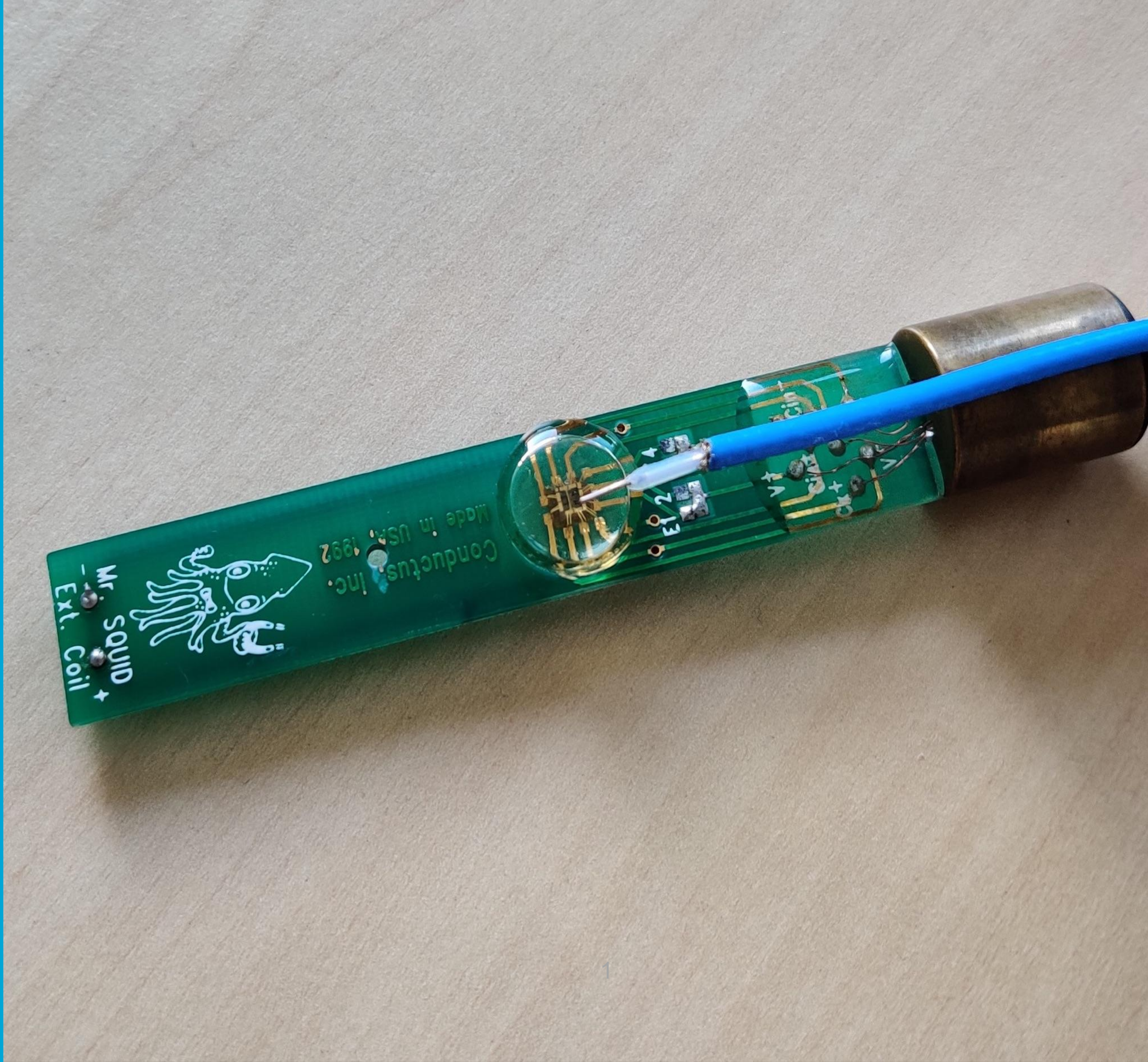
Using Machine Learning to Observe and Understand Microwave Induced Steps in the IV of a SQUID

19 January 2022

Nick Verhoeks, Bram Wagemakers, Lennart Bult

TU Delft



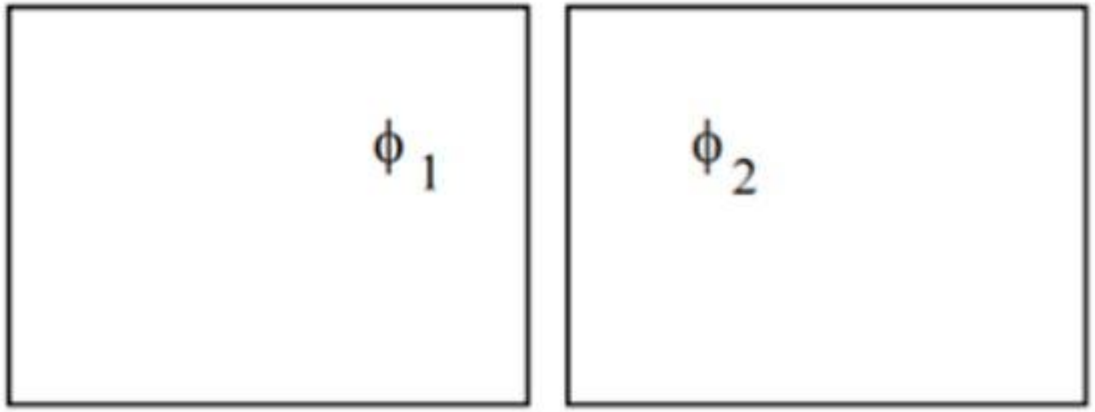
Content

- Theory
- Setup
- Measurements
- Machine learning
- Shapiro Steps
- Conclusion

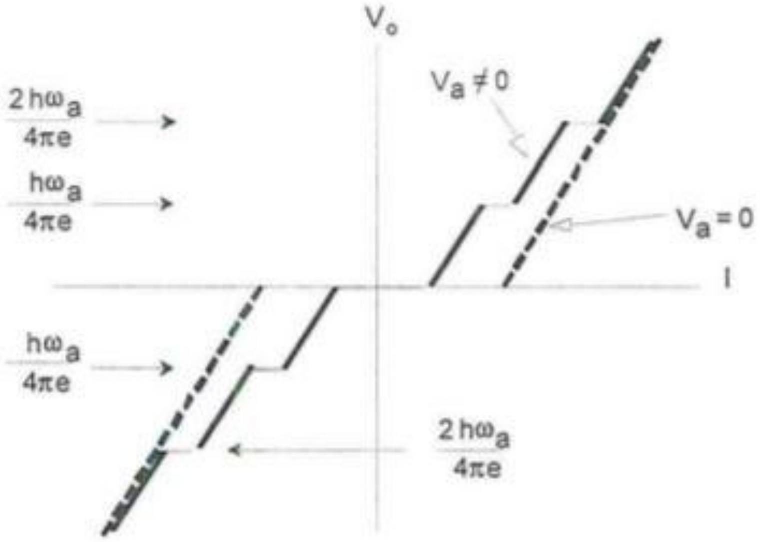
TUDelft

Theory

- Expanding wavefunctions
- Overlap
- Related
- Electrical current
- AC Josephson effect



Theory



$$V(t) = V_0 + V_a \cos(\omega_u t)$$

$$\sin(X \sin q) = \sum_{n=1}^{\infty} J_N(X) \sin(nq)$$

$$J(t) = J_0 \sum_{n=1}^{\infty} (-1)'' J_n \left(\frac{4\pi e V_a}{h\omega_e} \right) \sin \left[\delta'(0) + \left(n\omega_a - \left(\frac{4\pi e V_0}{h} \right) \right) \right]$$

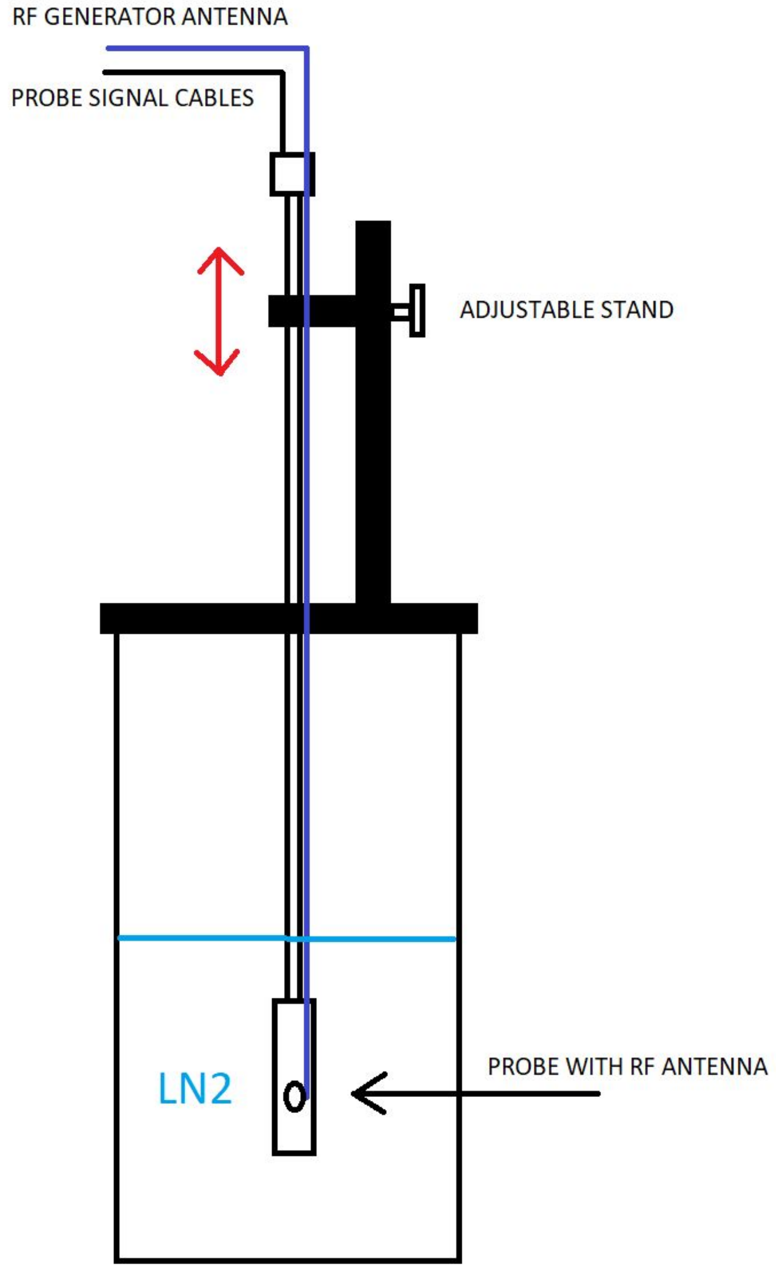
$$V_0 = nh\omega_\omega\ 4\backslash\mathrm{pi}$$

TU Delft

Experimentation Method: Setup

Components:

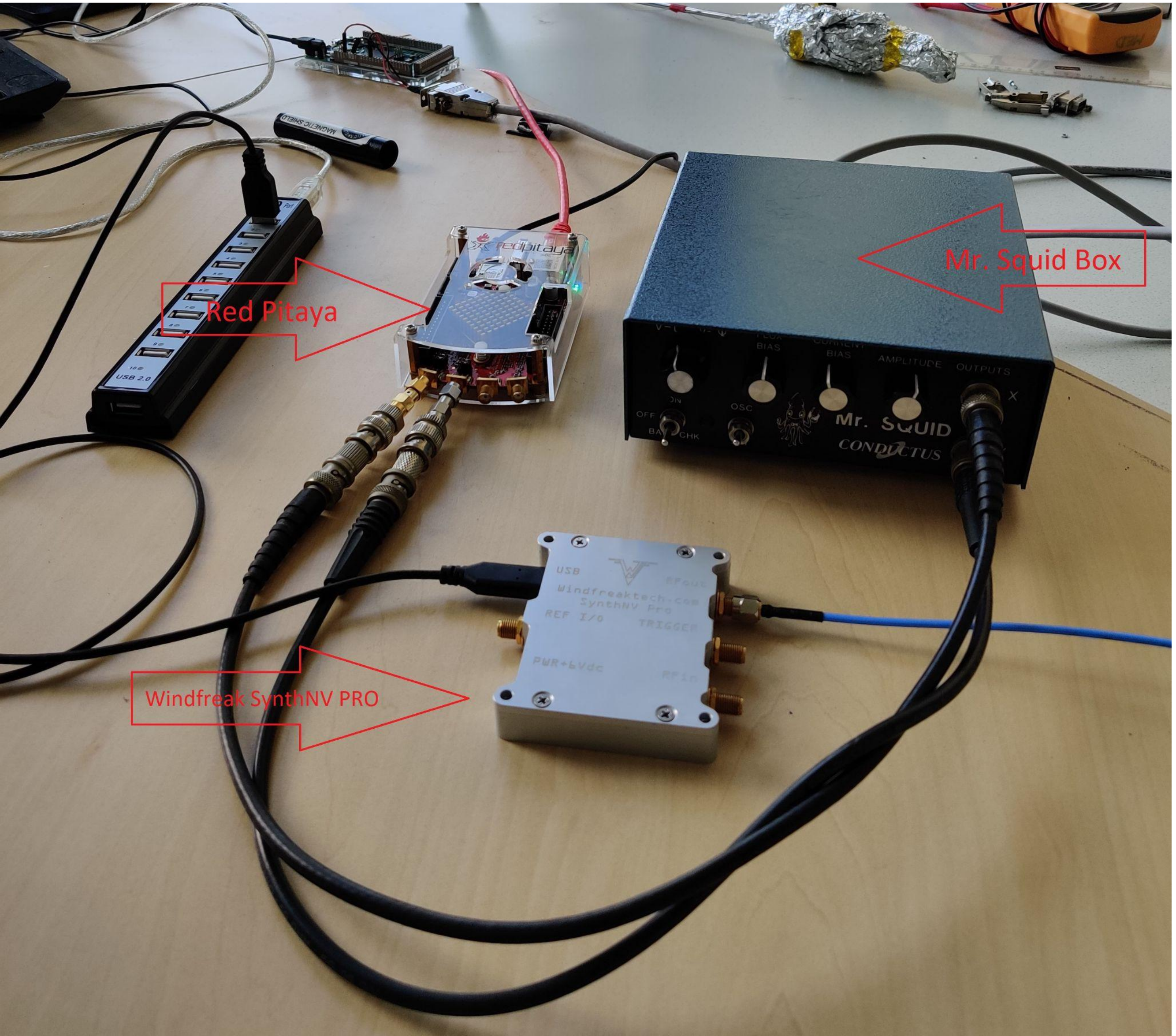
- Mr. Squid probe (superconducting quantum interference device)
- Mr. Squid box (probe control box)
- Red Pitaya (RF applications control board)
- Windfreak SynthNV PRO (RF signal generator)
- Dewar with cryogenic liquid (LN2)
- Windows desktop PC (with custom Python measurement software)

