

Supplementary information to
“Machine learning-based time-series forecasting prevents electrode corrosion in
organic electrochemistry”

Josef Tausendschön*¹, Michael Pölzl¹, Nikola Petrovic¹, Jason Williams¹, Elisabeth Fink¹

¹ Research Center Pharmaceutical Engineering GmbH, Inffeldgasse 13, Graz, 8010, Austria

*corresponding author email: josef.tausendschoen@rcpe.at

Sections: A-C

Number of tables: 5

Number of figures: 3

Number of pages: 6

A. Experiments

Designed current paths were chosen with the idea of mimicking the cell behavior in case of a malfunction. In Figure S1, path (a) illustrates a process where the inhomogeneous layer at the electrode surface forms or dissipates gradually over time. Path (b), on the other hand, reflects a sudden and pronounced change at the electrode interface, which could result from mechanical damage to the electrode.

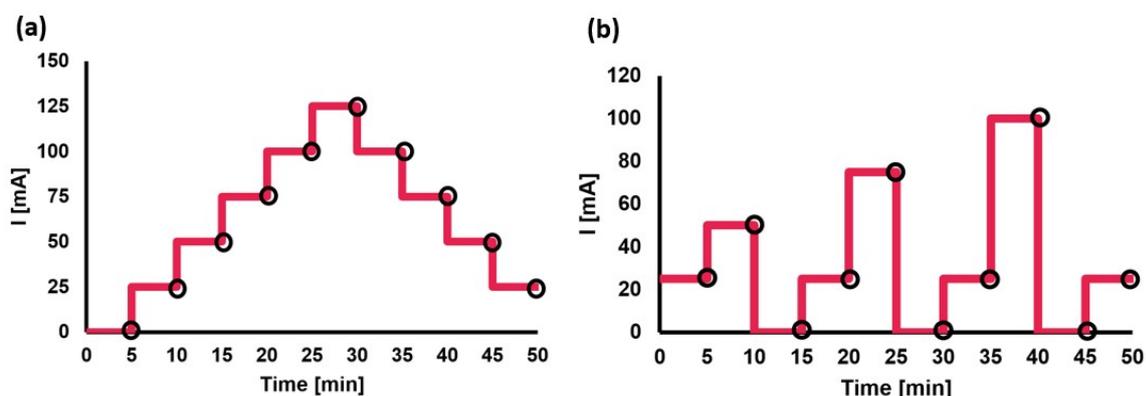


Figure S1: The two types of designed current path applied in the experiments. The black circles show when a HPLC sample was drawn from the tank.

Figure S2 shows the total experimental setup used.

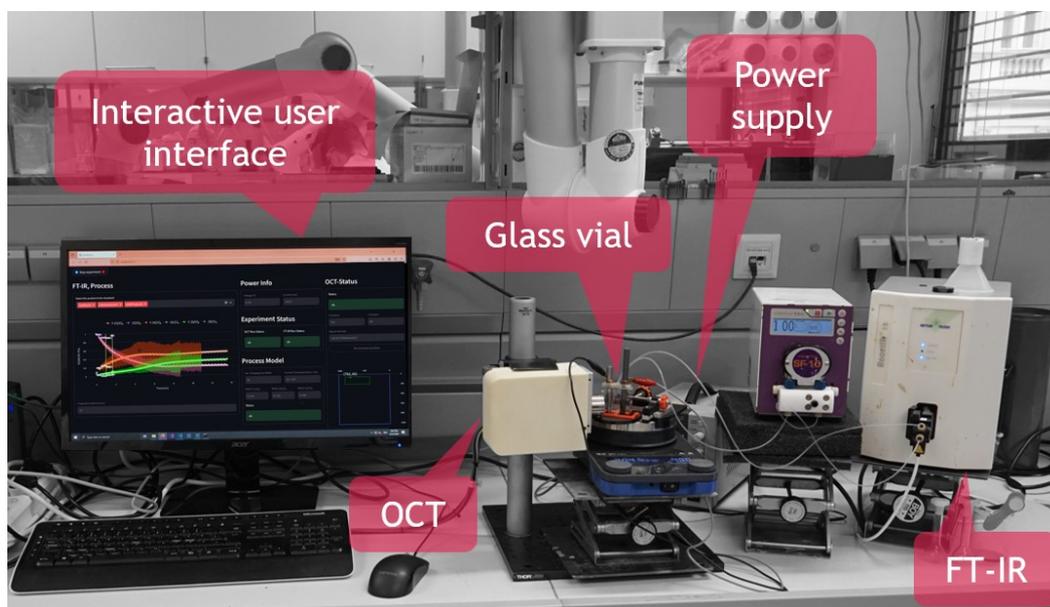


Figure S2: Total experimental setup.

