evolutionary learning

- Population based search heuristics
- Prone to local optima
- Suitable for search of heterogeneous spaces/problems
- Does not require additional information such as gradients
- Encoding of metaparameters of kernel networks
  - Floating point parameters together with binary input splits and integer indices of kernel types
- Standard operations of arithmetic crossover for floats, one point crossover for discrete variables, and mutations
- Selection based on cross-validated performance of the fully-trained model

EXPERIMENTS

Data

- The dataset contain tens of thousands measurements of gas multi-sensor MOX array devices recording concentrations of several gas pollutants.
- Collocated with a conventional air pollution monitoring station that provides labels for the data.
- The data are recorded in 1 hour intervals.
- S. De Vito et al.

Table 1. Overview of data sets sizes.

Task	train set	test set	Task	train set	test set
sparse CO	1224	6120	CO i1-5	1469	5875
sparse NO2	1233	6160	NO2 i1-5	1479	5914
sparse NOx	1233	6163	NOx i1-5	1480	5916

Preliminary experiments - overview

Crossvalidation errors

Task	Gaussian kernel E_{avg}	Gaussian kernel stddev	Product kernels E_{avg}	Product kernels stddev	Sum kernels E_{avg}	Sum kernels stddev
CO	0.152	0.000	**0.148**	0.002	0.152	0.003
NO2	0.429	0.003	**0.407**	0.009	0.434	0.012
NOx	0.227	0.000	**0.207**	0.006	0.229	0.005

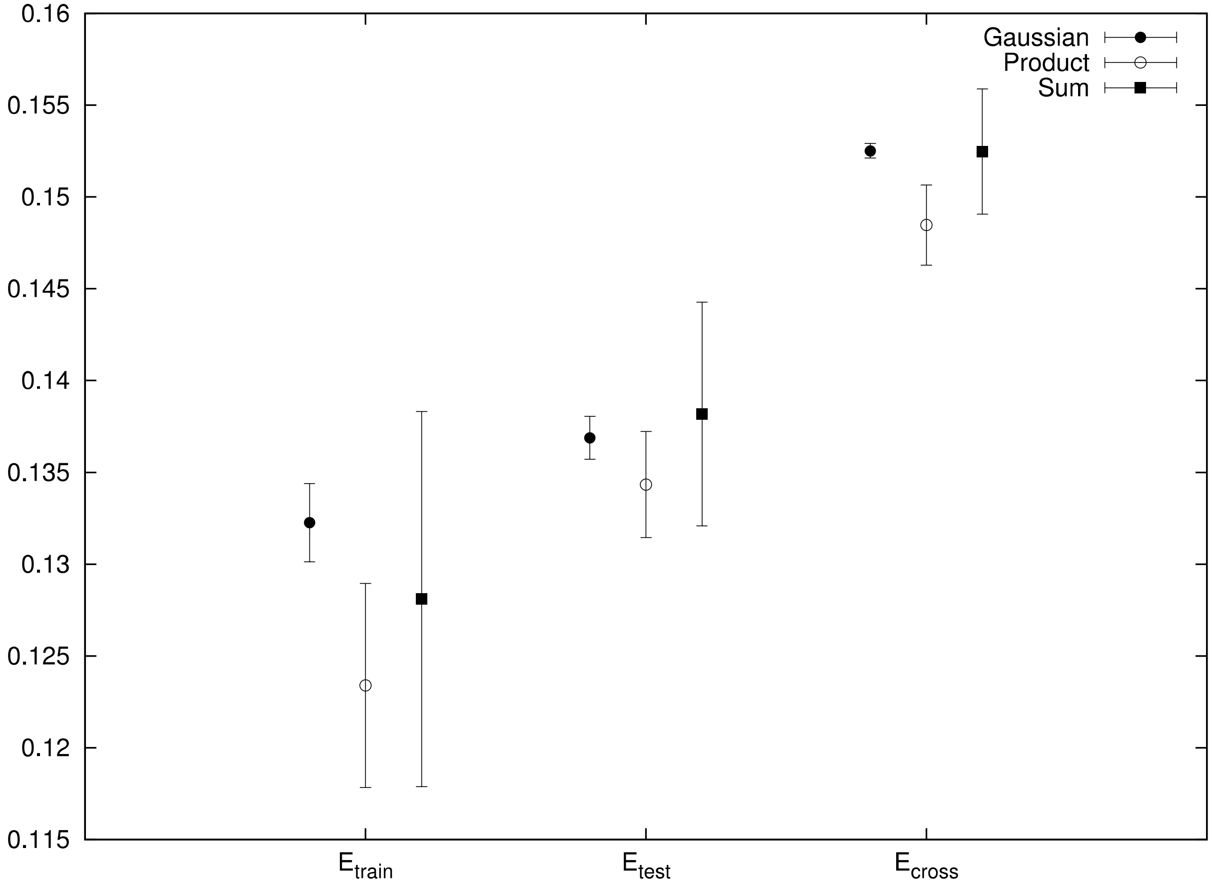
Training errors

Task	Gaussian kernel E_{avg}	Gaussian kernel stddev	Product kernels E_{avg}	Product kernels stddev	Sum kernels E_{avg}	Sum kernels stddev
CO	0.132	0.002	**0.123**	0.005	0.128	0.010
NO2	0.308	0.002	**0.277**	0.025	0.312	0.003
NOx	0.139	0.001	**0.135**	0.011	0.139	0.002

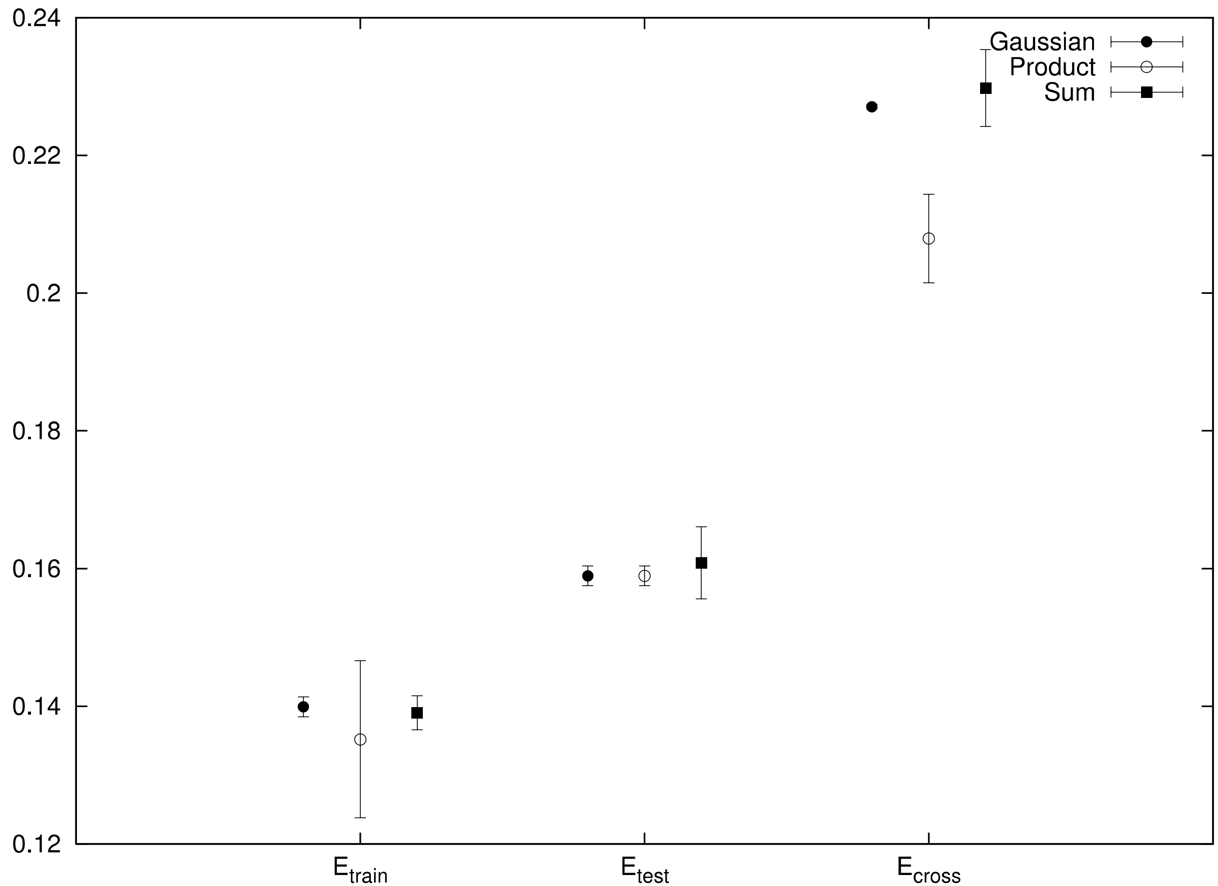
Testing errors

Task	Gaussian kernel E_{avg}	Gaussian kernel stddev	Product kernels E_{avg}	Product kernels stddev	Sum kernels E_{avg}	Sum kernels stddev
CO	0.136	0.001	**0.134**	0.002	0.138	0.006
NO2	**0.334**	0.002	0.343	0.011	0.338	0.004
NOx	**0.158**	0.001	**0.158**	0.008	0.160	0.005