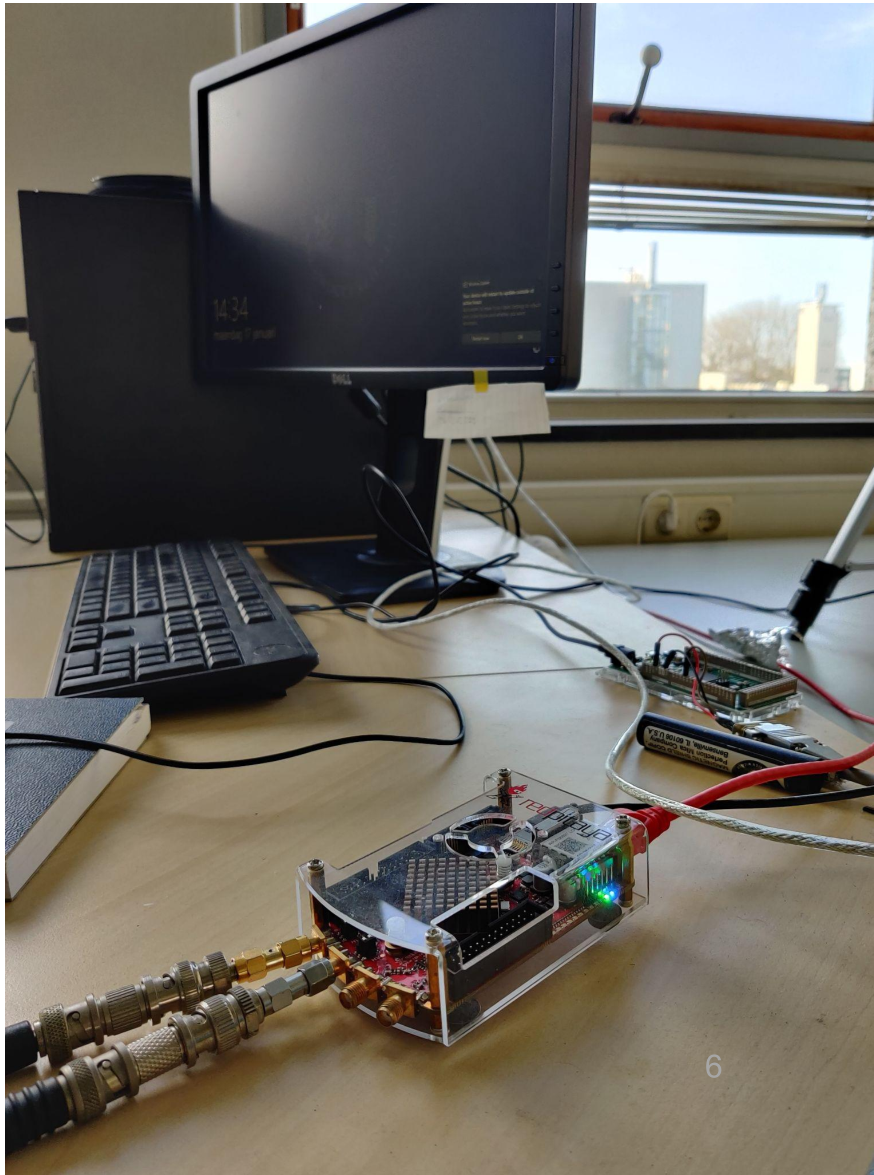


Schematic view of the setup.

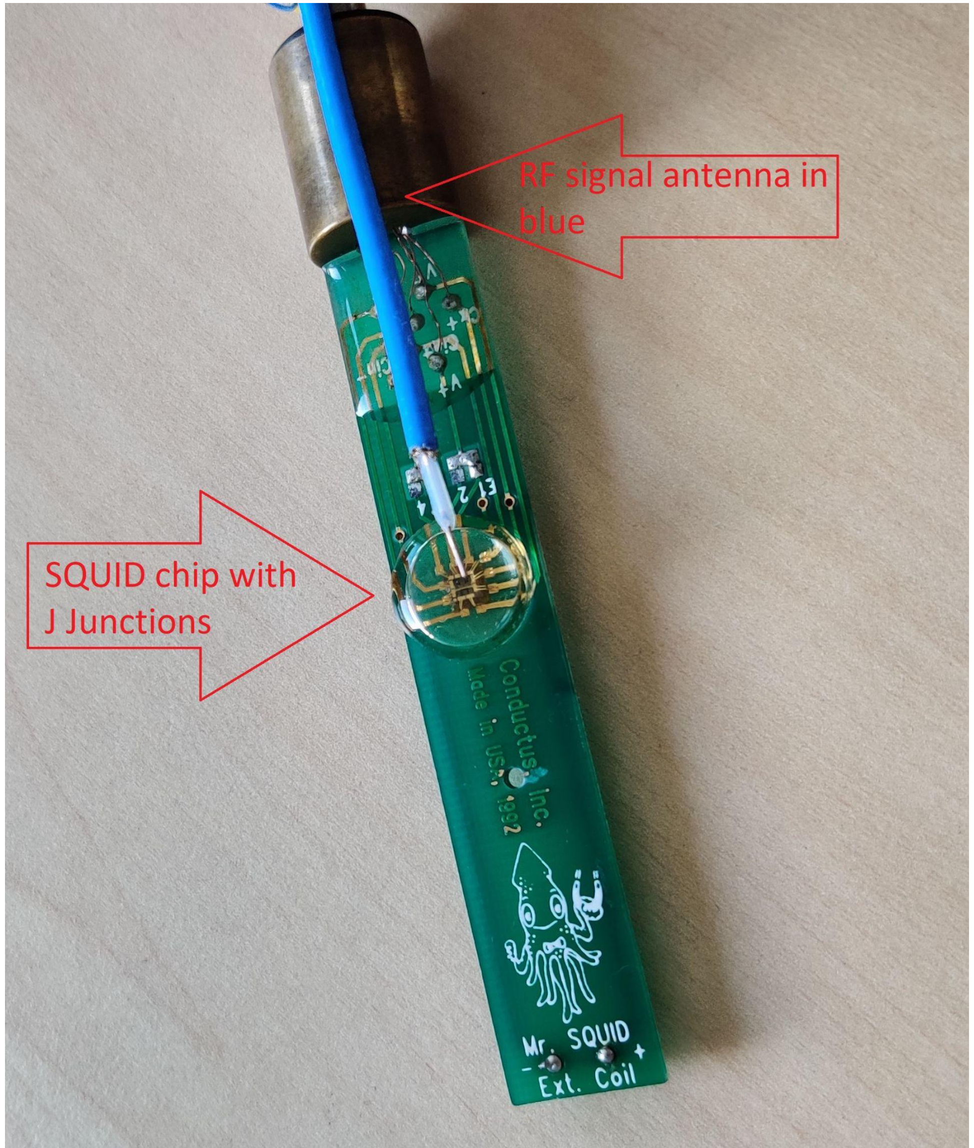
TU Delft

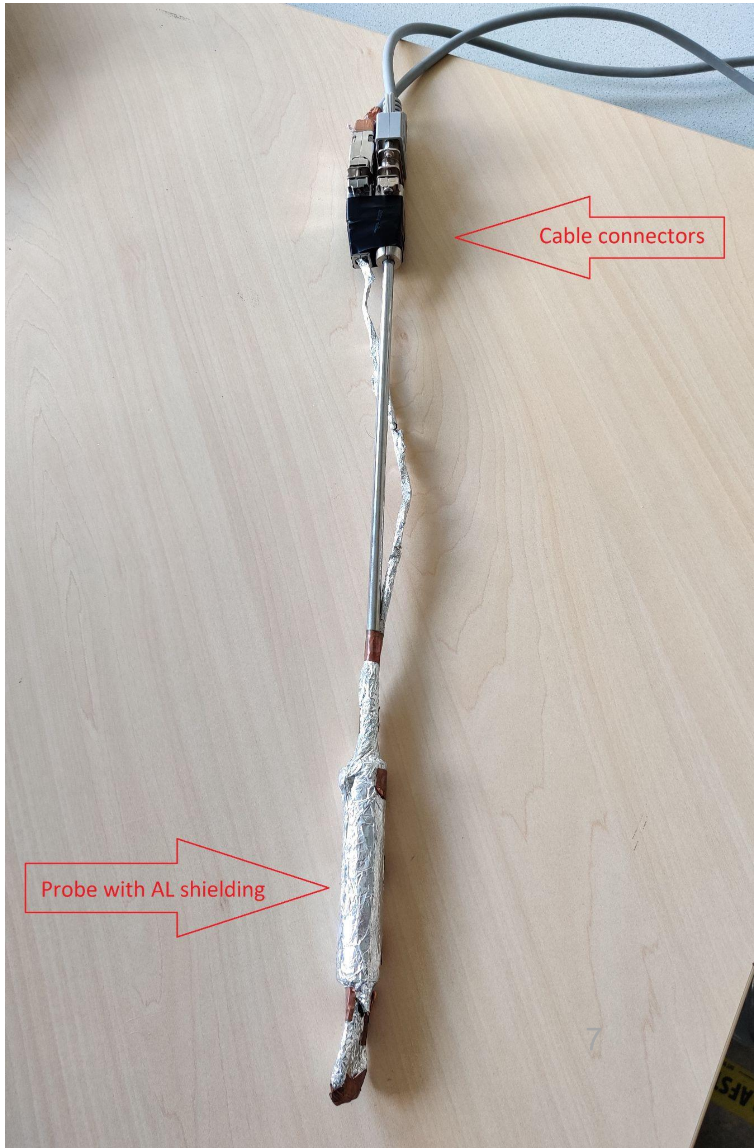
Experimentation Method: Setup - Photos Part 1





Experimentation Method: Setup - Photos Part 2

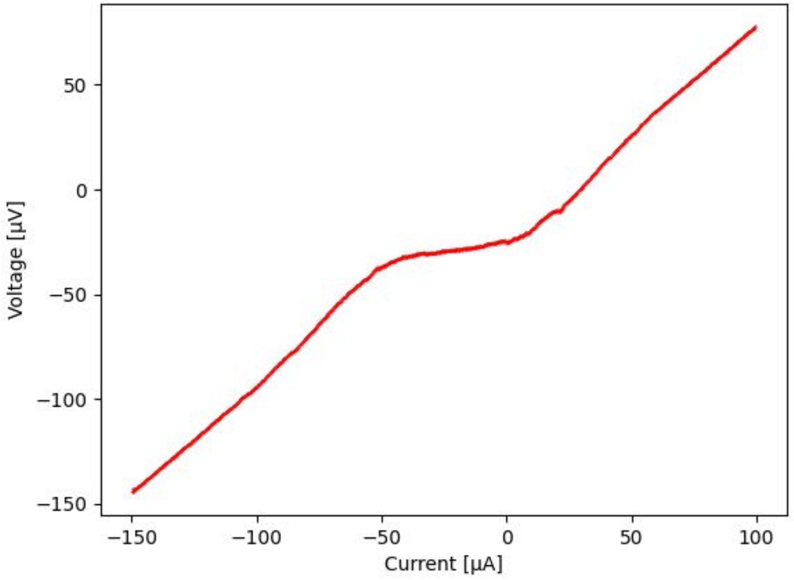




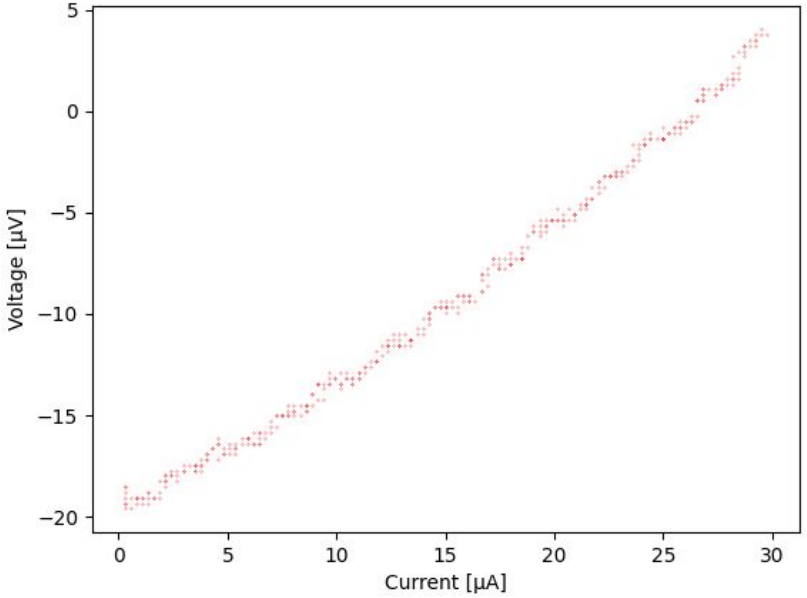
Experimentation Method: Measurements

Order of operations:

- Cooling down probe to 77K
- Mr. SQUID Box settings
    - Adjust Current Bias, Amplitude, (and flux bias)
- Visualization of IV curves
- Automatic data collection of IV curve data points
    - Sweep: 1000 - 5000MHz and -50 - 0dBm



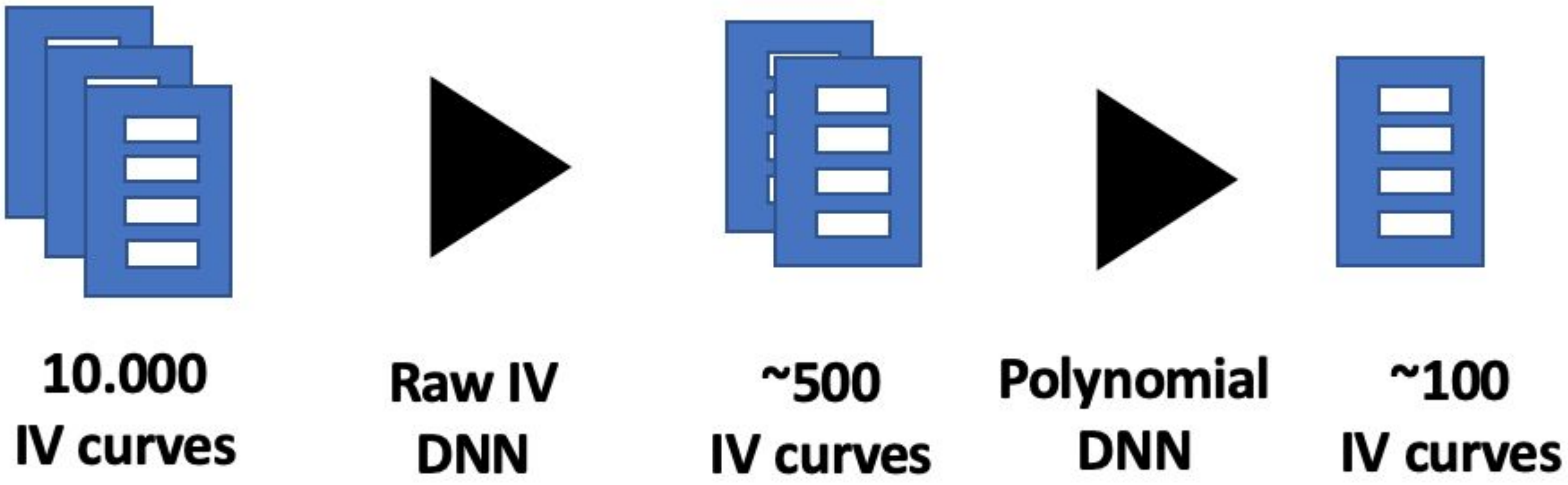
IV curve of superconducting probe



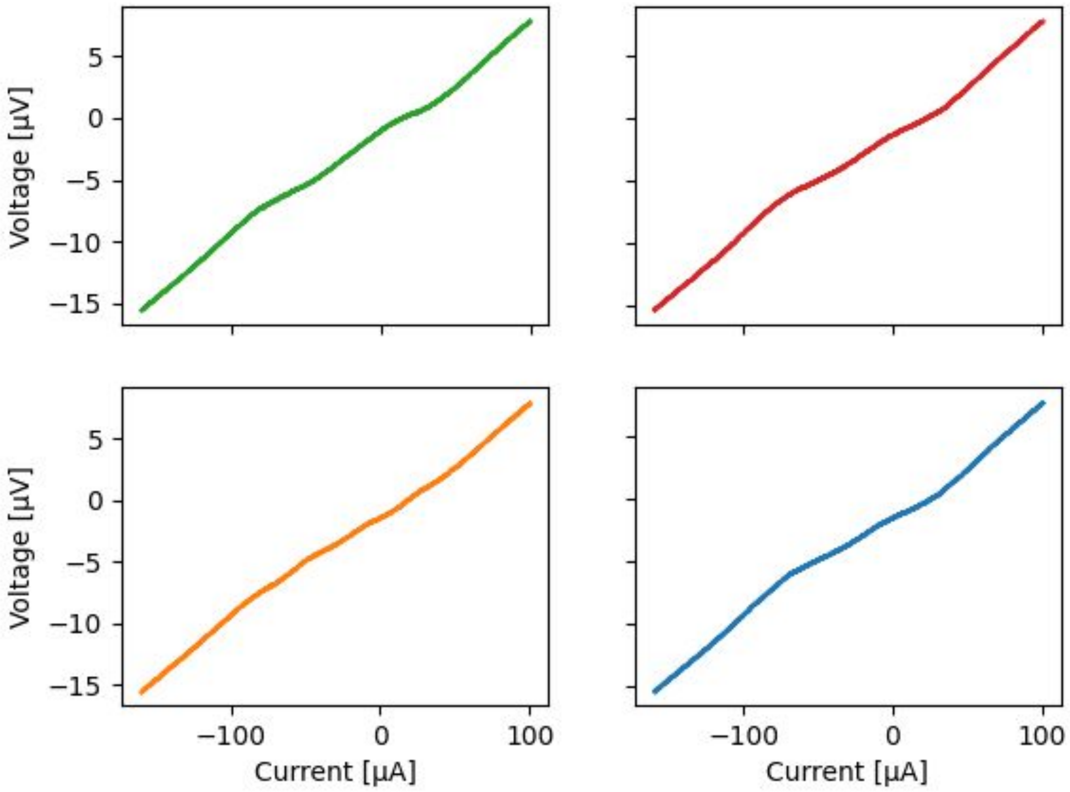
IV curve with applied RF signal

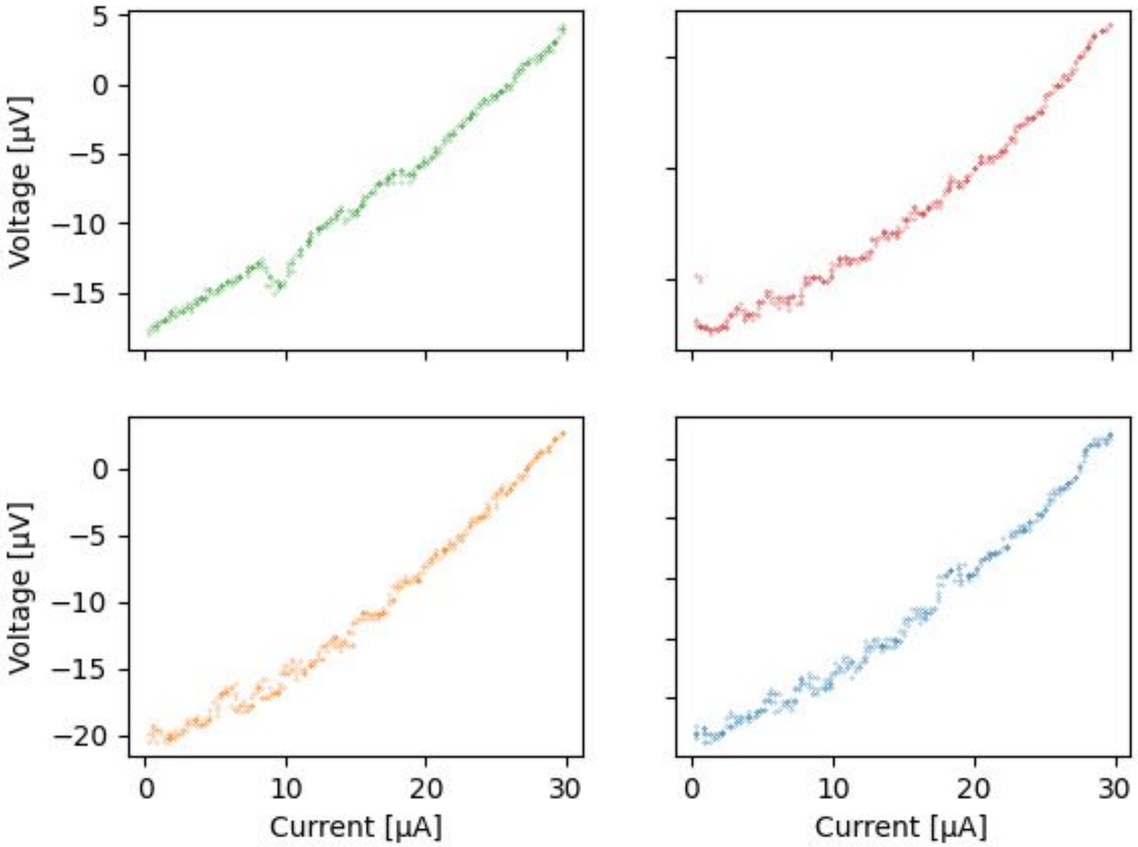
TU Delft

Machine Learning Method

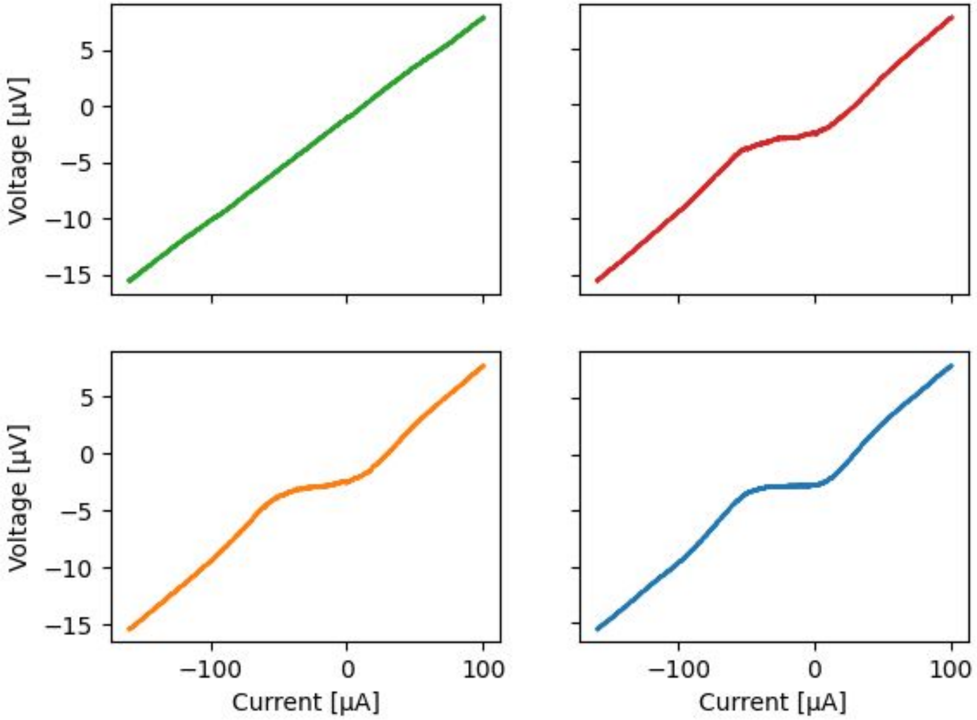


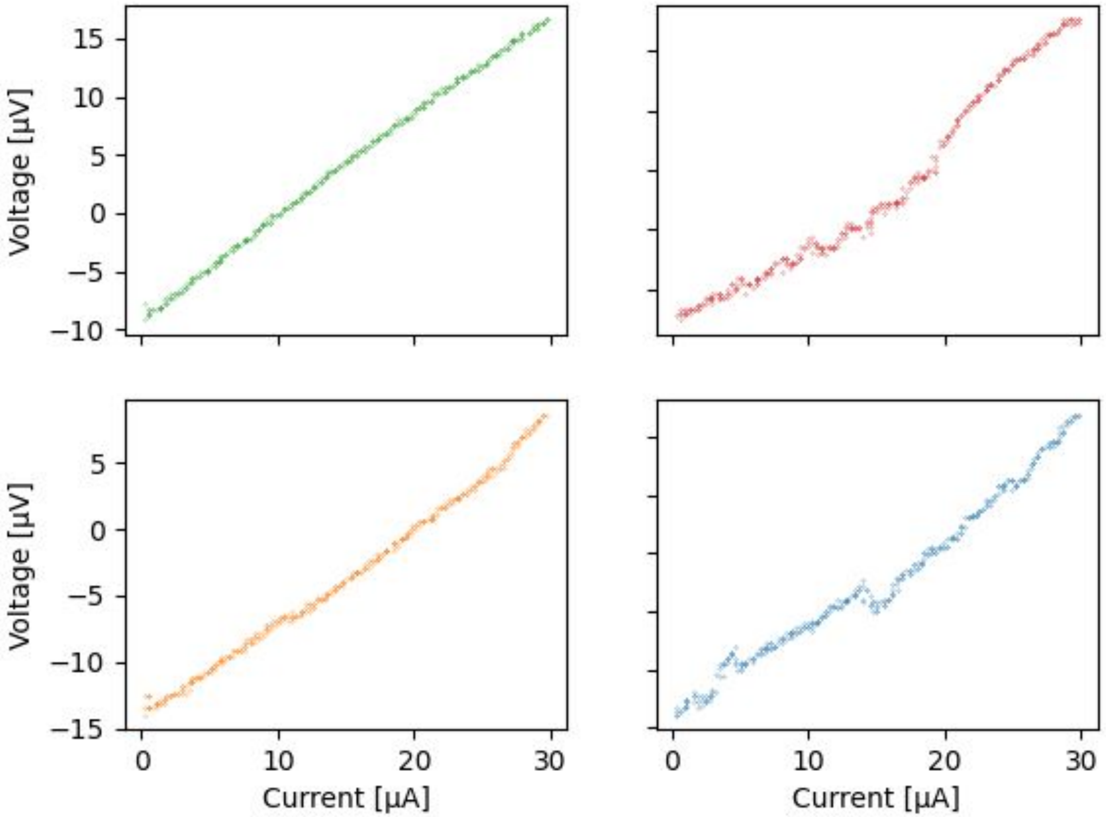
ML Method: what do we want the model to do?