All experiments performed and the analysis of the collected data can be found in Table S1. A single experiment was added to the dataset if each data source (power supply, FT-IR/PLS and OCT) yielded correct data.

Table S1: List of all performed experiments.

Experiment	Curren	Charg	HPL	Stirre	Solven	In FTIR dataset	In OCT
------------	--------	-------	-----	--------	--------	-----------------	--------

	t (mA)	e (F/mol)	C	d	t ratio		dataset
20230420_	125	4	No	Yes	40 : 1	No, neg. conc.	Yes
A							
20230420_	75	4	No	Yes	40 : 1	No, neg. conc.	No data
B							
20230421_	25	4	No	Yes	40 : 1	No, neg. conc.	No data
A							
20230421_	50	4	No	Yes	40 : 1	No, neg. conc.	Yes
B							
20230421_	100	4	No	Yes	40 : 1	No, neg. conc.	No data
C							
20230425_	125	6	Yes	Yes	40 : 1	Yes	Yes
A							
20230425_	100	6	Yes	Yes	40 : 1	Yes	No data
B							
20230429_	100	6	No	Yes	40 : 1	No, data logging failed	No data
A							
20230429_	100	6	Yes	Yes	40 : 1	Yes	No data
B							
20230502_	25	5	Yes	Yes	40 : 1	Yes	Yes
A							
20230502_	50	5	Yes	Yes	40 : 1	Yes	Yes
B							
20230503_	75	5	Yes	Yes	40 : 1	Yes	Yes
A							
20230507_	75	5	Yes	Yes	40 : 1	Yes	Yes
A							
20230507_	100	5	Yes	Yes	40 : 1	Yes	No data
B							
20230507_	125	5	Yes	Yes	40 : 1	Yes	Yes
C							
20230507_	Path (a)	Varyin	Yes	Yes	40 : 1	Yes	Yes
D		g					
20230508_	Path (b)	Varyin	Yes	Yes	40 : 1	Yes	No data
A		g					
20230508_	Path (b)	Varyin	Yes	Yes	40 : 1	Yes	Yes
B		g					
20230508_	Path (a)	Varyin	Yes	Yes	40 : 1	Yes	Yes
C		g					
20230508_	Path (b)	Varyin	Yes	Yes	40 : 1	Yes	Yes
D		g					
20230508_	Path (a)	Varyin	Yes	No	40 : 1	Yes	Yes
E		g					
20230508_F	Path (a)	Varyin	Yes	No	40 : 1	Yes	Yes
		g					
20230509_	Path (b)	Varyin	Yes	No	40 : 1	Yes	No, bad images
A		g					
20230509_	75	5	Yes	No	40 : 1	Yes	No, bad images
B							
20230509_	25	5	Yes	No	40 : 1	Yes	No, bad images
C							
20230509_	25	5	Yes	No	40 : 1	Yes	No, bad images
D							
20230509_	75	5	Yes	No	40 : 1	Yes	No, bad

E								images
20230509_F	Path (b)	Varying	Yes	Yes	80 : 1	Yes		No, bad images
20230511_A	25	5	Yes	Yes	80 : 1	Yes		No data
20230512_A	75	5	Yes	Yes	80 : 1	Yes		No, bad images
20230513_A	25	5	Yes	Yes	80 : 1	Yes		Yes
20230513_B	Path (b)	Varying	Yes	Yes	80 : 1	Yes		Yes
20240801_A	125	6	No	Yes	40 : 1	Yes		No data
20240801_B	125	6	No	Yes	40 : 1	Yes		Yes
20240801_C	75	6	No	Yes	40 : 1	Yes		No, bad images
20240801_D	100	6	No	Yes	40 : 1	Yes		Yes
20240802_A	12.5	6	No	Yes	40 : 1	No, neg. conc.		Yes
20240806_A	12.5	6	No	Yes	40 : 1	No, neg. conc.		No data
20240808_A	25	6	No	Yes	40 : 1	No, control failed		No data
20240808_B	50	6	No	Yes	40 : 1	No, data logging failed		No data