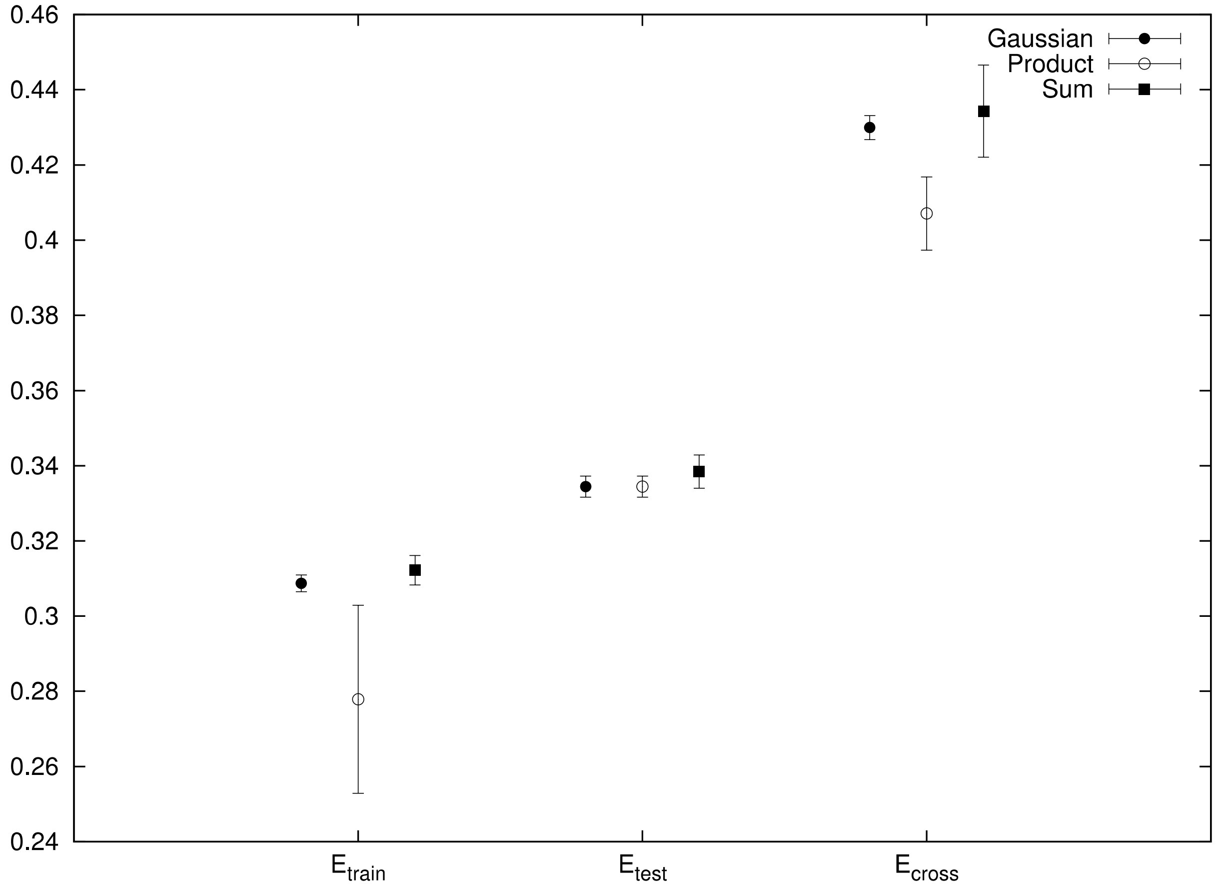
Preliminary experiments - CO



Preliminary experiments – NO2



Preliminary experiments - NOx



Experiment 2 – Training errors

Training errors

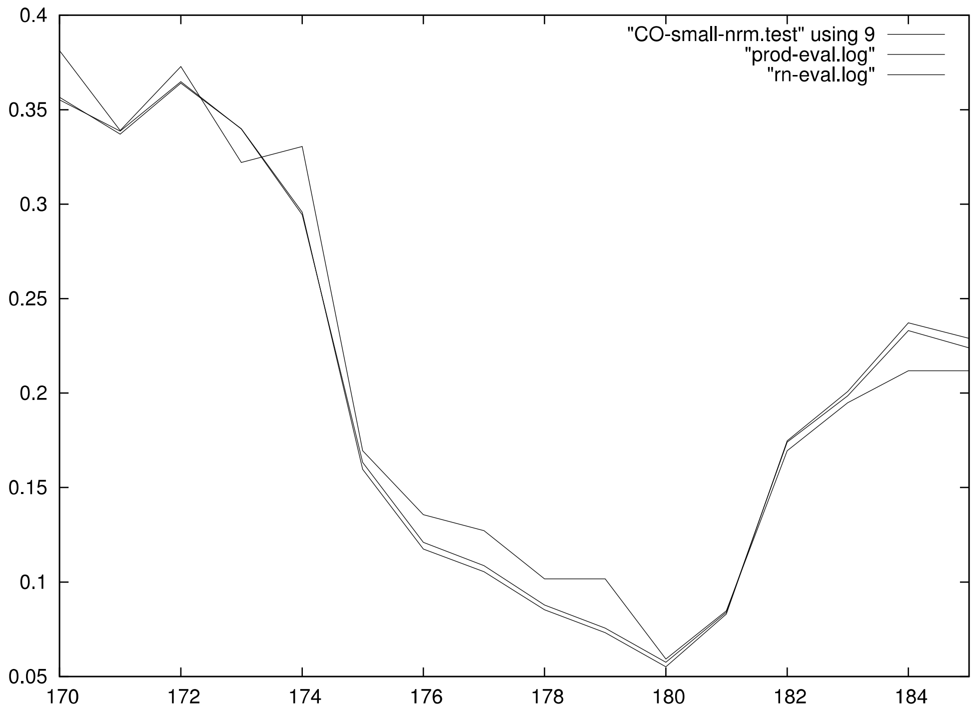
Task	Gaussian kernel E_{avg}	stddev	min	max	Product kernels E_{avg}	stddev	min	max
CO-i1	**0.050**	0.000	0.050	0.050	0.051	0.002	**0.049**	0.055
CO-i2	0.049	0.000	0.049	0.049	**0.046**	0.002	**0.043**	0.050
CO-i3	**0.054**	0.000	**0.053**	0.054	0.056	0.003	0.054	0.065
CO-i4	**0.333**	0.001	0.332	0.334	0.347	0.016	**0.325**	0.378
CO-i5	0.133	0.000	0.132	0.133	**0.097**	0.018	**0.077**	0.142
NO2-i1	**0.096**	0.002	0.093	0.101	0.100	0.015	**0.091**	0.141
NO2-i2	0.133	0.001	0.131	0.134	**0.122**	0.014	**0.105**	0.148
NO2-i3	0.388	0.001	0.384	0.389	**0.314**	0.077	**0.214**	0.434
NO2-i4	0.297	0.002	0.295	0.299	**0.287**	0.012	**0.265**	0.307
NO2-i5	**0.375**	0.001	0.374	0.376	0.389	0.032	**0.330**	0.435
NOx-i1	0.018	0.000	0.018	0.018	**0.017**	0.001	**0.016**	0.020
NOx-i2	0.026	0.000	0.026	0.027	**0.025**	0.002	**0.021**	0.028
NOx-i3	0.156	0.001	0.154	0.158	**0.152**	0.019	**0.121**	0.184
NOx-i4	0.231	0.002	0.229	0.234	**0.230**	0.017	**0.203**	0.258
NOx-i5	0.106	0.023	0.087	0.132	**0.095**	0.011	**0.083**	0.122

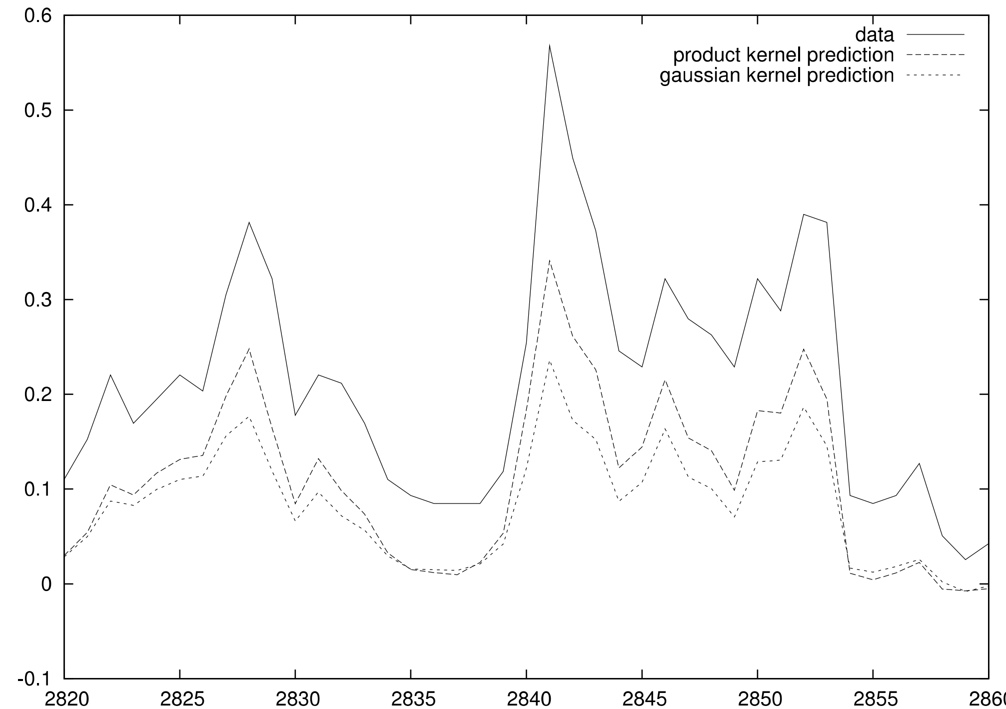
Experiment 2 – Testing errors

Testing errors

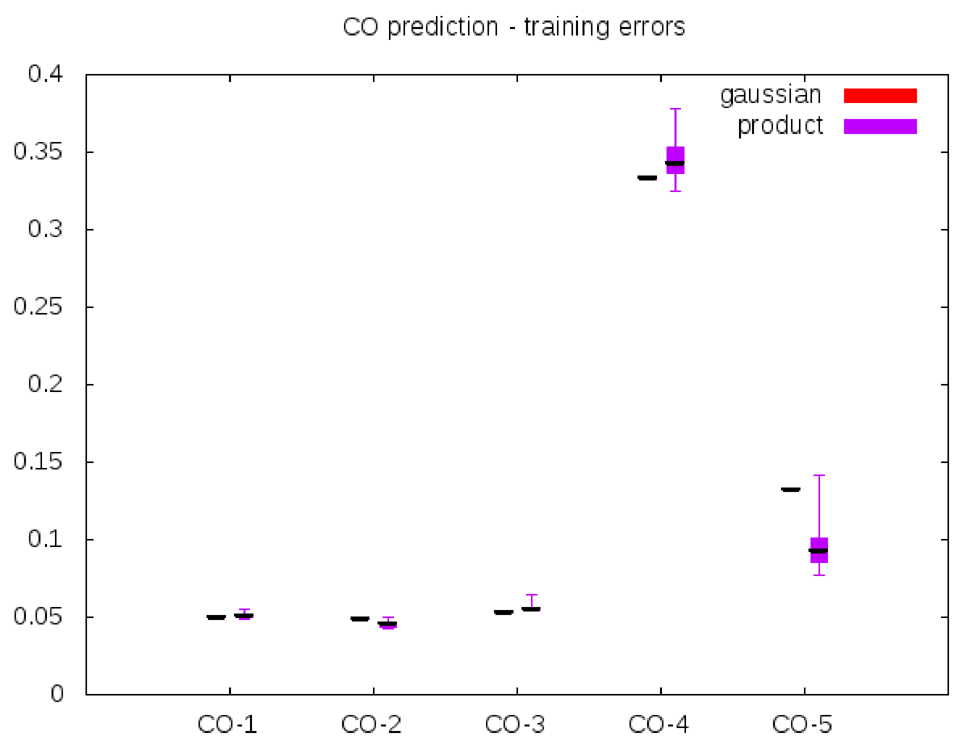
	Gaussian kernel				Product kernels			
Task	E_{avg}	stddev	min	max	E_{avg}	stddev	min	max
CO-i1	**0.210**	0.005	0.205	0.217	0.214	0.020	**0.192**	0.248
CO-i2	1.134	0.007	1.116	1.142	**0.878**	0.088	**0.709**	0.988
CO-i3	0.233	0.009	0.221	0.254	**0.228**	0.019	**0.197**	0.267
CO-i4	**0.326**	0.002	**0.323**	0.329	0.749	0.512	0.433	1.921
CO-i5	**0.296**	0.005	0.287	0.301	0.321	0.050	**0.204**	0.374
NO2-i1	**2.151**	0.052	2.096	2.267	2.263	0.540	**1.189**	2.997
NO2-i2	5.260	0.045	5.161	5.319	**3.928**	1.447	**2.661**	6.874
NO2-i3	**0.718**	0.004	**0.709**	0.721	1.033	0.218	0.764	1.351
NO2-i4	0.735	0.011	0.726	0.757	**0.734**	0.069	**0.669**	0.908
NO2-i5	**0.678**	0.024	**0.655**	0.735	0.913	0.183	0.709	1.302
NOx-i1	2.515	0.015	2.495	2.538	**2.409**	0.159	**2.093**	2.658
NOx-i2	3.113	0.019	3.081	3.139	**2.495**	0.068	**2.416**	2.592
NOx-i3	1.105	0.008	1.088	1.114	**0.956**	0.267	**0.730**	1.689
NOx-i4	**0.952**	0.008	0.941	0.970	1.256	0.520	**0.774**	2.610
NOx-i5	**0.730**	0.102	0.642	0.850	0.748	0.091	**0.544**	0.856