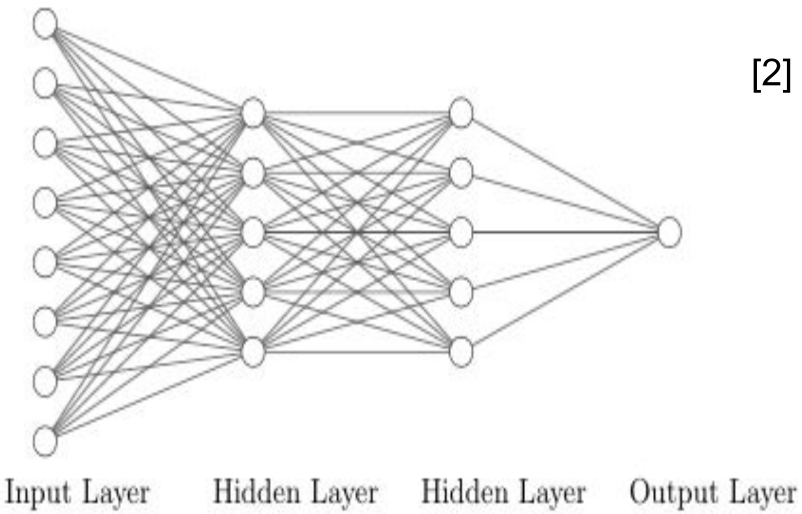


ML Method: what do we want the model *NOT* to do?





ML Method: MLPClassifier



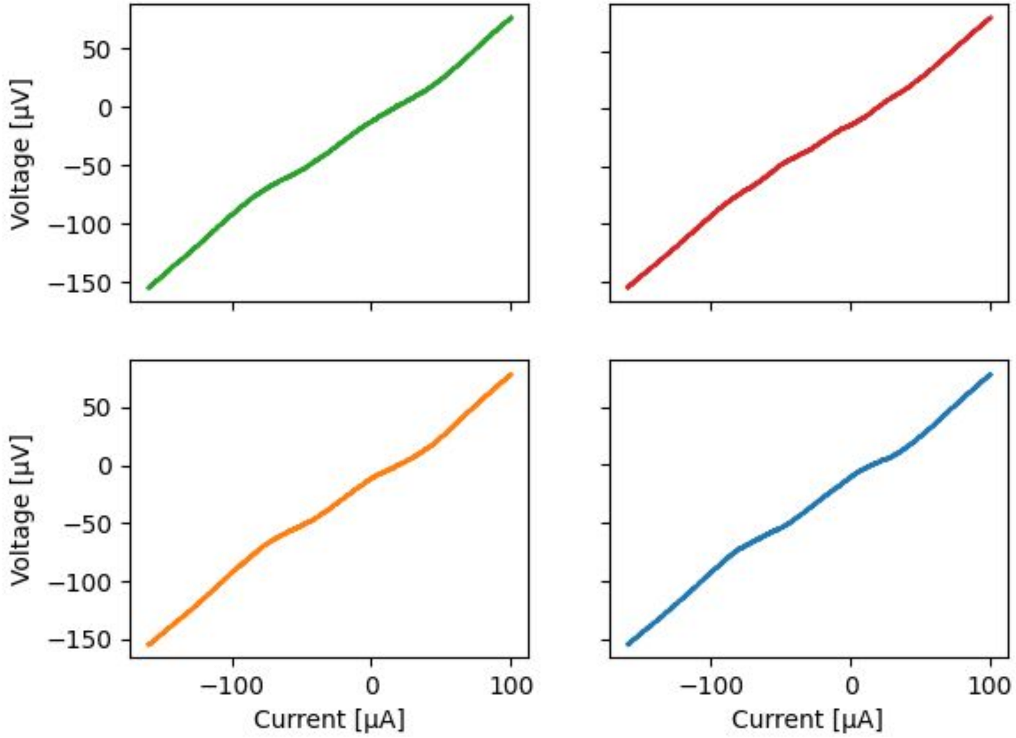
[2]

- MLP = Multilayer Perceptron
- sklearn.neural_network.MLPClassifier
- Mapping an input to an output through hidden layer(s)
- Requires unambiguous training examples
- Not the most elegant solution, but it works!
- DNN = deep neural network

[1]

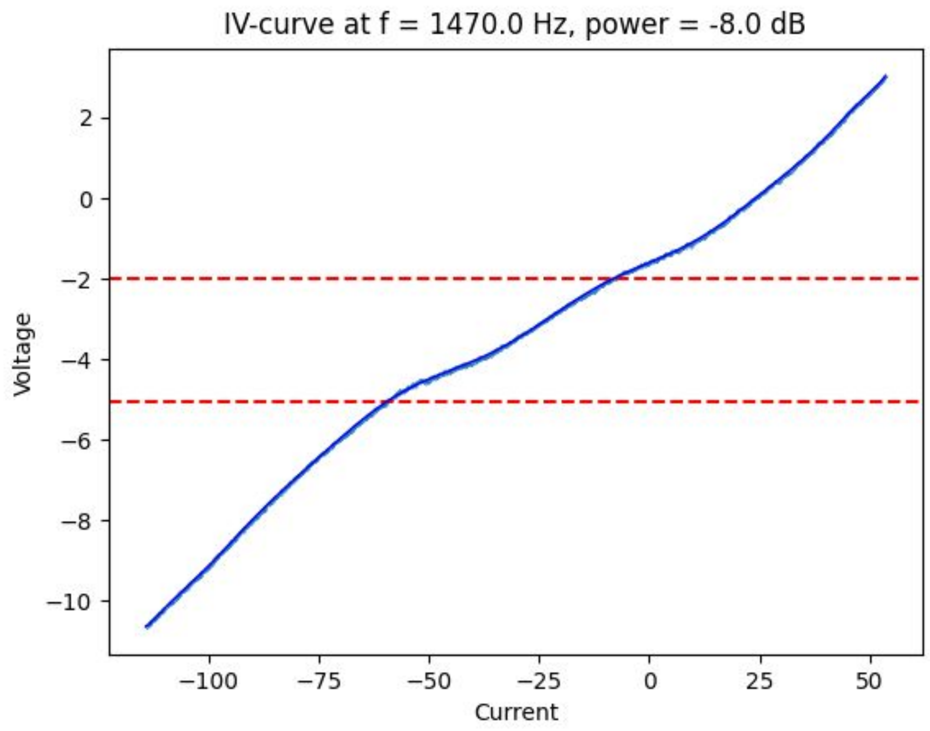
ML Method: Raw IV curve

- Use ~8000 voltage data points as input
- Output
  - “0” -> no steps detected
  - “1” -> steps detected

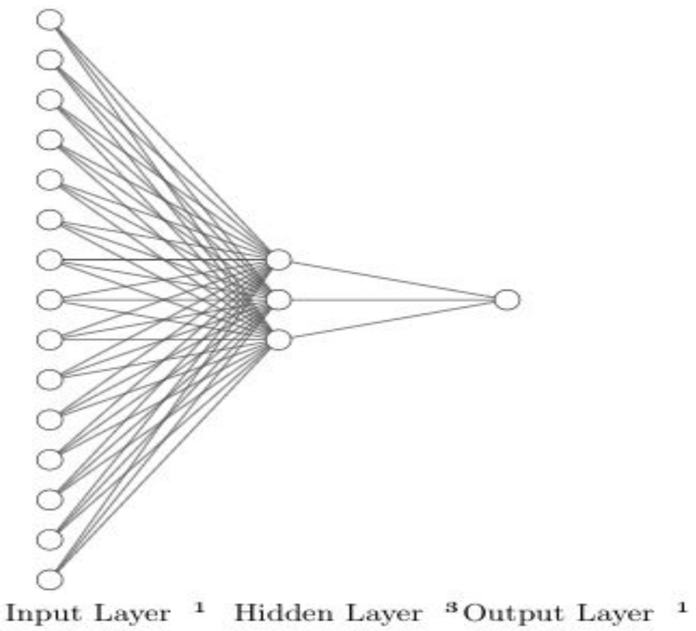


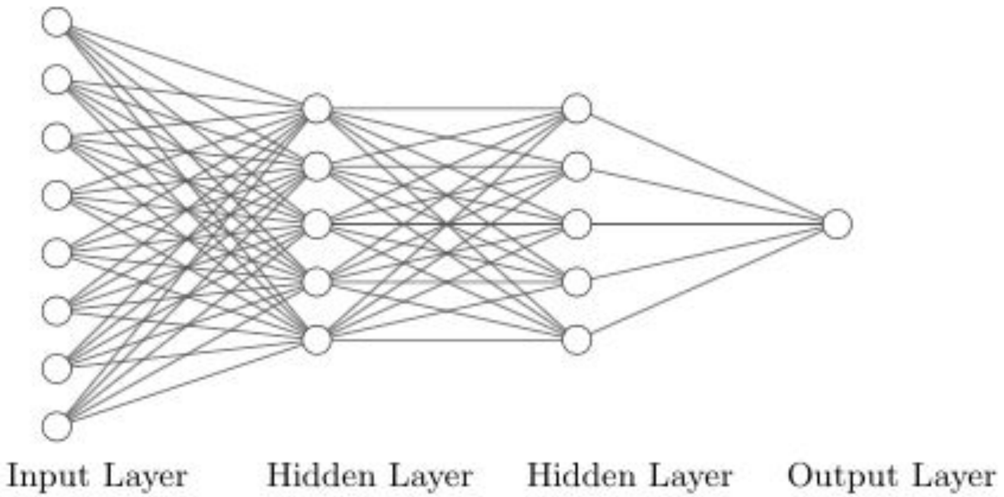
ML Method: Polynomial Fit

- Complement the Raw IV curve MLP
  - “Regular” superconducting hard to detect and filter
- Use 8 polynomial coefficients as input
- Output
  - “0” -> no steps detected
  - “1” -> steps detected