B. Model reliability

The variance of the predictions at timestep t_p and for species f is defined by:

$$s_{t_p, f}^2 = \frac{1}{T_{p_i \in t_p}} \sum_{i \in t_p}^{T_p} (z_i - \bar{z})^2 \quad (\text{S.1})$$

where T_p is the number of predictions available per timestep t_p , z_i is a prediction made at t_p and \bar{z} is the mean value of all forecasts at timestep t_p (e.g., the four predicted values at $t_p=15$ in the bottom panel of Figure S3). The first and last timestep of each rolling window are ignored in this analysis, as only one prediction is available (see bottom panel of Figure S3). The mean value of the standard deviation over all rolled up timesteps of a species f is then:

$$\bar{s}_f = \frac{1}{M} \sum_m^M \sqrt{s_{t_p, m}^2} \quad (\text{S.2})$$

where M is the total number of timesteps available in test dataset. The mean value among all three species is then taken.

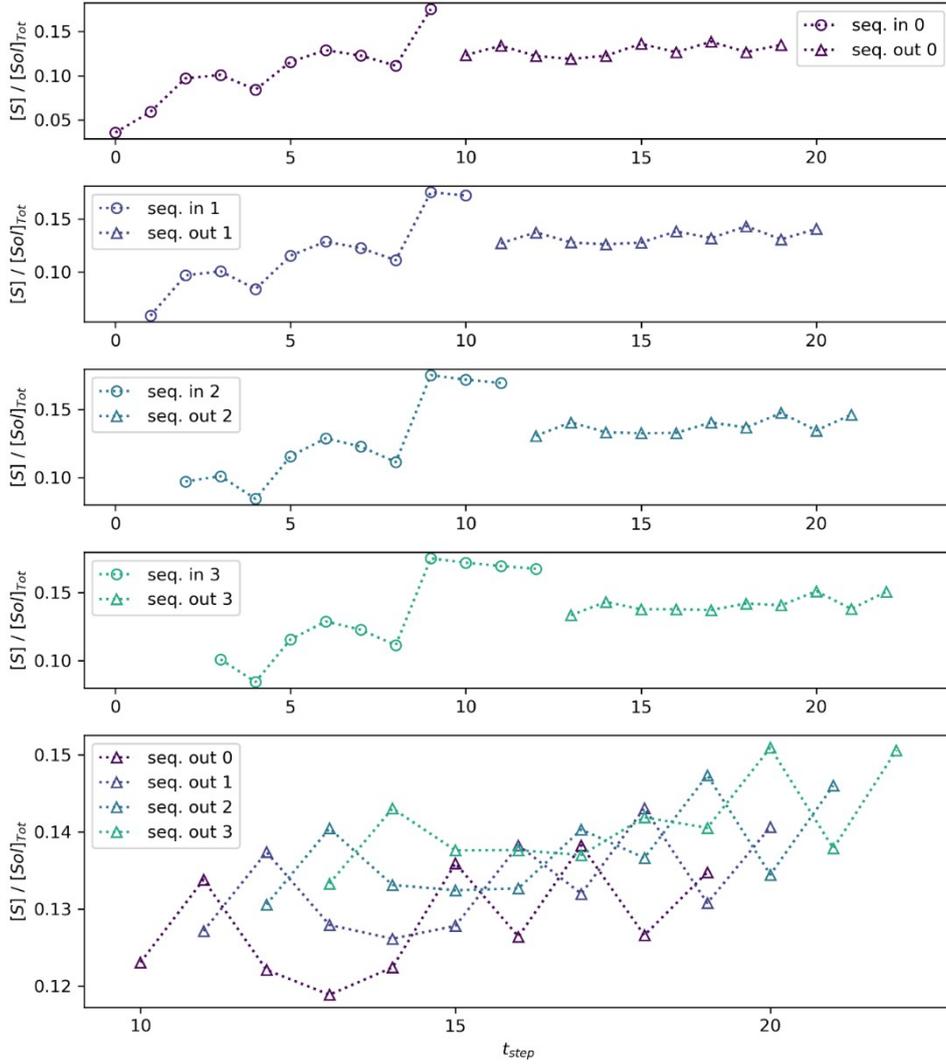


Figure S3: Rolling window model reliability on sequences of a random test batch. The four top panels show the rolling input and output sequences. The bottom panel summarizes all predicted output sequences.