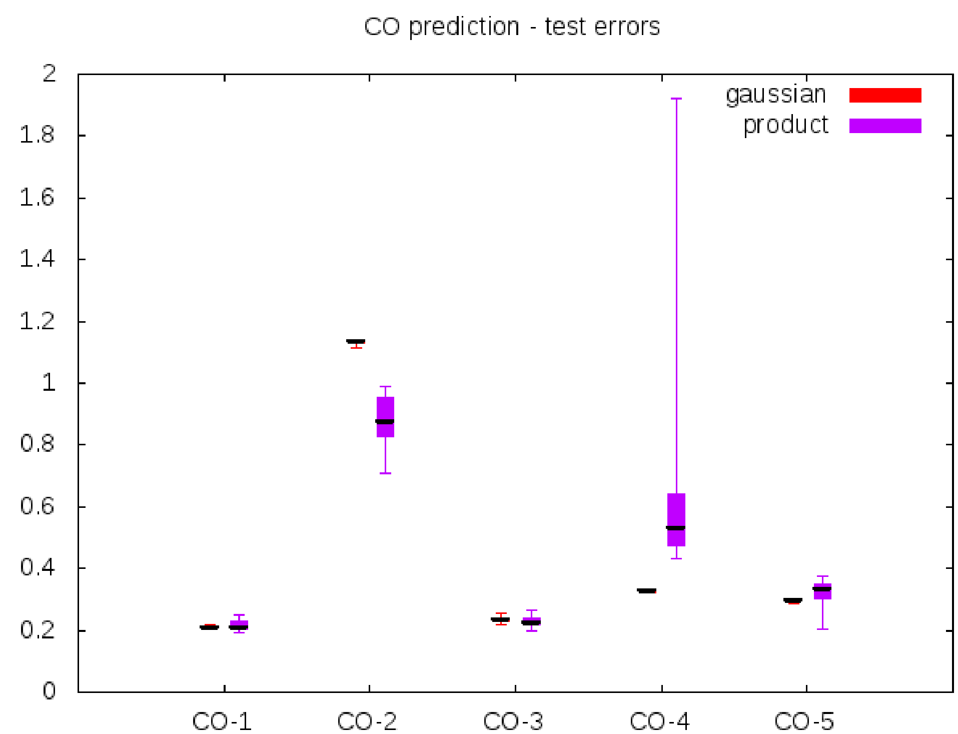
Experiment 2 - example



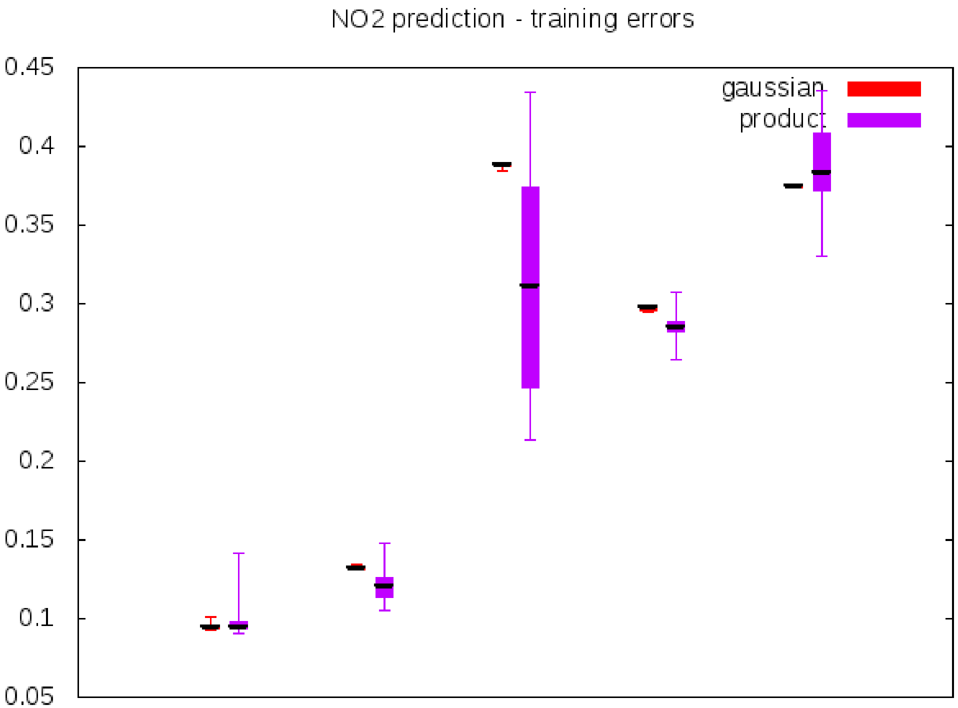


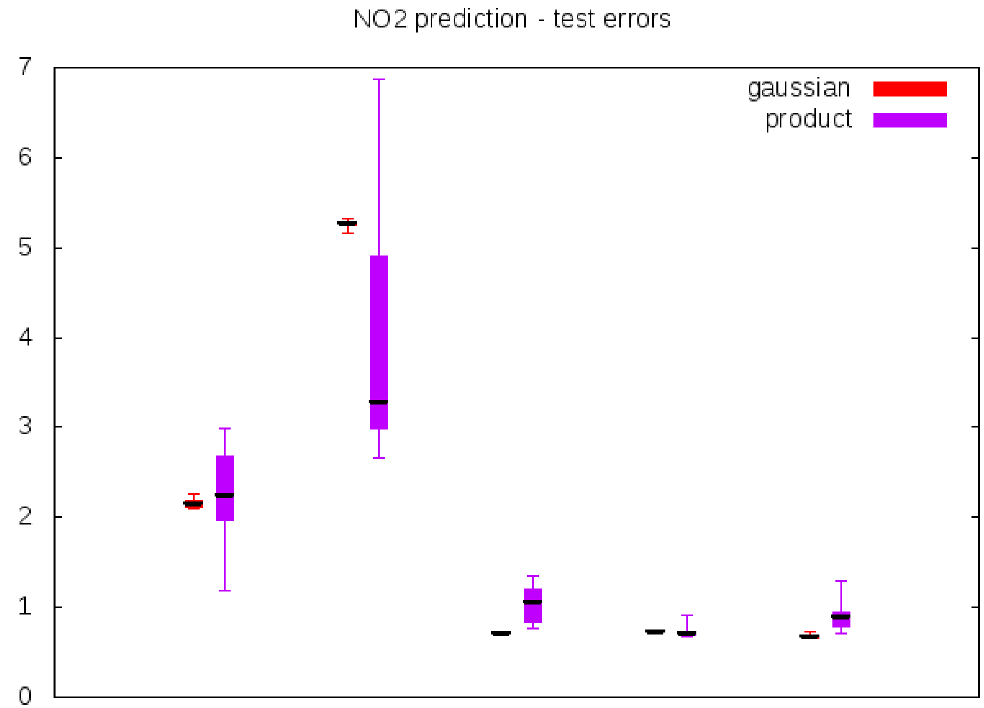
Experiment 2 - CO



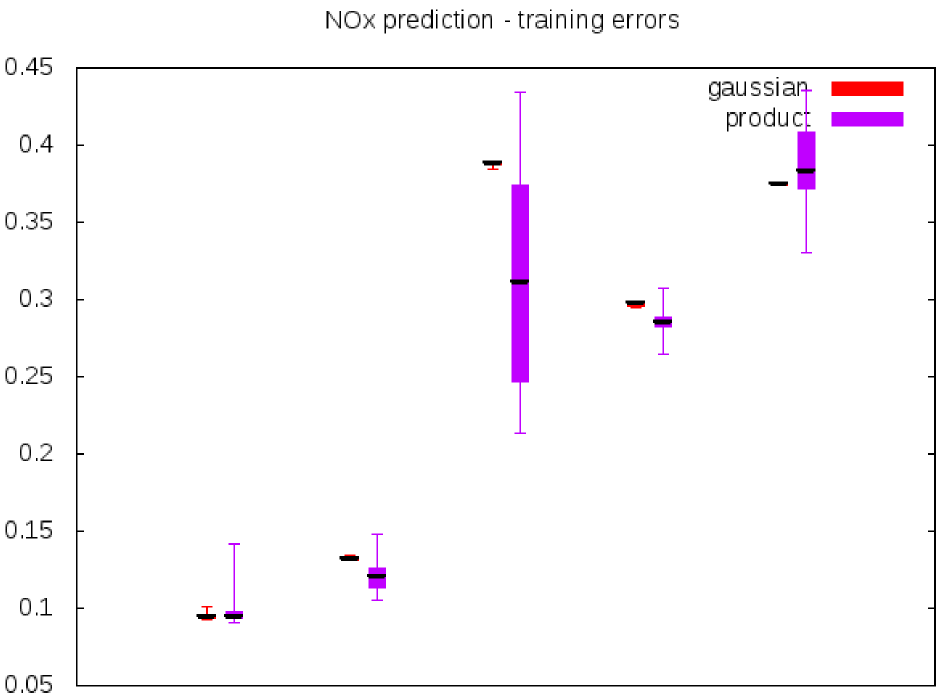


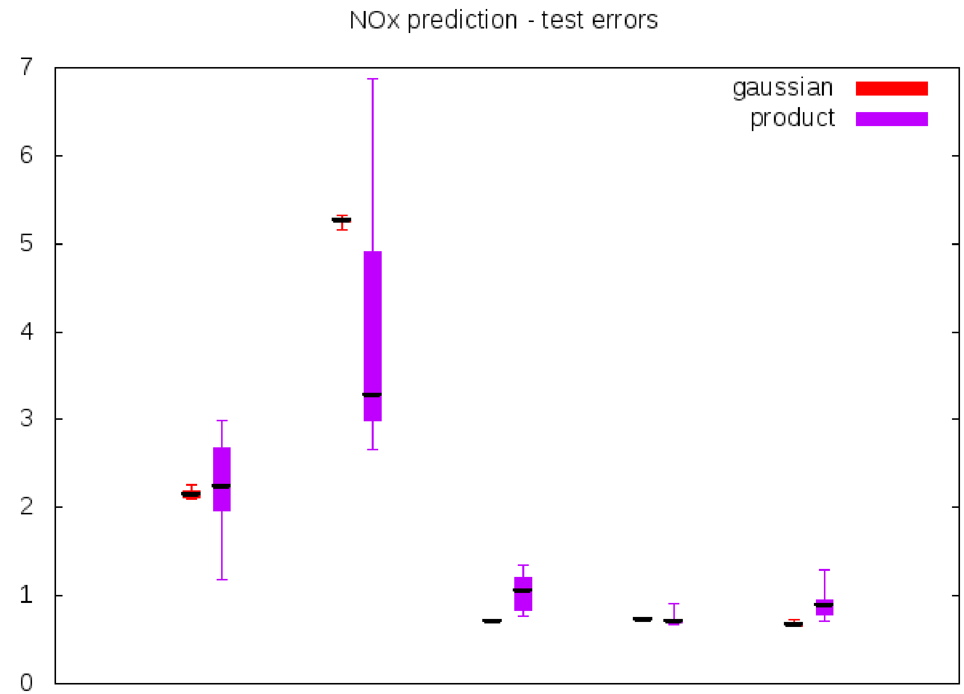
Experiment 2 – NO2