ML Method: MLP/DNN Architecture





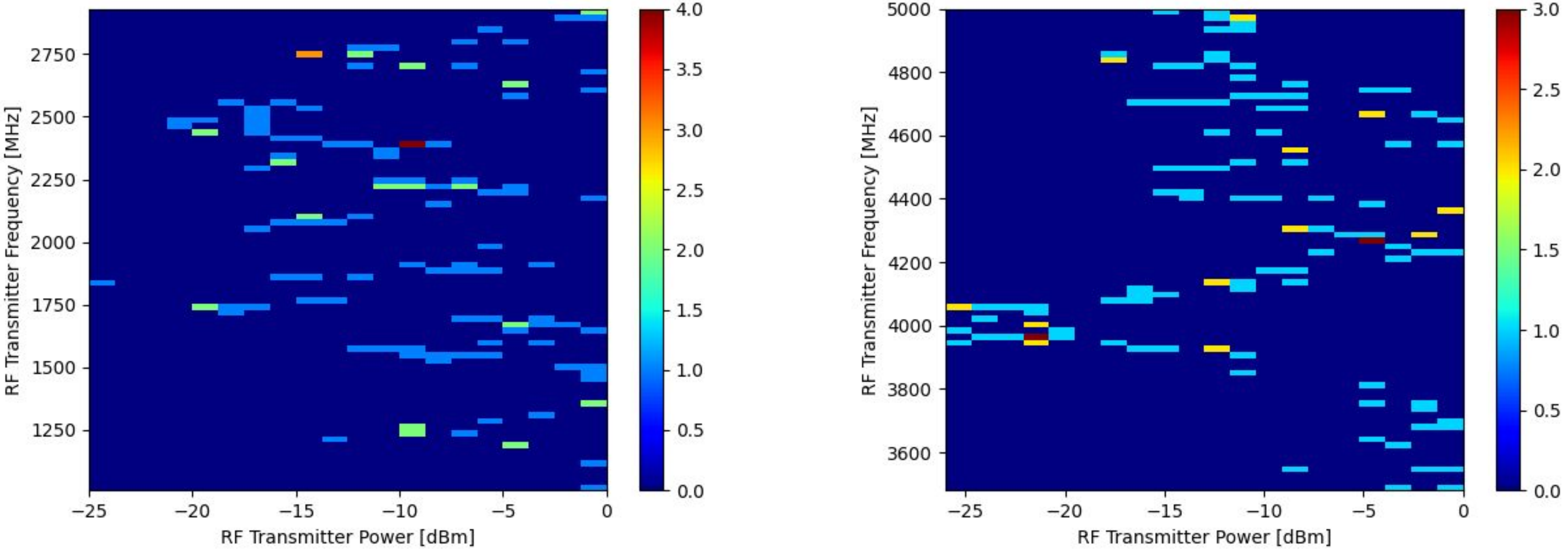
Raw IV MLP

Polynomial MLP

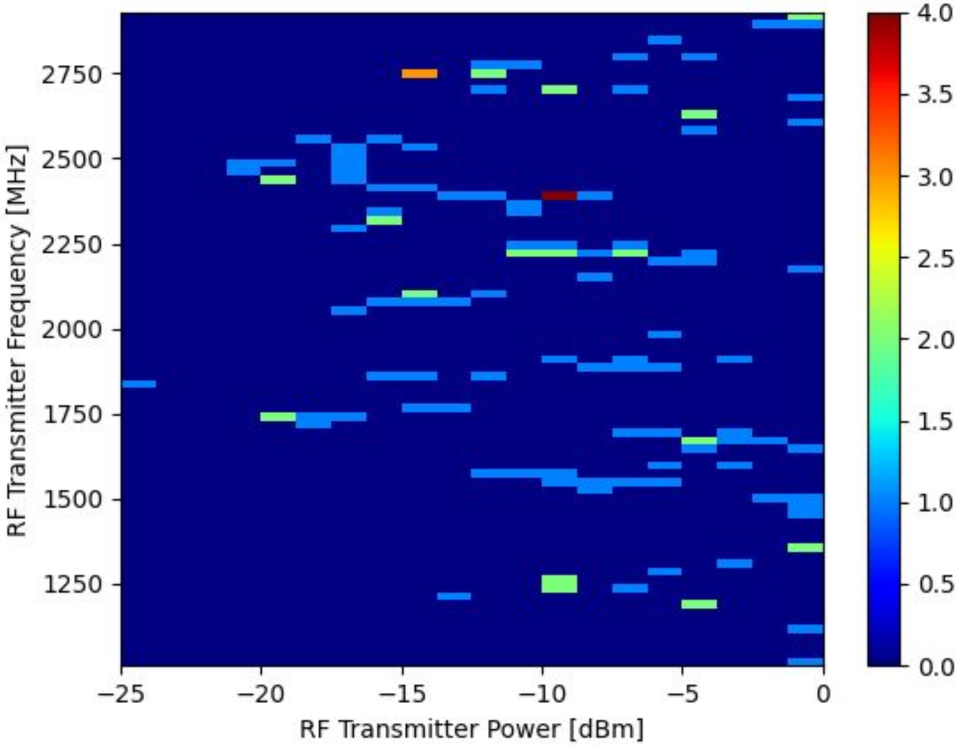
ML Method: Training Performance

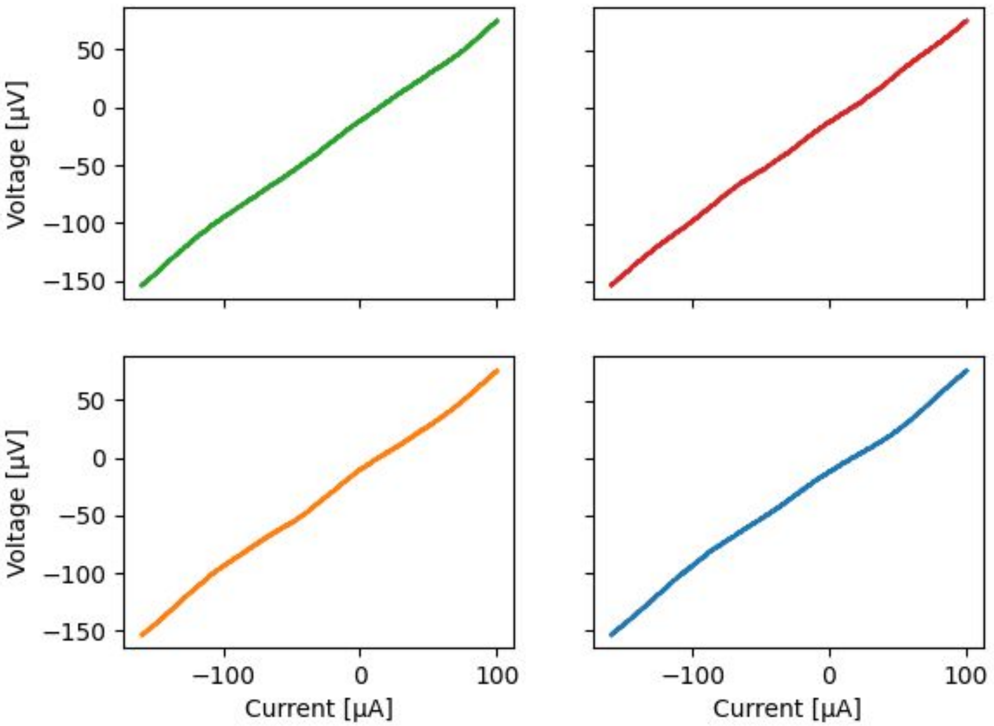
Model Name	Performance Training Set [%]	Performance Verification Set [%]
raw_IV_model_v1	59.6	20
raw_IV_model_v2	83.7	40
raw_IV_model_v3	87.9	80
raw_IV_model_v4	96.6	100
poly_model_v1	89.9	100
poly_model_v2	92.4	100
poly_model_v3	86.1	100
poly_model_v4	91.1	100

ML Method: Results - Frequencies and Powers



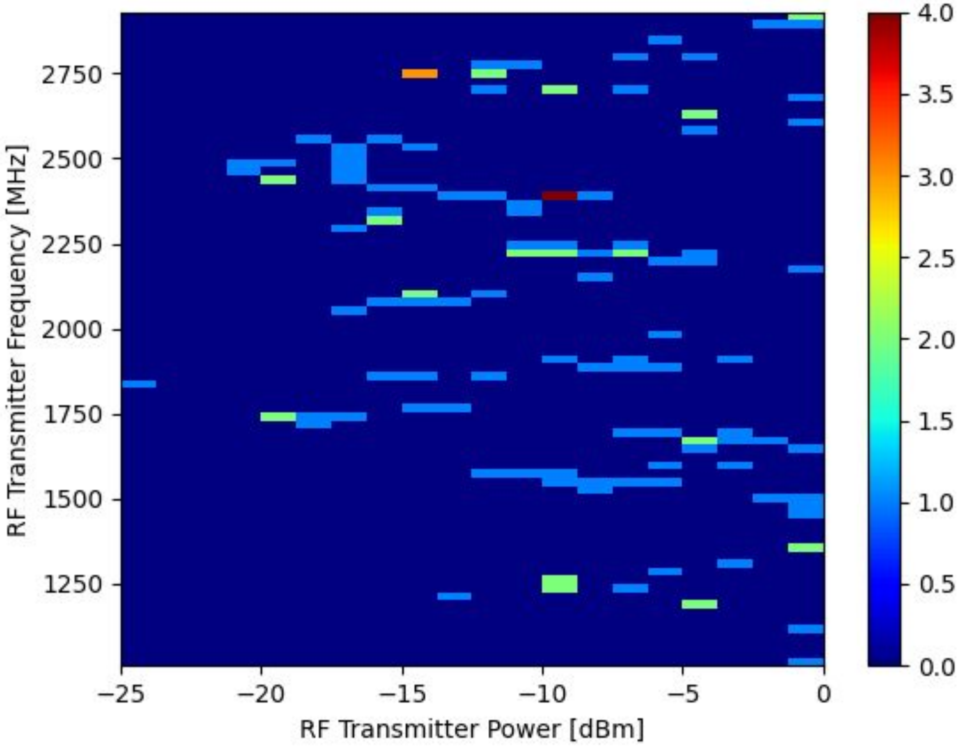
ML Method: Results - Random Samples - f = 1 to 3 GHz

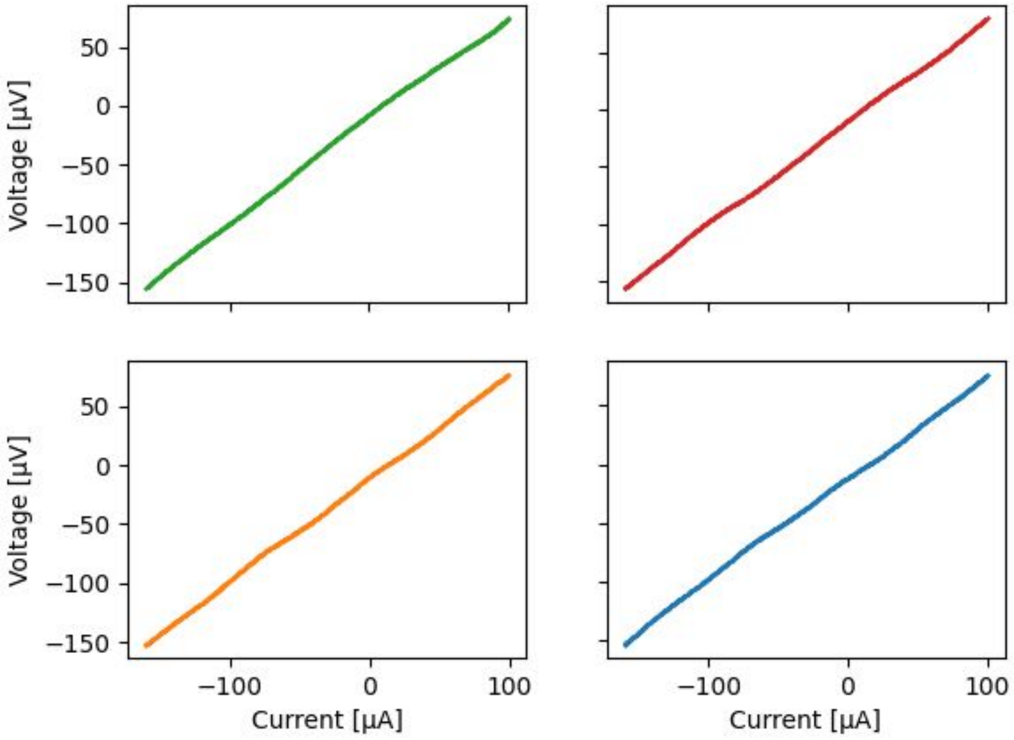




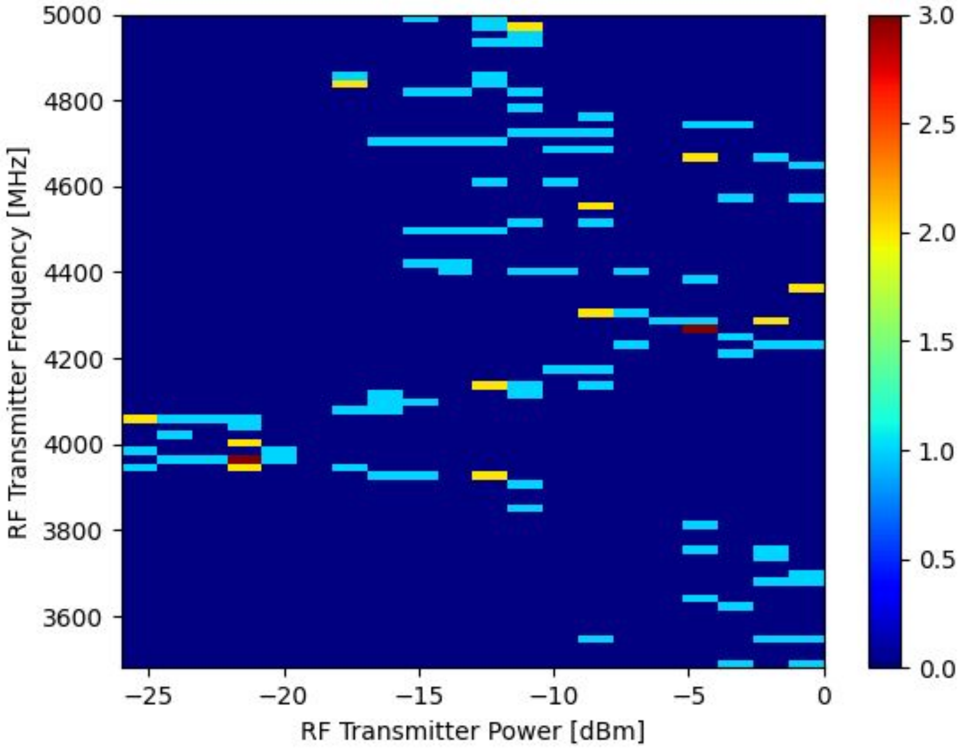
TU Delft

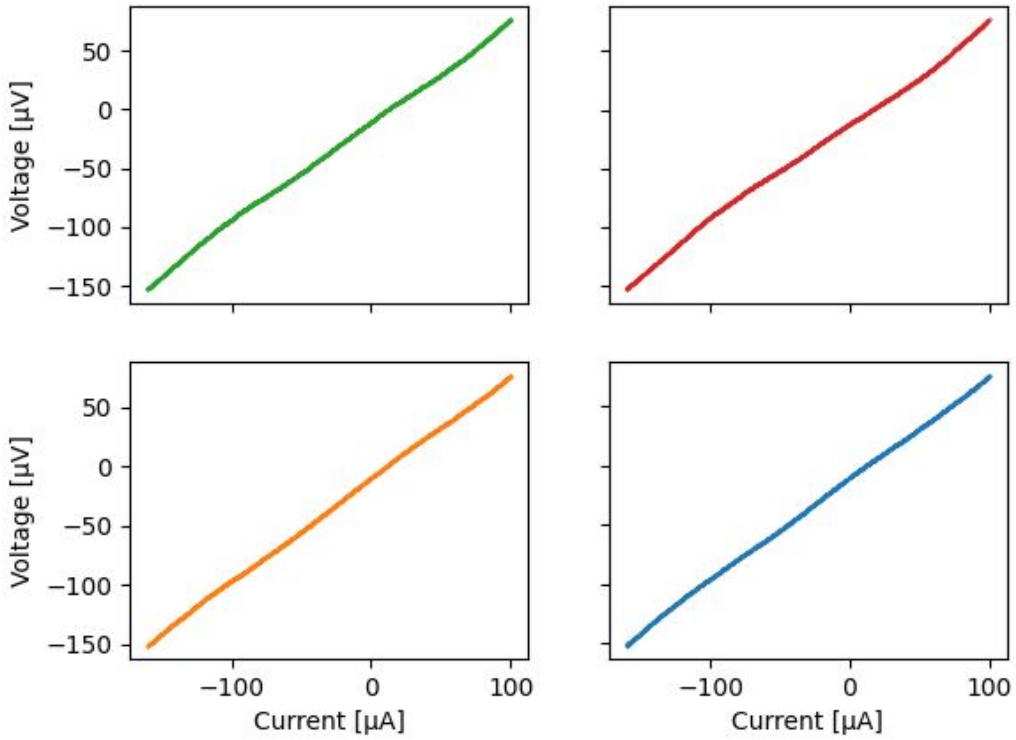
ML Method: Results - Random Samples - f = 1 to 3 GHz