C. Architecture and optimization results

The architecture of all model types is summarized in Table S2.

Table S2: Model architecture explanation.

Model type	Architecture
CNN	<ul style="list-style-type: none"> - Input (lambda) layer: Selects the searched kernel size from the input sequence. - 1D convolutional layer: Applies convolution over the selected window with searched number filters and kernel size, with ReLU activation. - Dense layer: Fully connected layer with searched hidden nodes and searched activation function. - Output (reshape) layer: Reshapes output to m future timesteps.
RNN (LSTM)	<ul style="list-style-type: none"> - Sequence input layer: Reads time-series input. - Conditional input layer: Receives conditional feature to initialize recurrent states.

- **Conditional recurrent wrapper with LSTM layer:** Wraps an LSTM with searched number of units. The conditional input is projected into the LSTM's initial hidden (and cell) state(s), modulating sequence processing. Returns only the final hidden state, which is then used for prediction.
- **Dense layer:** Projects the final LSTM hidden state into a flattened prediction, with the searched activation function.
- **Output (reshape) layer:** Restores temporal structure for multi-step output with m steps.

Autoregressive RNN (LSTM)	<ul style="list-style-type: none"> - Sequence input layer: Reads time-series input. - Conditional input layer: Receives conditional feature to initialize recurrent states. - Initial state computation: Each conditional input is passed through a dense layer to map it to the LSTM hidden dimension. If multiple conditions are given, their projected states are combined by a learned weighting into a single initial state. - LSTM cell: Core recurrent unit with searched number of units hidden size. - RNN layer with LSTM cell (warmup stage): Unrolls the LSTM across the observed input sequence to compute the final hidden and cell states, which serve as the starting point for prediction. - Autoregressive decoding loop: Instead of predicting all timesteps in parallel, the model generates outputs sequentially: the first prediction comes from the warmup state, then each subsequent prediction is fed back as input to the LSTM cell to produce the next output. - Dense layer: Projects each LSTM hidden state in the loop into the feature space using the searched activation. - Output (reshape) layer: Stacks predictions across timesteps and transposes to the final output shape.
----------------------------------	---

In Table S3 all found model and training parameters are listed from the optimization without the corrosion feature.

Table S3: Found model and training parameters without the corrosion feature.

Model parameters	CNN	LSTM	AR-LSTM
Number of filters	50	-	-
Kernel size	2	-	-
Number of hidden nodes	72	48	50
Input sequence length	15	14	15
Output sequence length	13	10	6
Output activation function	"Linear"	"Linear"	"Linear"
Training parameters			
Batch size	38	11	15
Epochs	28	29	29
Learning rate	9.986e-3	2.674e-2	3.425e-2

Table S4 lists all found model and training parameters from the optimization considering the corrosion feature.

Table S4: Found model and training parameters considering the corrosion feature.

Model parameters	CNN	LSTM	AR-LSTM
Number of filters	85	-	-
Kernel size	1	-	-
Number of hidden nodes	64	12	89
Input sequence length	15	15	15
Output sequence length	7	10	7
Output activation function	"Linear"	"Linear"	"Linear"
Training parameters			
Batch size	31	21	34
Epochs	47	23	32
Learning rate	3.499e-3	2.832e-2	3.796e-3

D. OCT Dataset

Table S5 summarizes the statistics of the parabola thickness measured from the first 1000 images of processable OCT data. It is assumed that corrosion levels are not yet severe within these initial images; therefore, the average parabola thickness (height) can be meaningfully evaluated.

Table S5 Parabola height statistics

Experiment	Parabola thickness mean (pixel)	Parabola thickness std. (pixel)	Parabola thickness max (pixel)
20230420_A	16.155	2.443	29
20230421_B	17.619	1.950	30
20230425_A	12.804	2.916	25
20230502_A	14.627	1.903	24
20230502_B	10.022	5.553	27
20230503_A	11.557	2.185	26
20230507_A	6.289	4.762	20
20230507_C	9.217	4.485	28
20230507_D	5.553	4.641	18
20230508_C	15.018	1.675	29
20230508_D	6.369	4.936	22
20230508_E	23.407	2.526	34
20230508_F	11.042	8.350	29
20230513_A	13.422	2.479	21
20230513_B	14.691	1.486	26
20240801_B	9.273	4.450	18
20240801_D	9.13	4.117	14
20240802_A	13.387	1.134	16