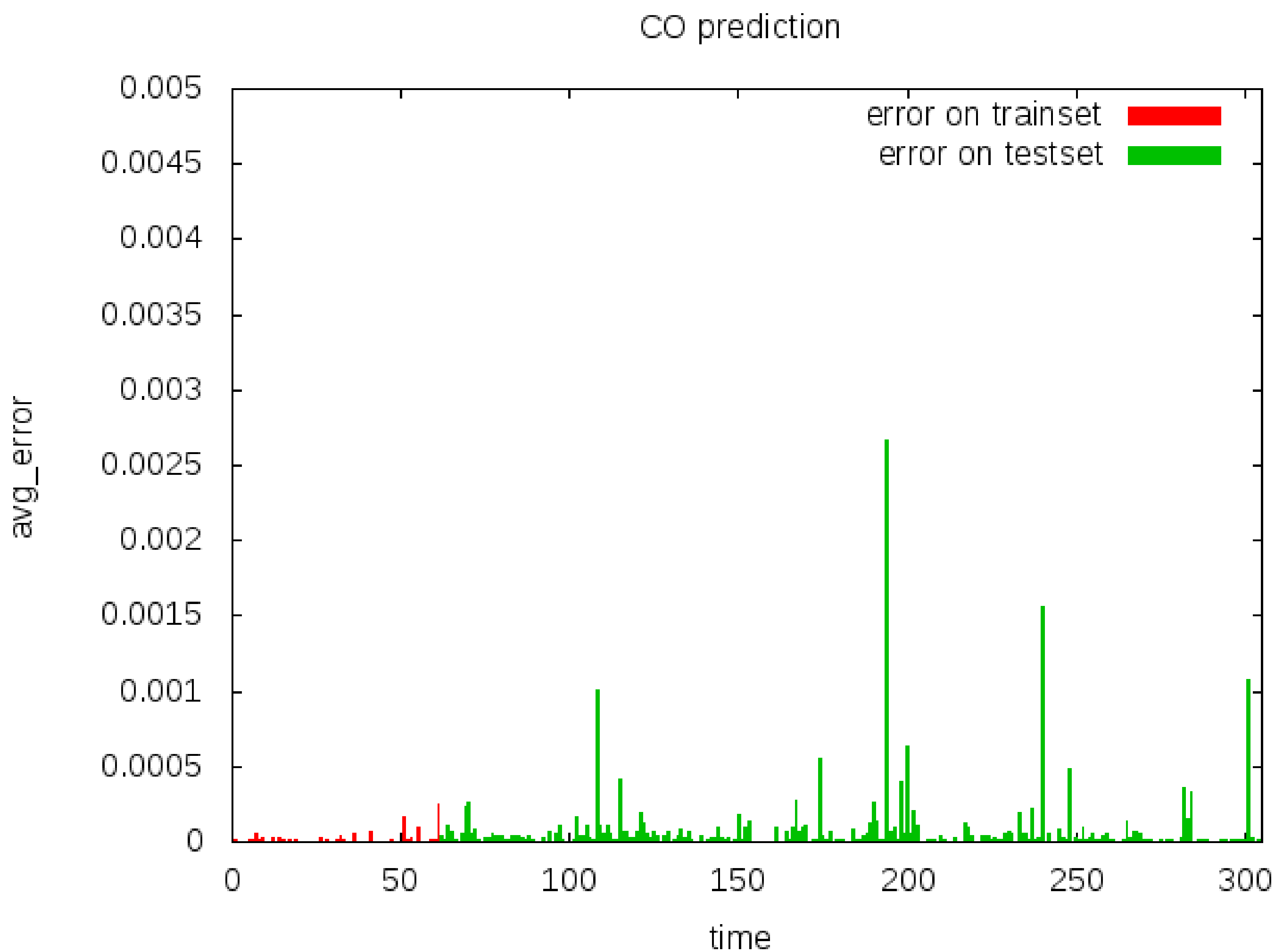


Experiment 2 - NOx

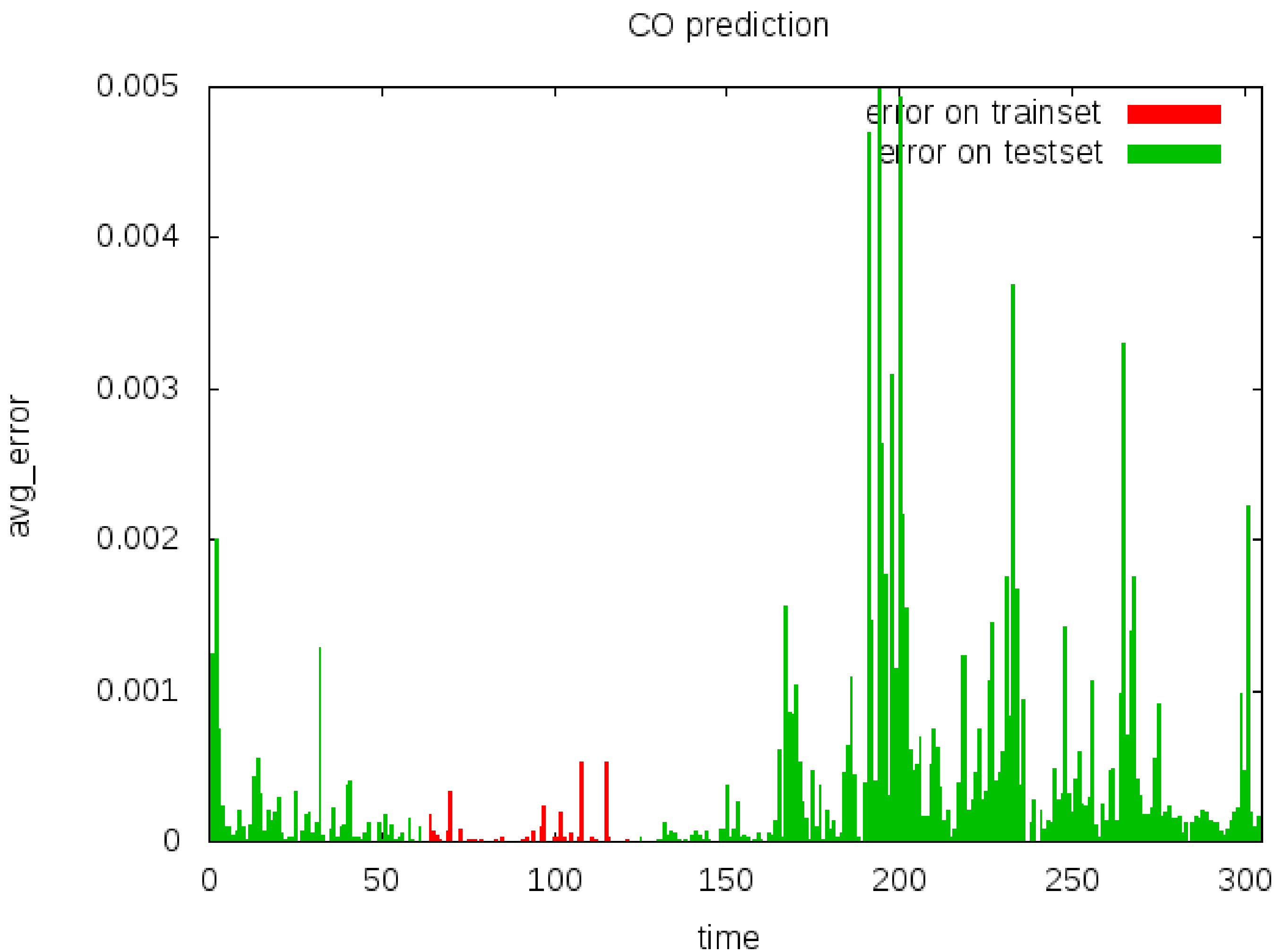




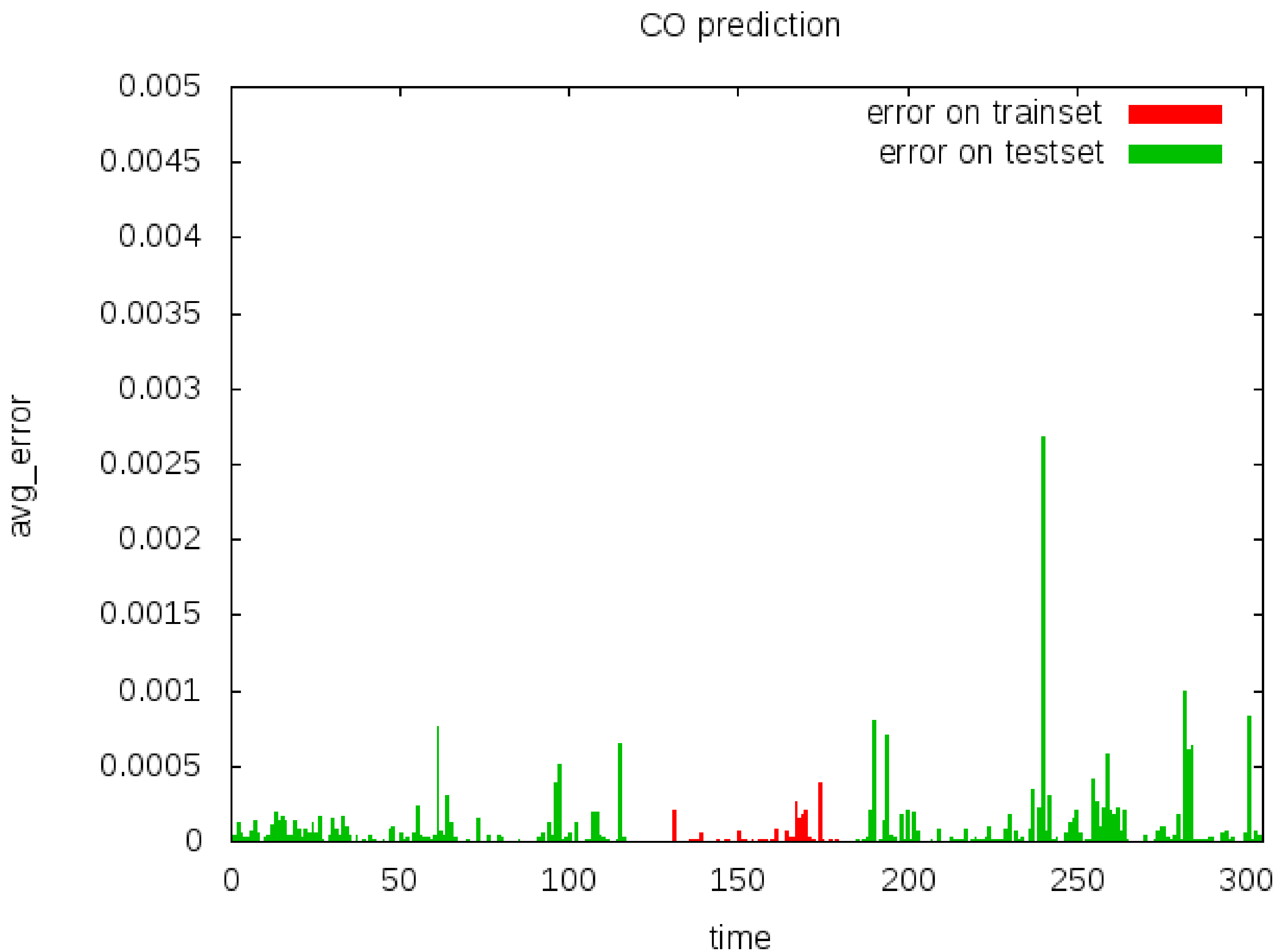
Experiment 2 - CO



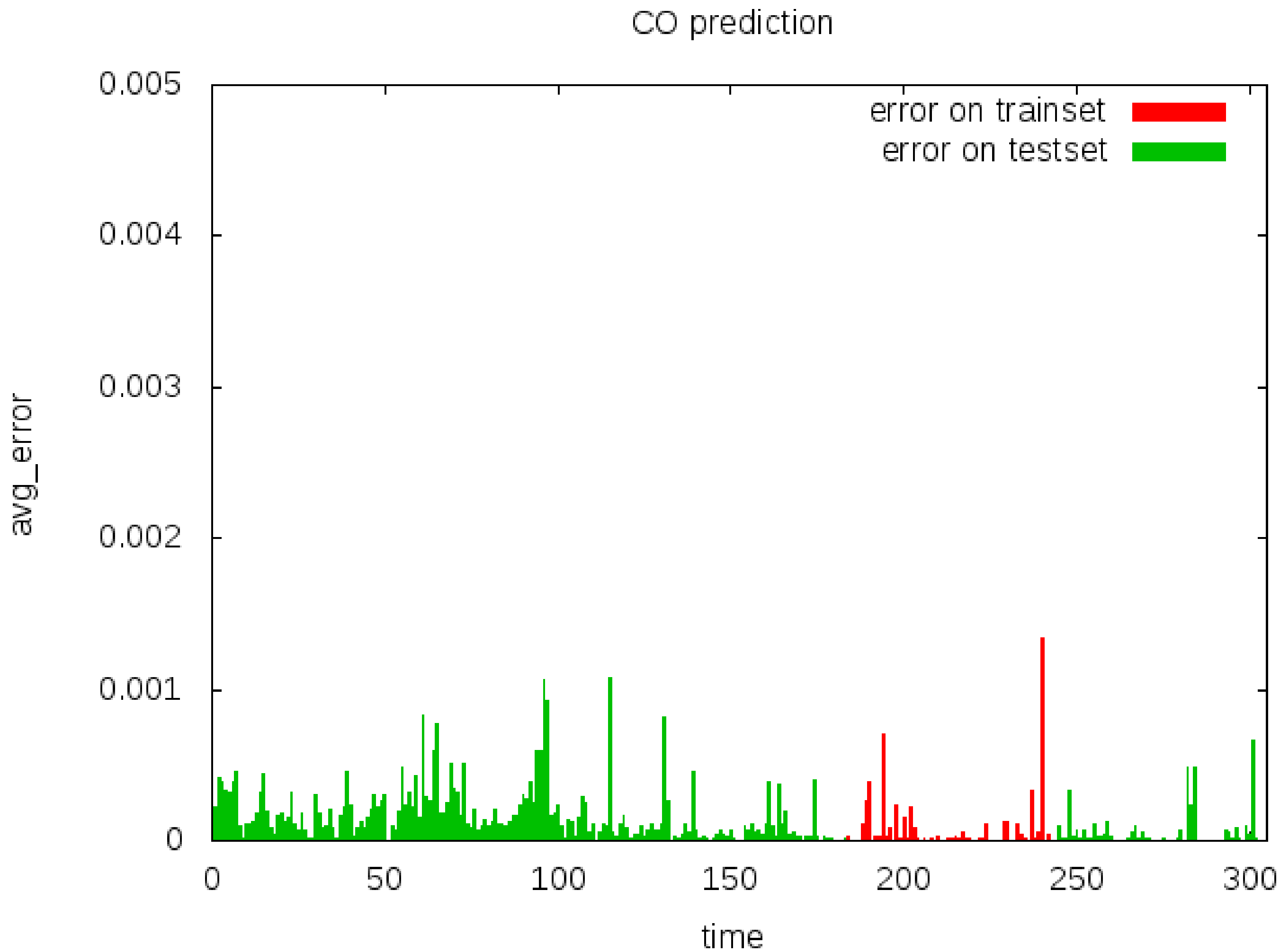
Experiment 2 - CO



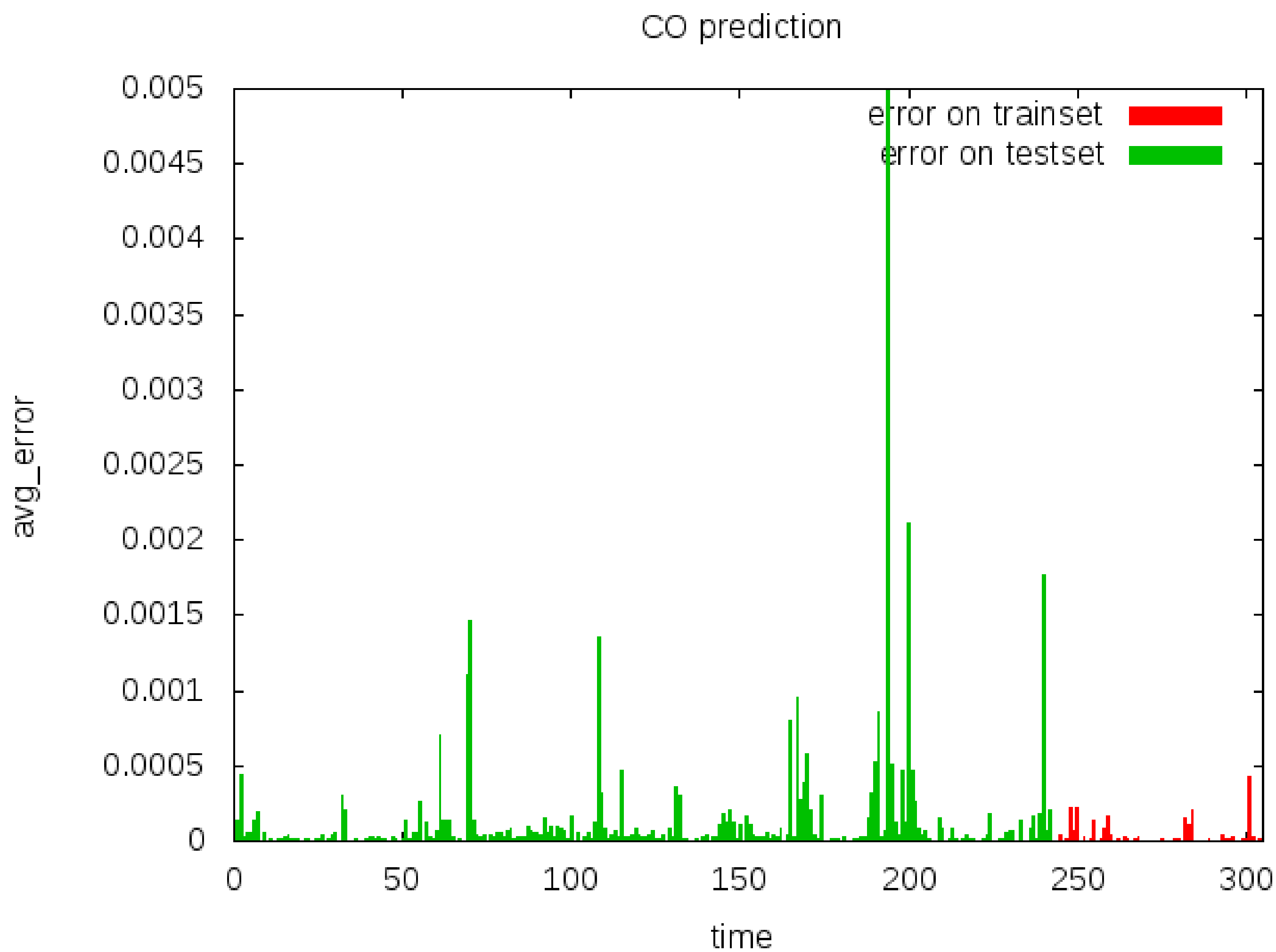
Experiment 2 - CO



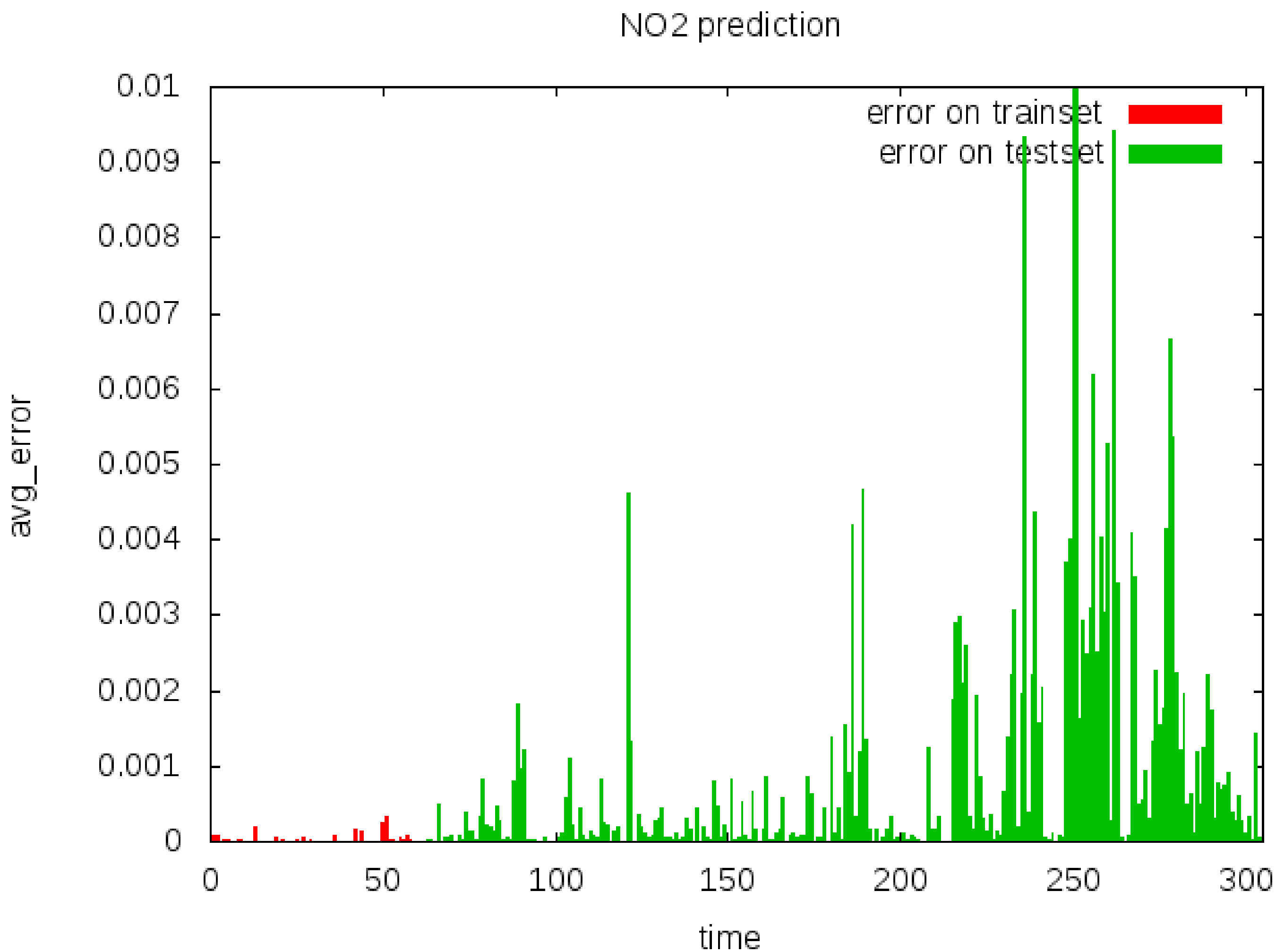