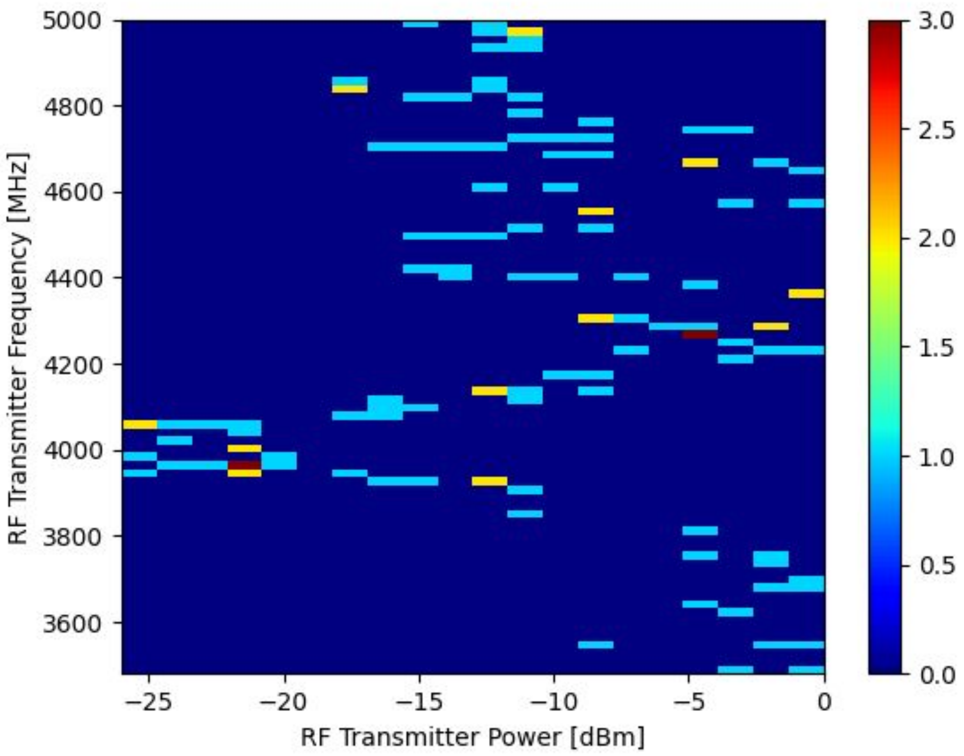
ML Method: Results - Random Samples - f = 3 to 5 GHz

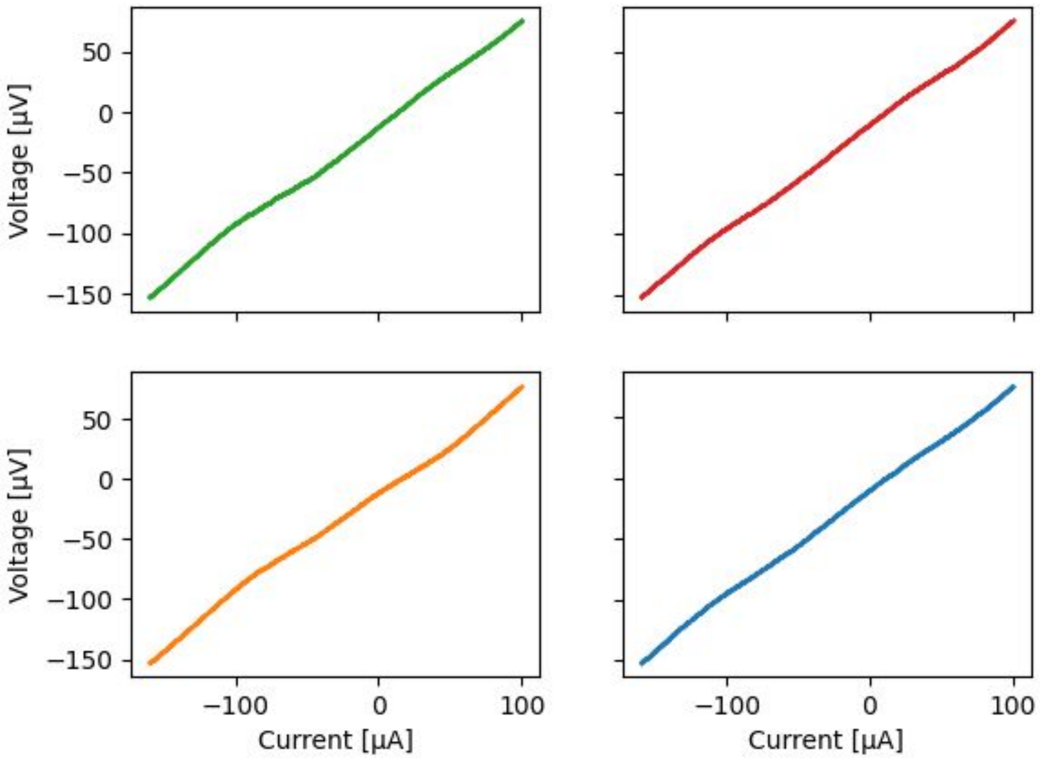




TUDelft

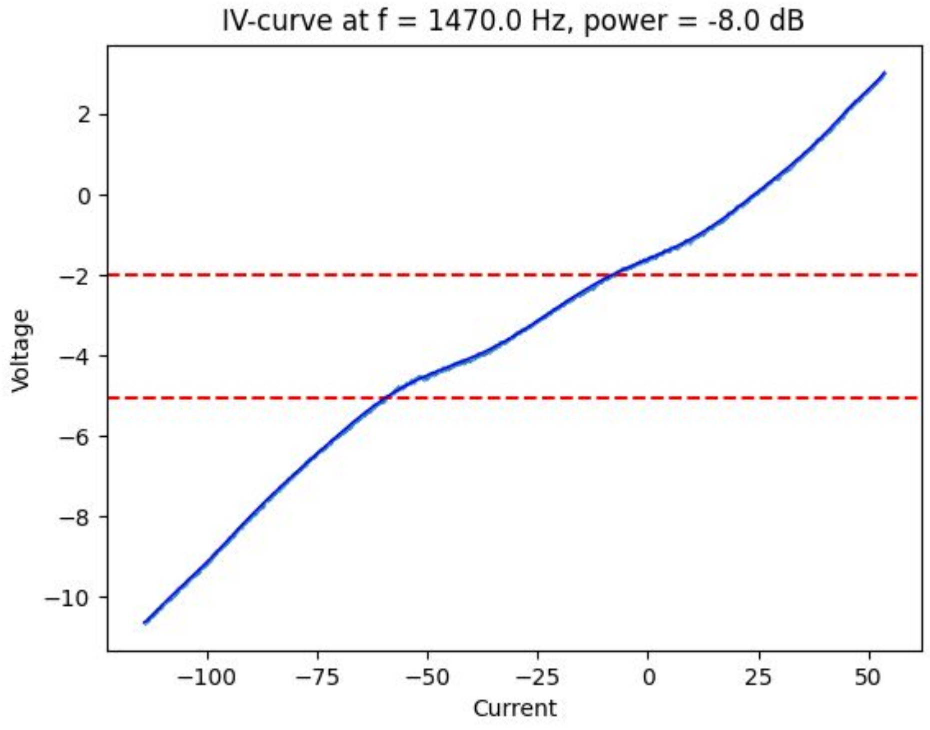
ML Method: Results - Random Samples - f = 3 to 5 GHz

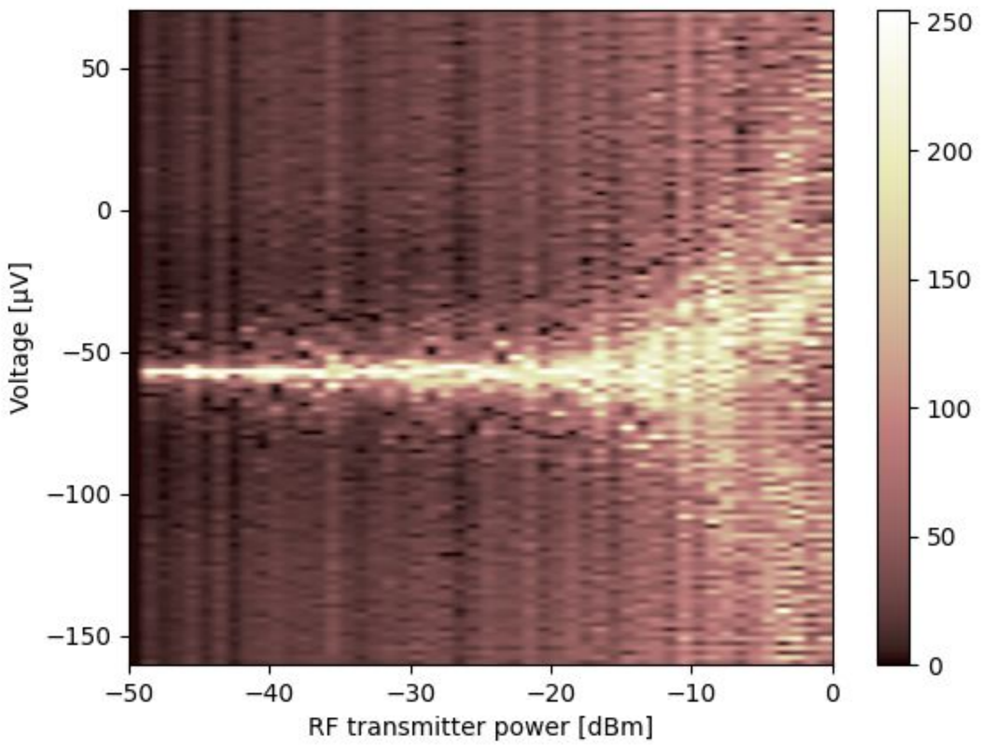




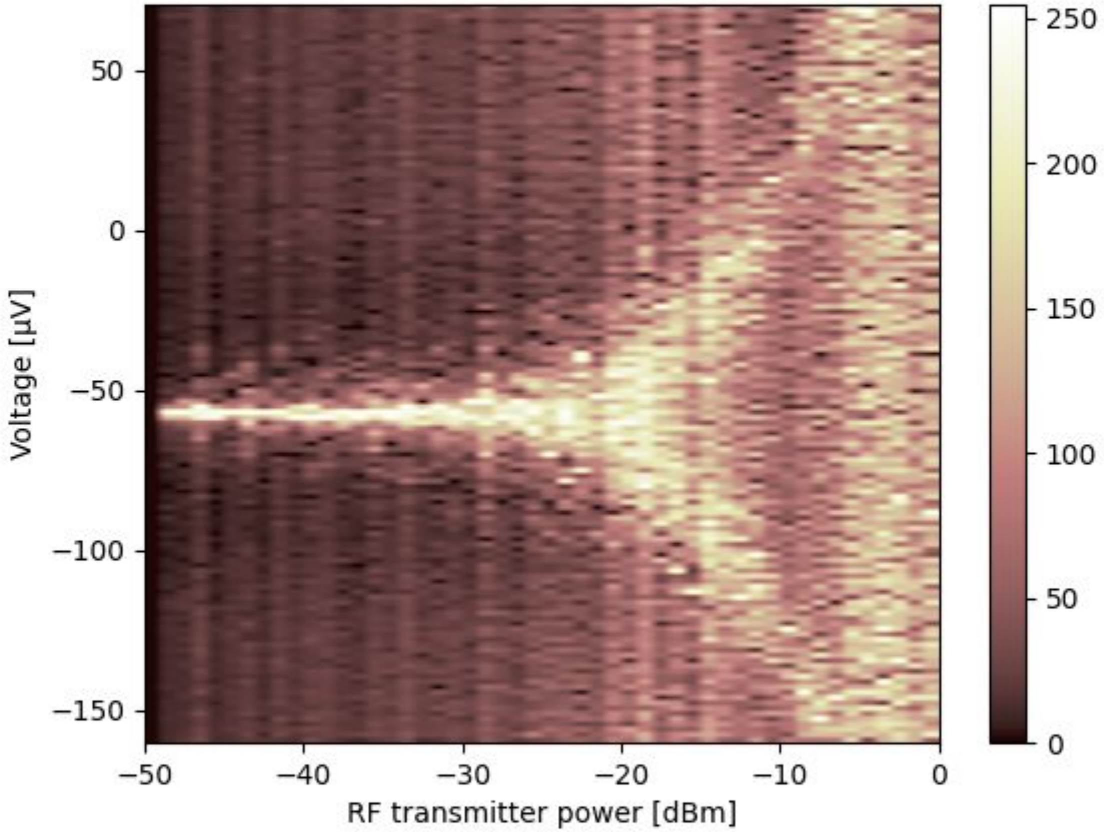
![TU Delft logo]