Experiment 2 - CO



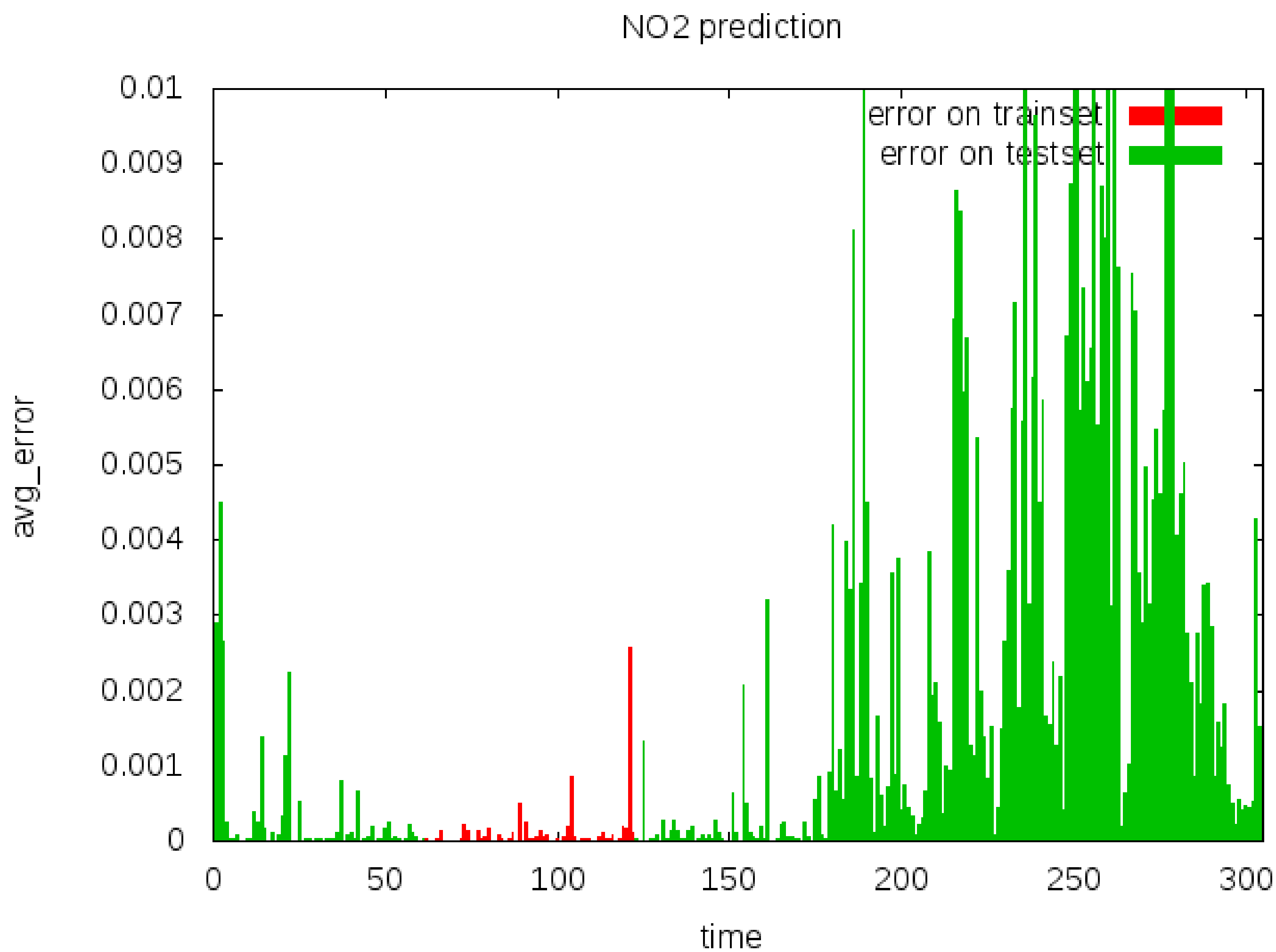
Experiment 2 - CO



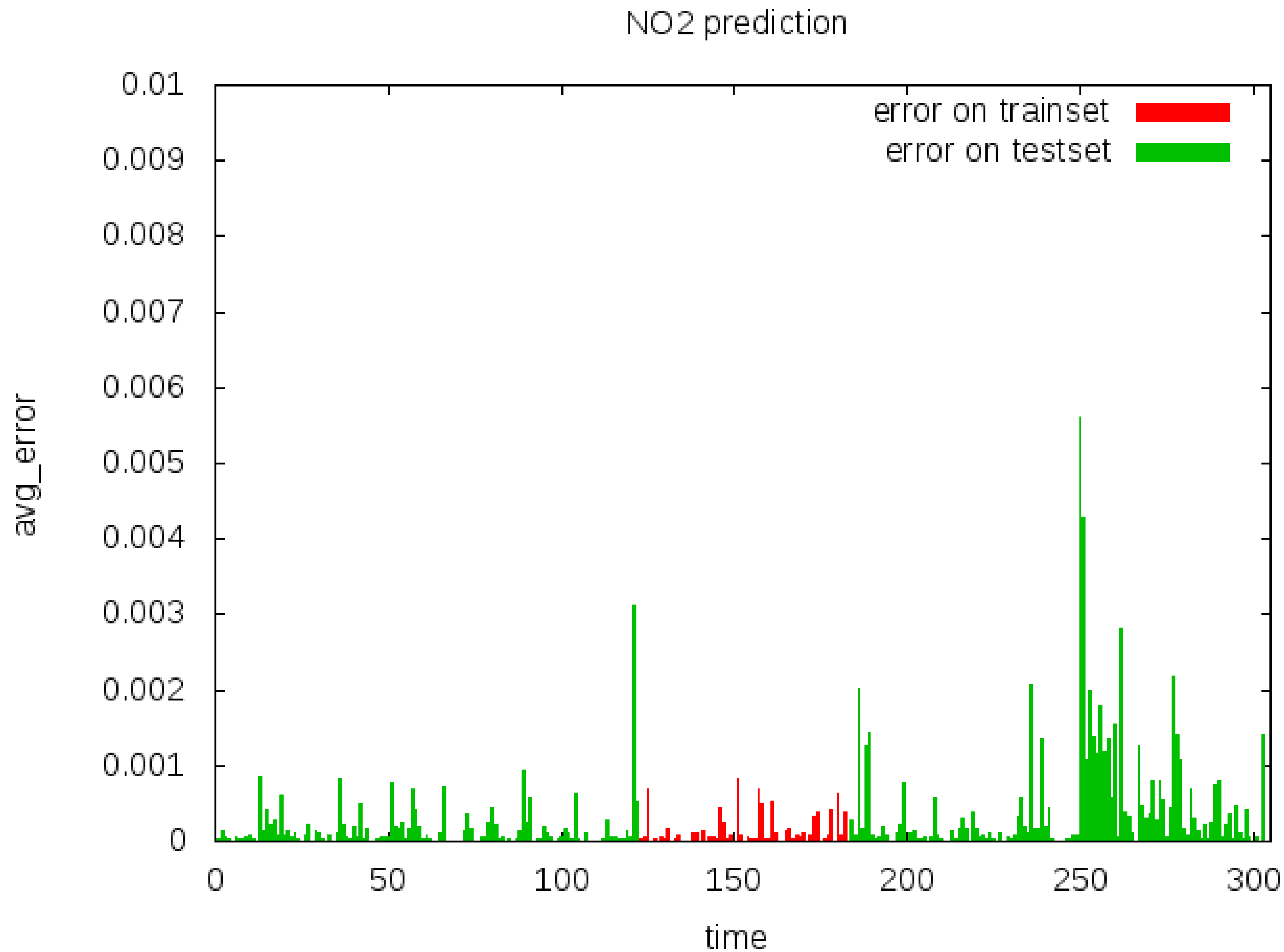
Experiment 2 – NO2



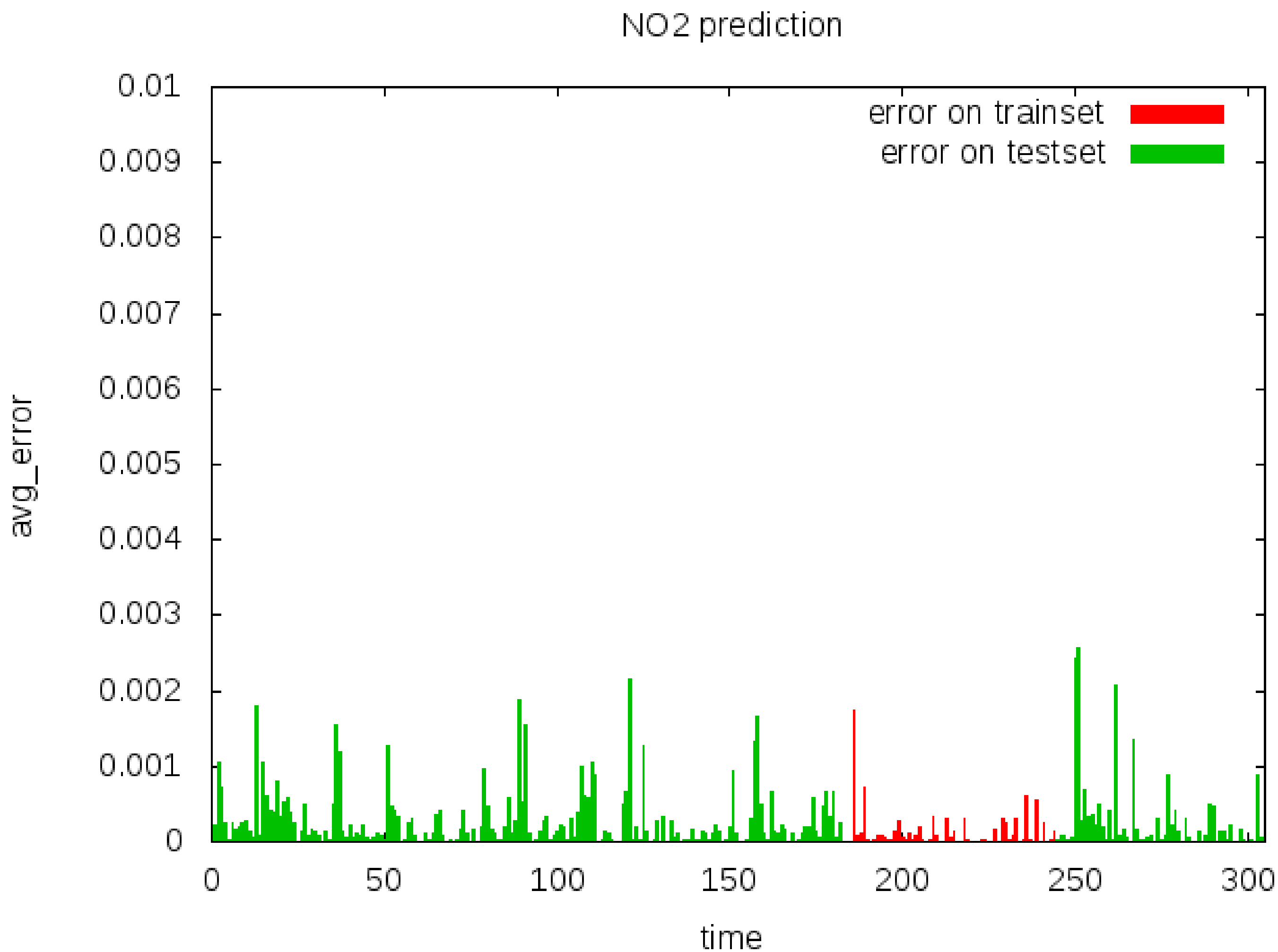
Experiment 2 – NO2



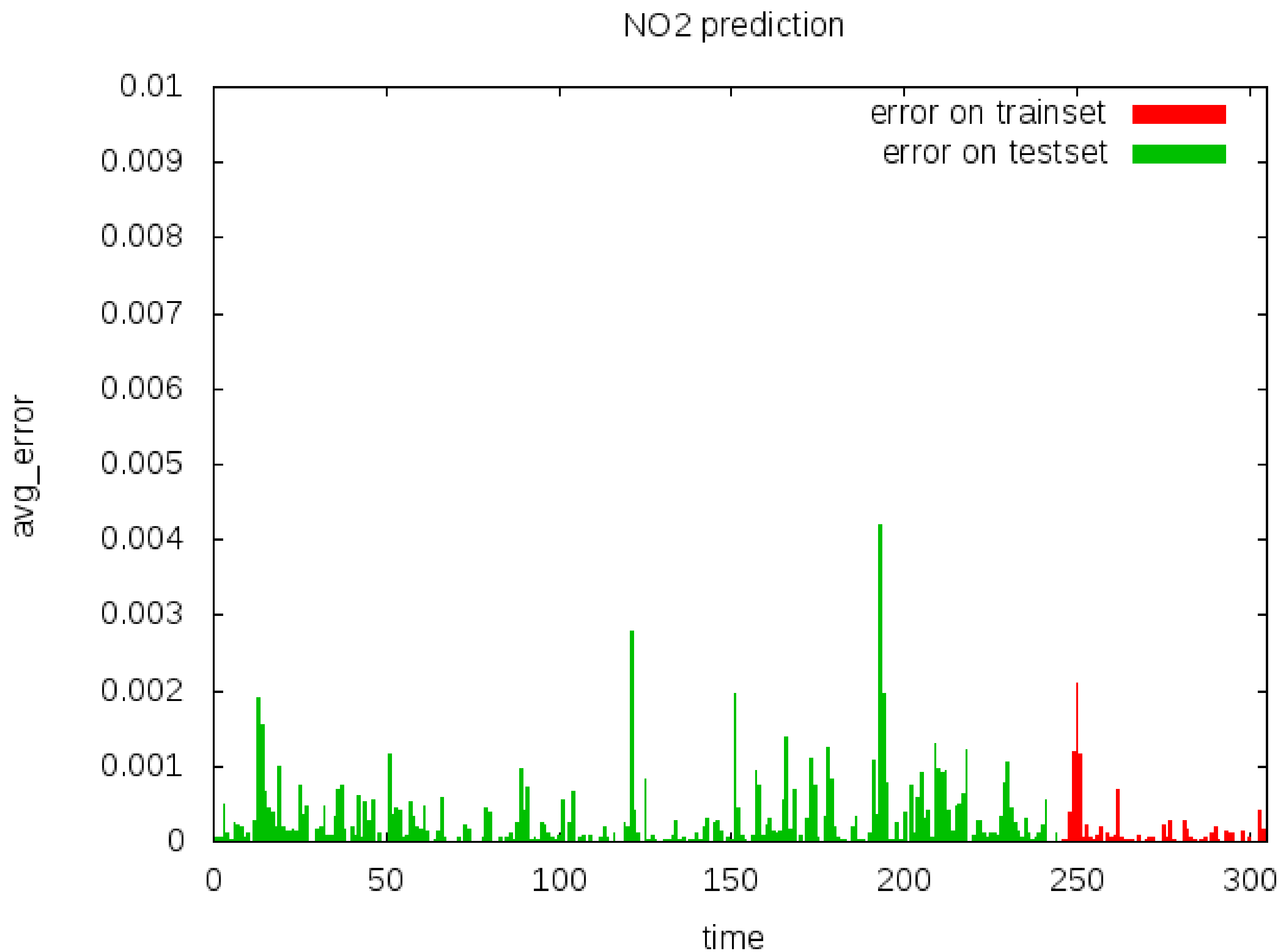
Experiment 2 – NO2



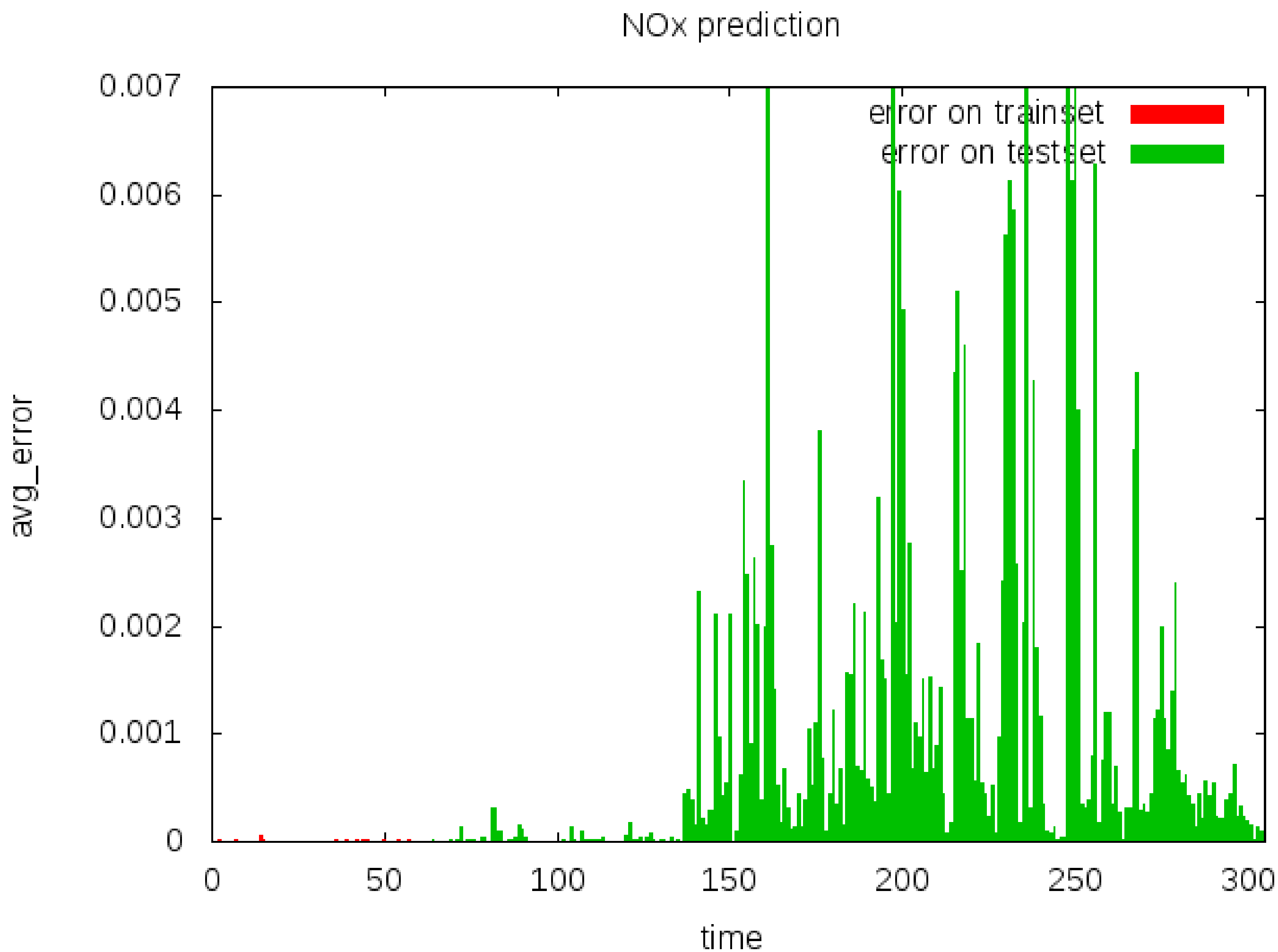
Experiment 2 – NO2



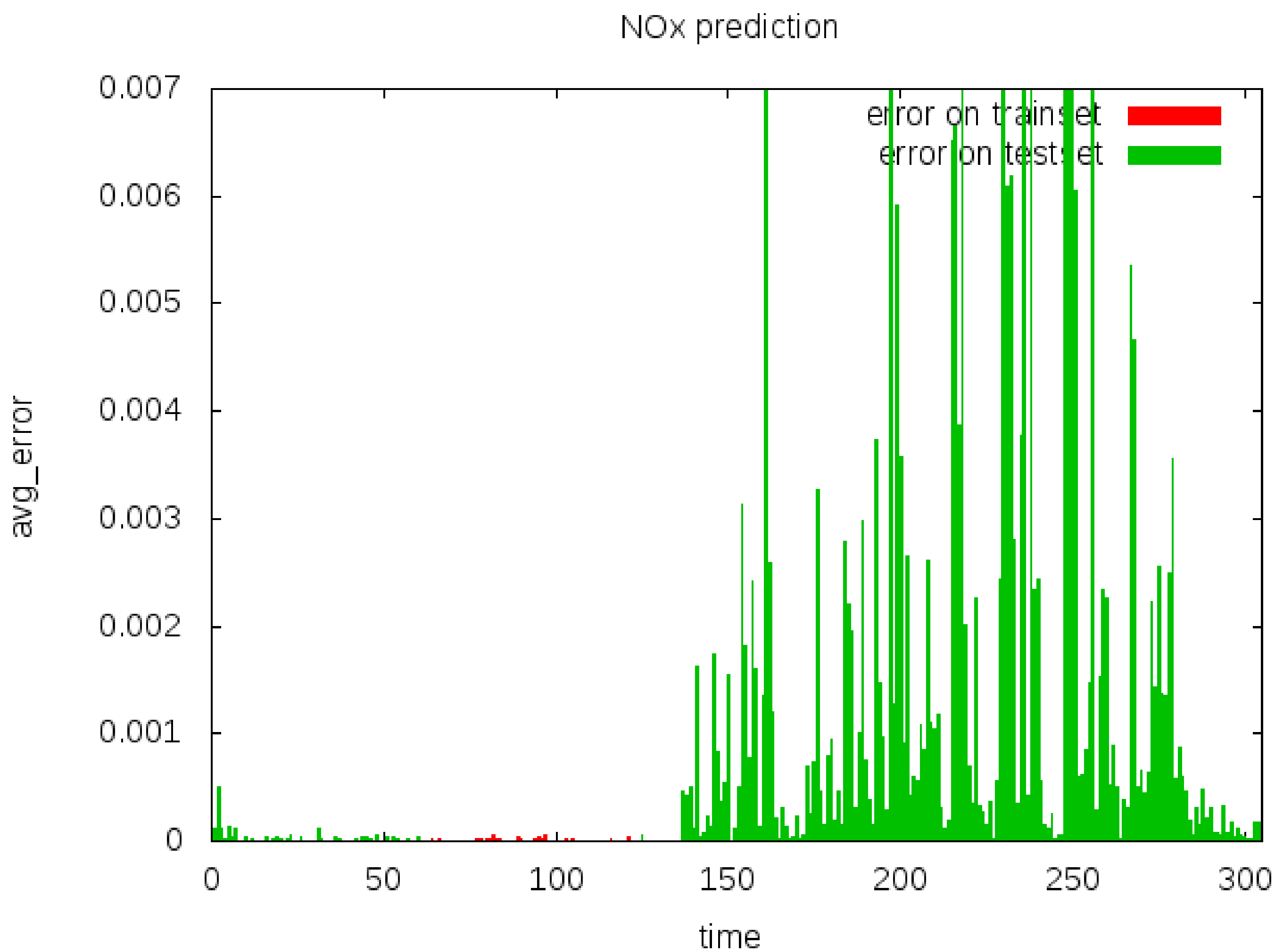
Experiment 2 – NO2



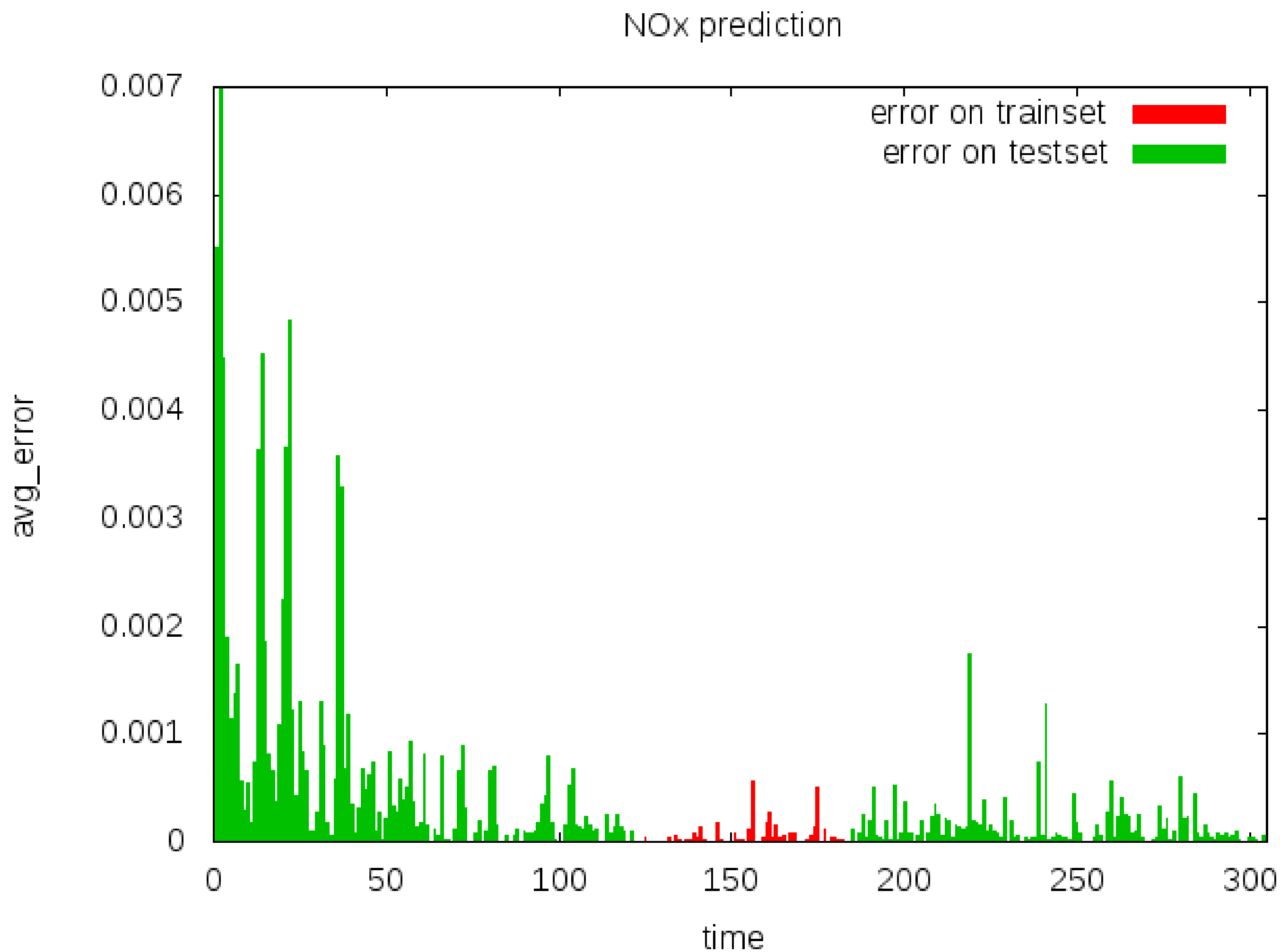
Experiment 2 – NOx



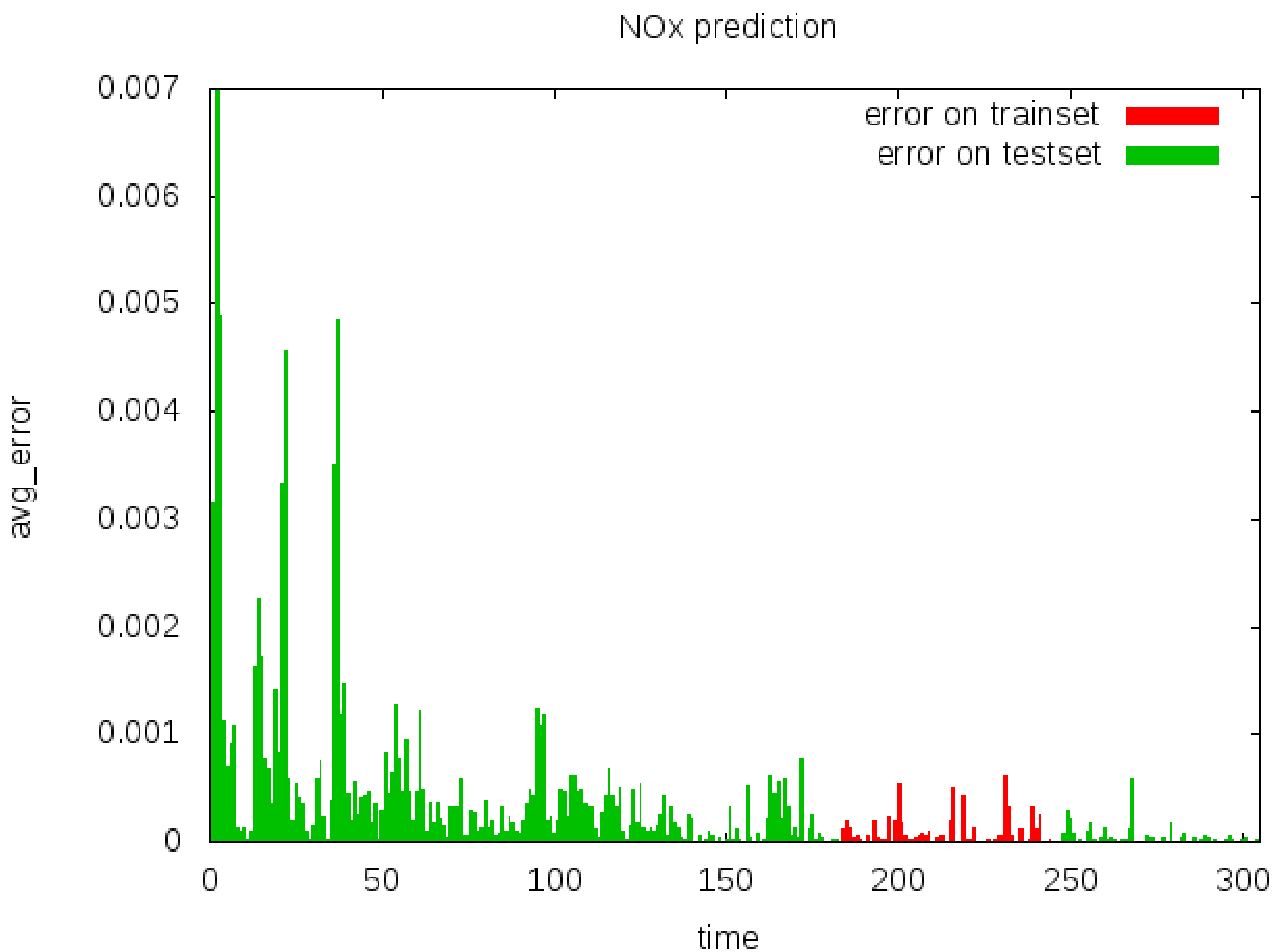
Experiment 2 – NOx



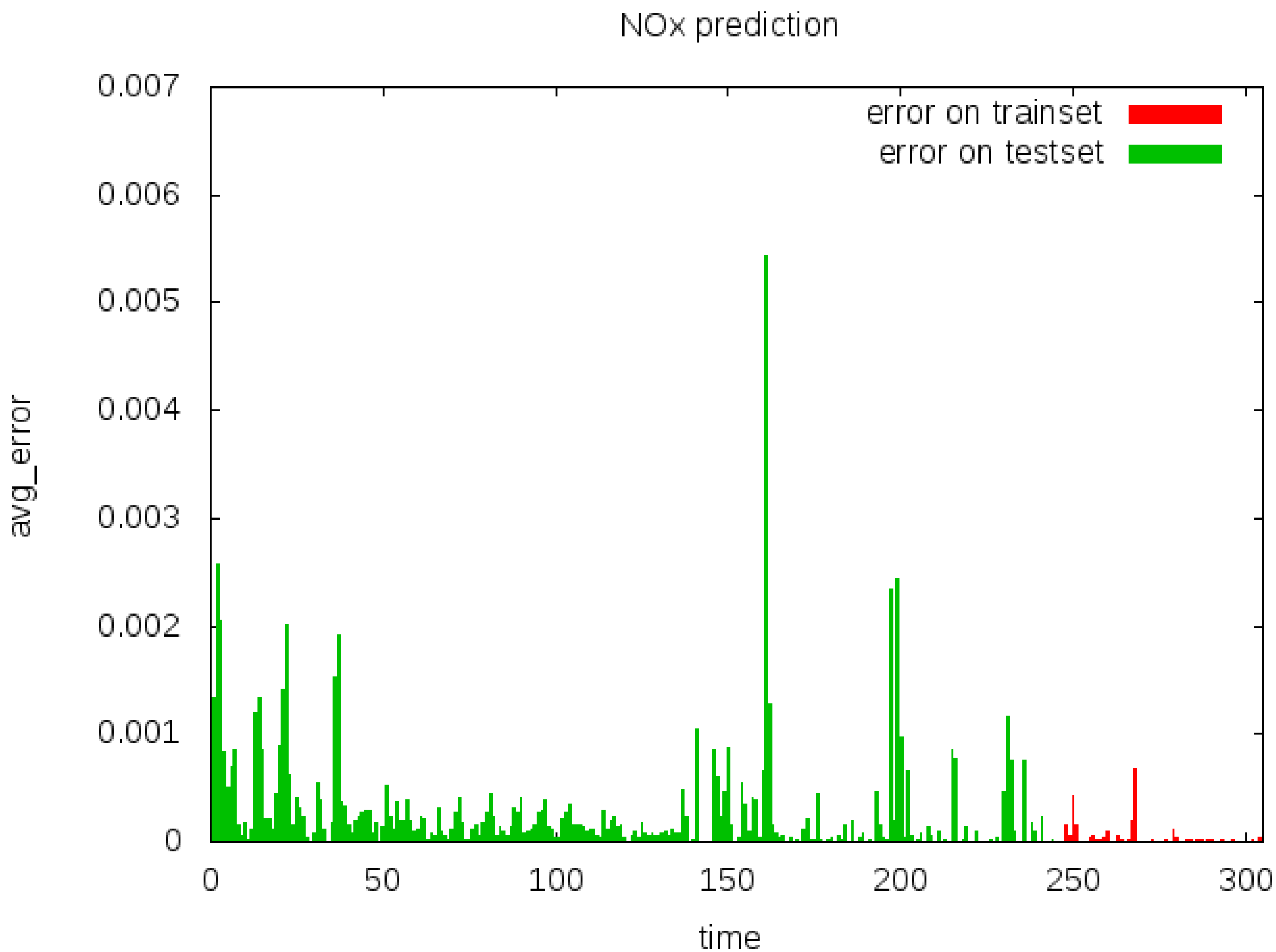
Experiment 2 – NOx



Experiment 2 – NOx



Experiment 2 – NOx




WHAT WORKED?

Conclusions

- Modeling with kernel networks works well for sensor data
- The evolutionary search for parameters was able to find better models in comparison to ad-hoc/standard techniques
- The resulting models are quite small and fast


WHAT
QUITE
NOT?

Challenges

- Missing data
  - Semi supervised learning (S. de Vito)
  - Surrogate models
- Large data
  - Meta-learning takes long time
  - Preprocessing
- Expert insight into data
  - Influence of factors like time of the year, …
  - Ensemble models

THANK YOU roman@cs.cas.cz