ML Method - The Histogram - Power Plot



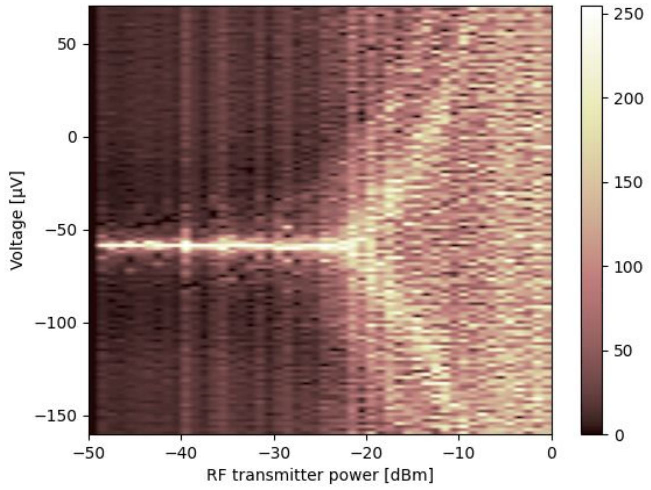


ML Method - Zigzag Pattern and Step Shift

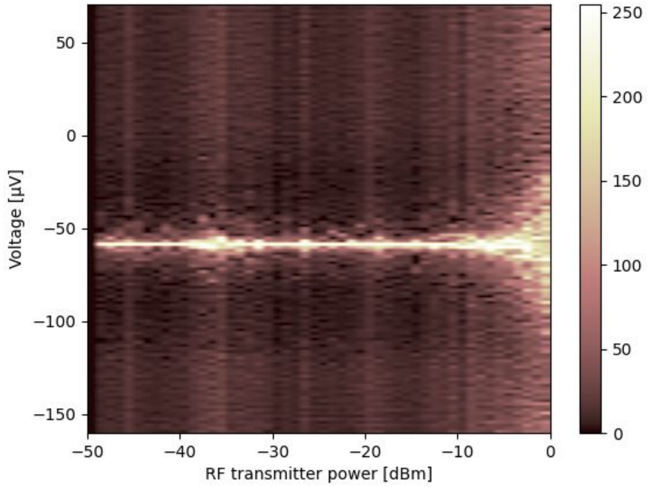


ML Method - Zigzag Pattern and Step Shift

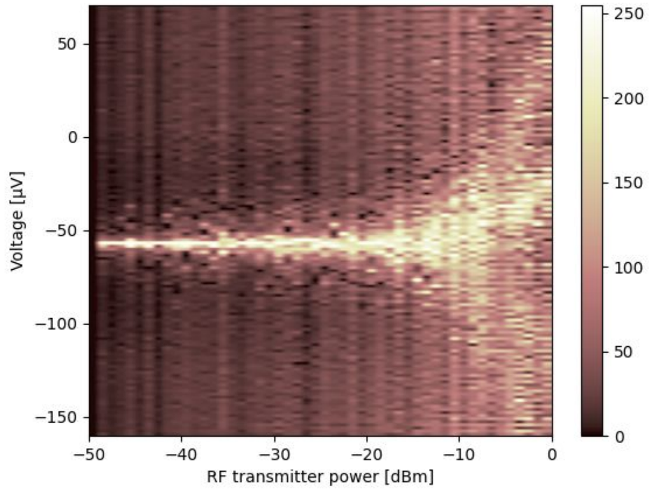
2500 MHz



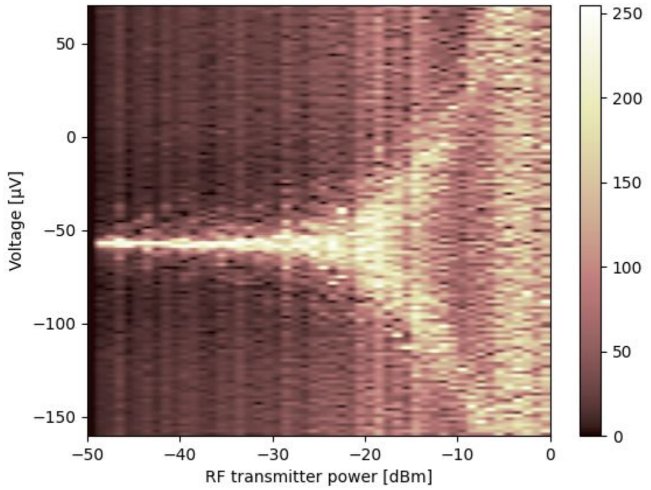
2950 MHz



3400 MHz



4500 MHz

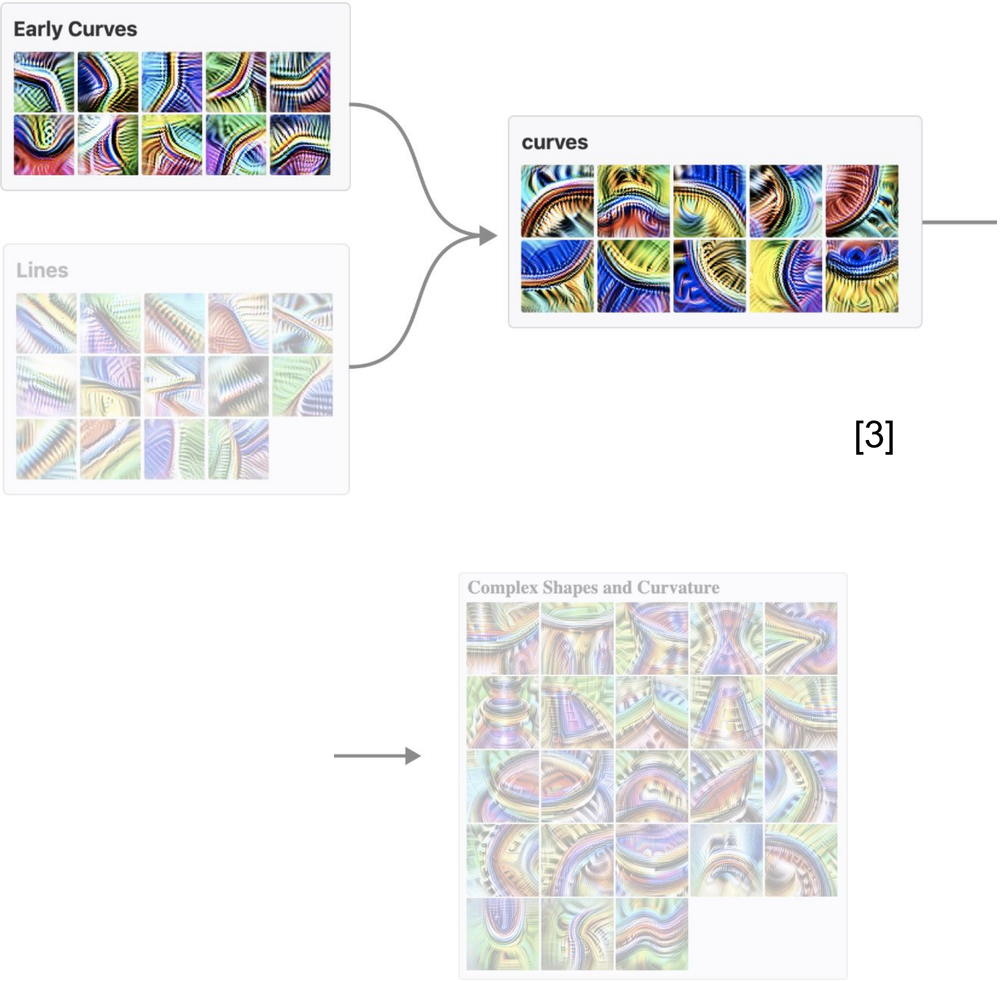


ML Method: Recommendations - 1

- Improve further on the training set
- Focus data filtering on regions of interest, i.e. regions where interesting curves were previously (or manually) found
- Implement filtering on the project PC, find the source of the bug which hindered this

ML Method: Recommendation - 2

- Replace the MLP by a Convolutional Neural Network (or any kind of image recognition)
- Input the IV curve matploblib plot
- We also select curves based on the plots
- Detection of features in the curves
  - Features such as steps, straight line segments, curved segments



[3]

Results: Steps

How to find out if Shapiro steps were found?

Criteria:

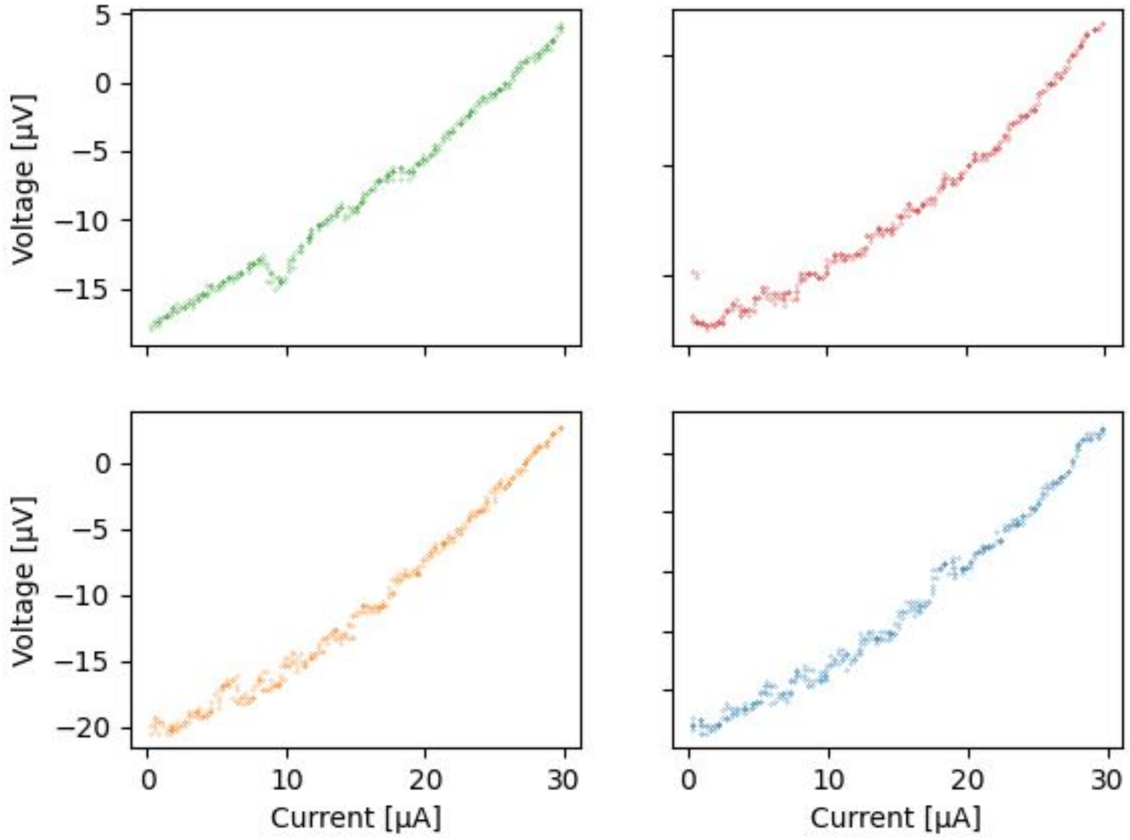
1. Step size matches theoretical value at given frequency
2. Step spacing depend linearly on frequency
3. Step spacing is independent of power

Methods:

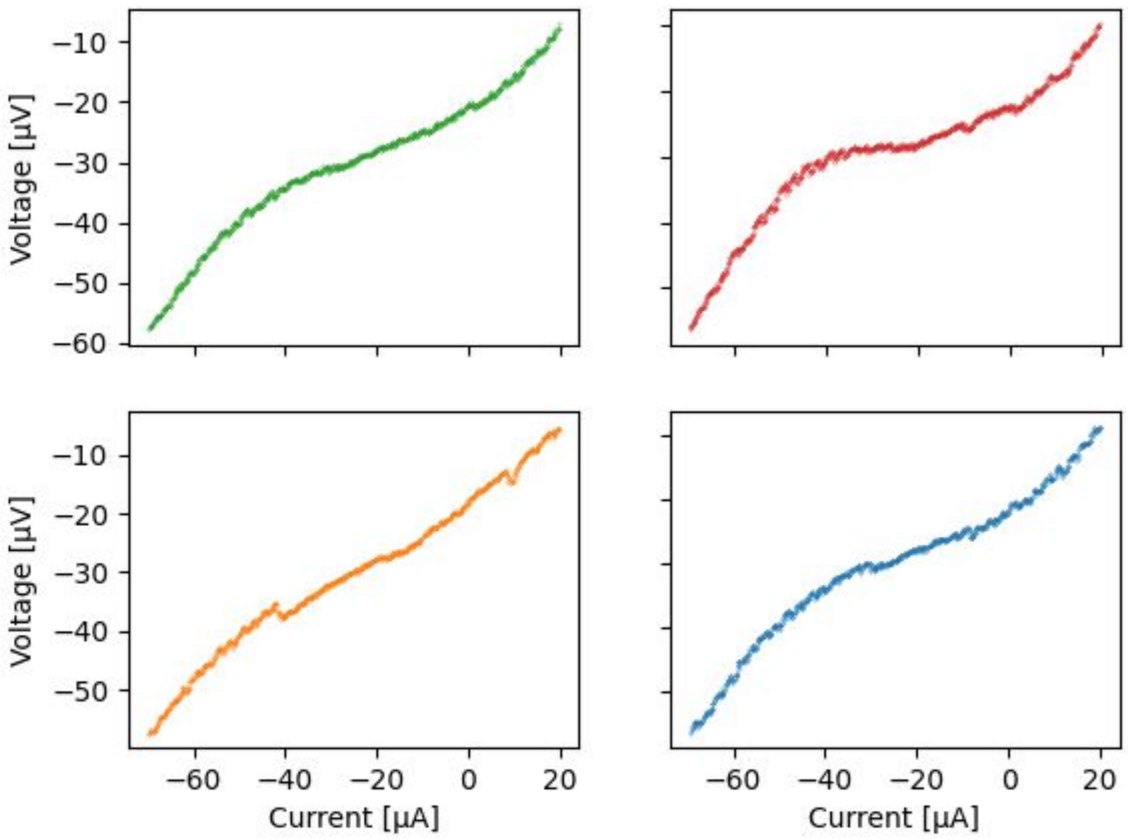
- IV curve visual inspection
  - IV filtering: 1500 > "DNN3" (raw IV DNN) > 14
- Histogram analysis
  - # counts ~ 1/slope

Results: Steps - Visual Inspection

IV curves of the right side section

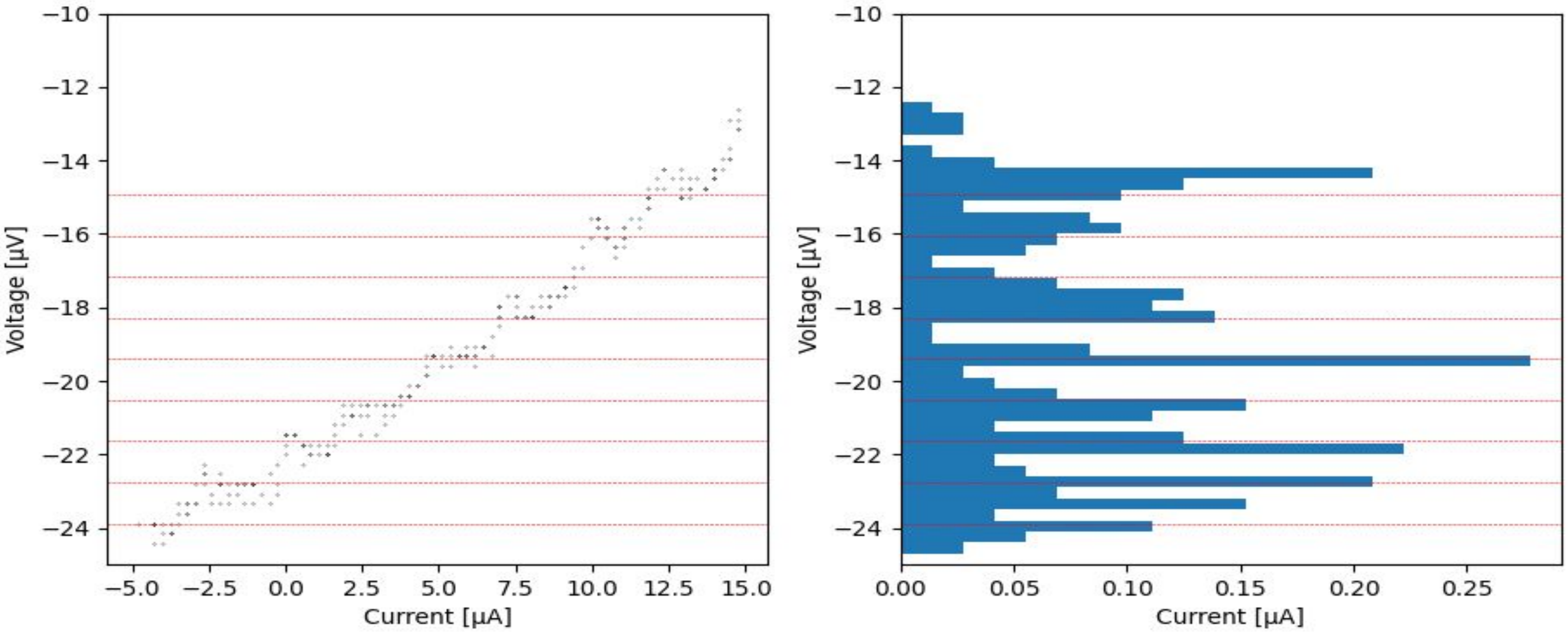


IV curves of the middle section



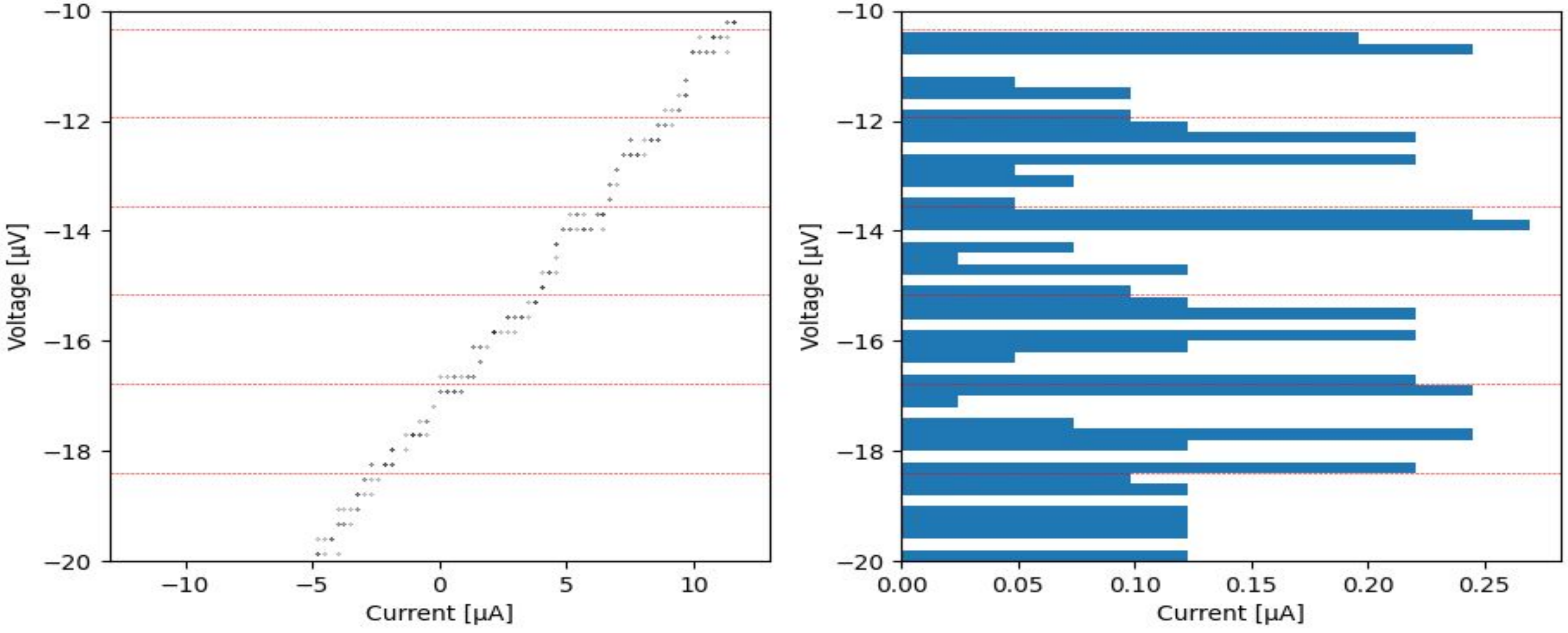
TUDelft

Results: Steps - Histogram Analysis



IV curve and histogram with Frequency: 3400 MHz and power: -23 dBm

Results: Steps - Histogram Analysis

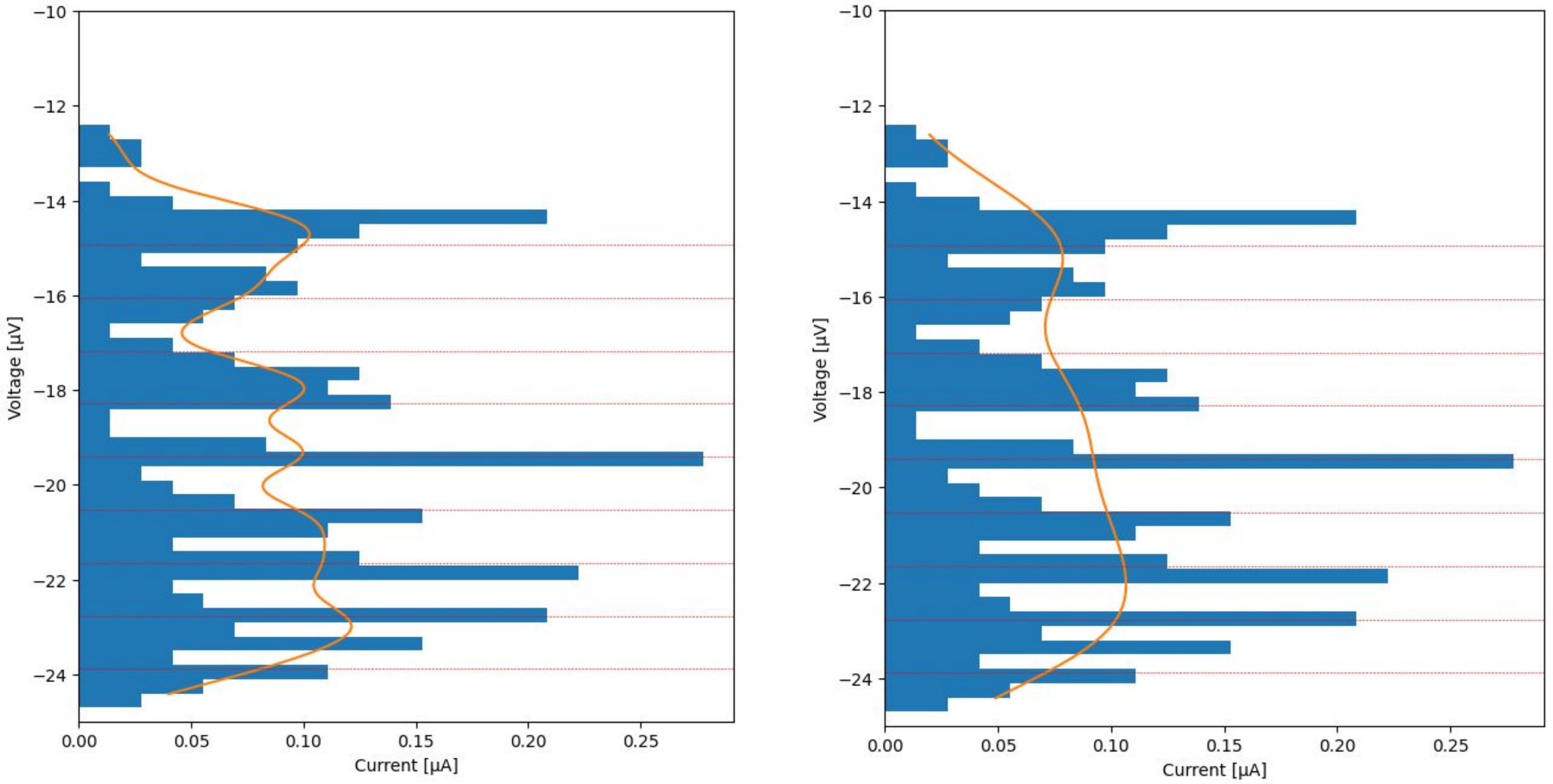


IV curve and histogram with Frequency: 4900 MHz and power: -25 dBm

Results: Steps - Recommendations

Determination of bin step size:

- SciPy / Scikit-Learn
  - KDE
  - Gaussian Mixture
- Higher resolution



Frequency: 3400MHz, Power: -23dBm

Conclusion

- Machine learning
- Different RF power and varying frequencies
- Theoretical value

Thank you for listening

Feel free to take a look at our GitHub page

References

[1] https://scikit-learn.org/stable/

[2] http://alexlenail.me/NN-SVG/index.html

[3] https://distill.pub/2020/circuits/zoom-in/#natural-science

TU Delft