natureresearch
https://doi.org/10.1038/s41565-020-00779-y

Supplementary information

A deep-learning approach to realizing functionality in nanoelectronic devices

In the format provided by the
authors and unedited

A Deep-Learning Approach to Realising Functionality in Nanoelectronic Devices

Supplementary Information

Hans-Christian Ruiz Euler, Marcus N. Boon, Jochem T. Wildeboer, Bram van de Ven, Tao Chen, Hajo Broersma, Peter A. Bobbert, Wilfred G. van der Wiel†

†Correspondence to: W.G.vanderWiel@utwente.nl

1. Comparison of different sampling methods

Supplementary Table S1 | Sampling parameters. Modulation frequencies, amplitudes and offsets used to sample the 7-dimensional voltage input space of device 1 (Boolean logic and ring classification functionality) and device 2 (2×2 pixel feature mapping functionality).

Terminal		**1**	**2**	**3**	**4**	**5**	**6**	**7**
Frequency / 0.05 (Hz)		$\sqrt{2}$	$\sqrt{3}$	$\sqrt{5}$	$\sqrt{7}$	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{19}$
Device 1	**Amplitude (V)**	0.9	0.9	0.9	0.9	0.9	0.5	0.5
	Offset (V)	-0.3	-0.3	-0.3	-0.3	-0.3	-0.2	-0.2
Device 2	**Amplitude (V)**	0.85	0.85	0.85	0.75	0.75	0.1	0.15
	Offset (V)	-0.35	-0.35	-0.35	-0.25	-0.35	0.0	0.0

We compare here the sampling density of three different sampling methods of the 7-dimensional input voltage space of the device: 1) a uniform grid sampling with the fastest running input voltage (i.e. the voltage at an input electrode that swipes the fastest through the defined grid), a one-but-fastest running input voltage, etc., 2) a sinusoidal sampling with modulation frequencies with irrational ratios given in Table S1, and 3) a triangular wave sampling with the same modulation frequencies used in 2. We obtain an estimate of the sampling density and time as follows. The 7-dimensional input voltage space is subdivided into 12 equally sized small bins per input terminal between the lowest and highest voltage, totalling 12^7 bins. For all three sampling methods, the bins containing at least one input data point are regarded as 'sampled' and the empty bins as 'unsampled'. Table S2 shows for the three different sampling methods the fraction of sampled bins in the 7-dimensional input space as 'density' and the sampling time (rounded to hours) as 'time', for different numbers of grid points per input terminal. To estimate the sampling time for the uniform grid, the fastest running voltage of the grid sampling is taken to have the same frequency as the highest modulation frequency of the sinusoidal and triangular wave sampling, which is 0.22 Hz (see Table S1). For N sample voltages per terminal, the inverse modulation frequencies of the 7 different voltage signals are proportional to N, N^2, N^3, …, N^7 and the total sampling time in seconds is then given by $N^7/(2N \times 0.22)$. Table S2 shows only small differences in the sampling density for the different sampling methods. However, the sinusoidal and triangular wave sampling methods are considerably faster than the uniform grid sampling. The triangular wave sampling becomes slightly faster than the sinusoidal sampling with growing N.

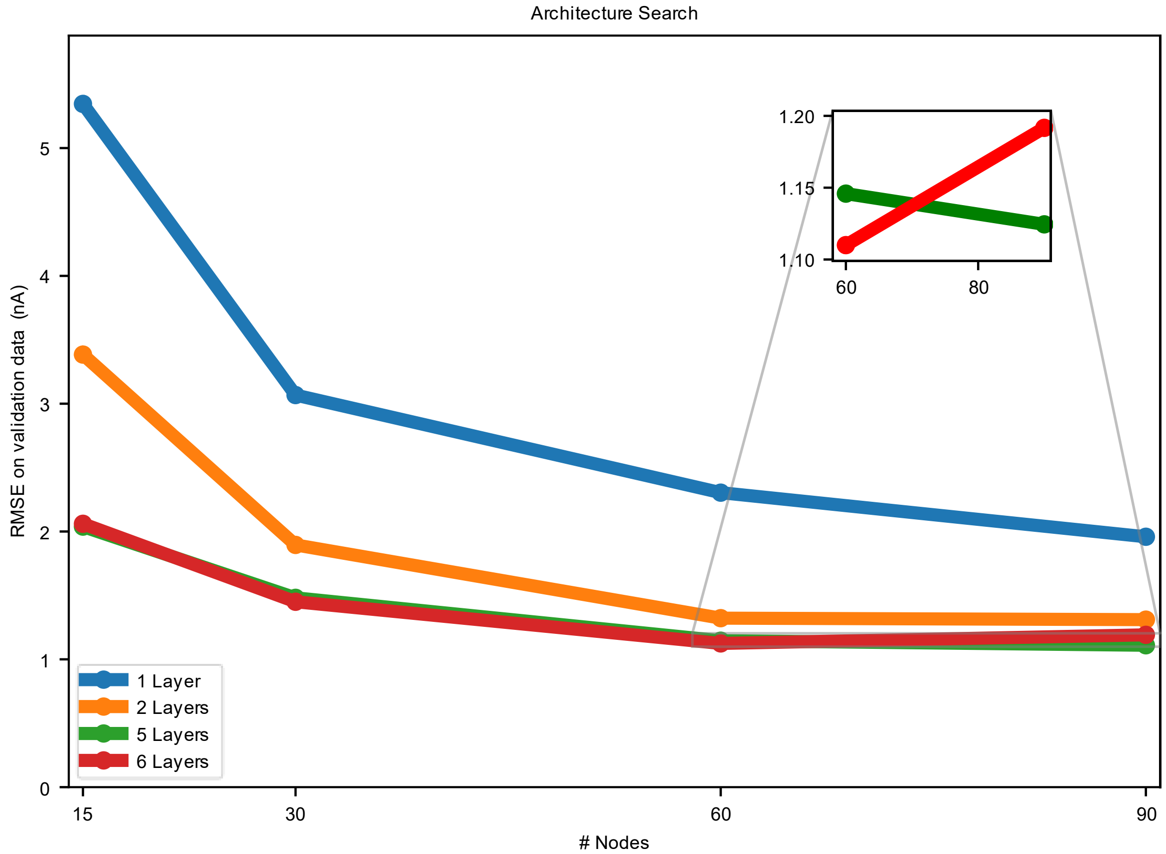
The two DNN models for the two devices in the main article are trained and tested using data sets obtained during, respectively, 48 hours and 3 hours of sampling with sinusoidal or triangular waves. The sampling in that case is somewhat denser than $N = 8$ sample voltages per terminal. Besides the advantage in data acquisition time, sampling with waves reduces the transient behaviour in the output current observed with fast voltage changes.

Supplementary Table S2 | Comparison of different sampling methods. Sampling density (sampled fraction of 12^7 equally sized bins in the 7-dimensional input voltage space) and total sampling time (rounded to hours) for three different sampling methods and different sample voltages *N* per terminal. The modulation frequencies used in the sinusoidal and triangular wave sampling are given in Table S1. A sampling frequency of 50 Hz is used in both methods.

	Uniform grid		**Sinusoidal waves**		**Triangular waves**	
N	**Density**	**Time (h)**	**Density**	**Time (h)**	**Density**	**Time (h)**
6	0.00781	30	0.00732	4	0.00738	4
7	0.0230	74	0.0231	13	0.0239	13
8	0.0585	165	0.0592	36	0.0592	33
9	0.133	336	0.135	92	0.135	78
10	0.279	631	0.279	240	0.280	175

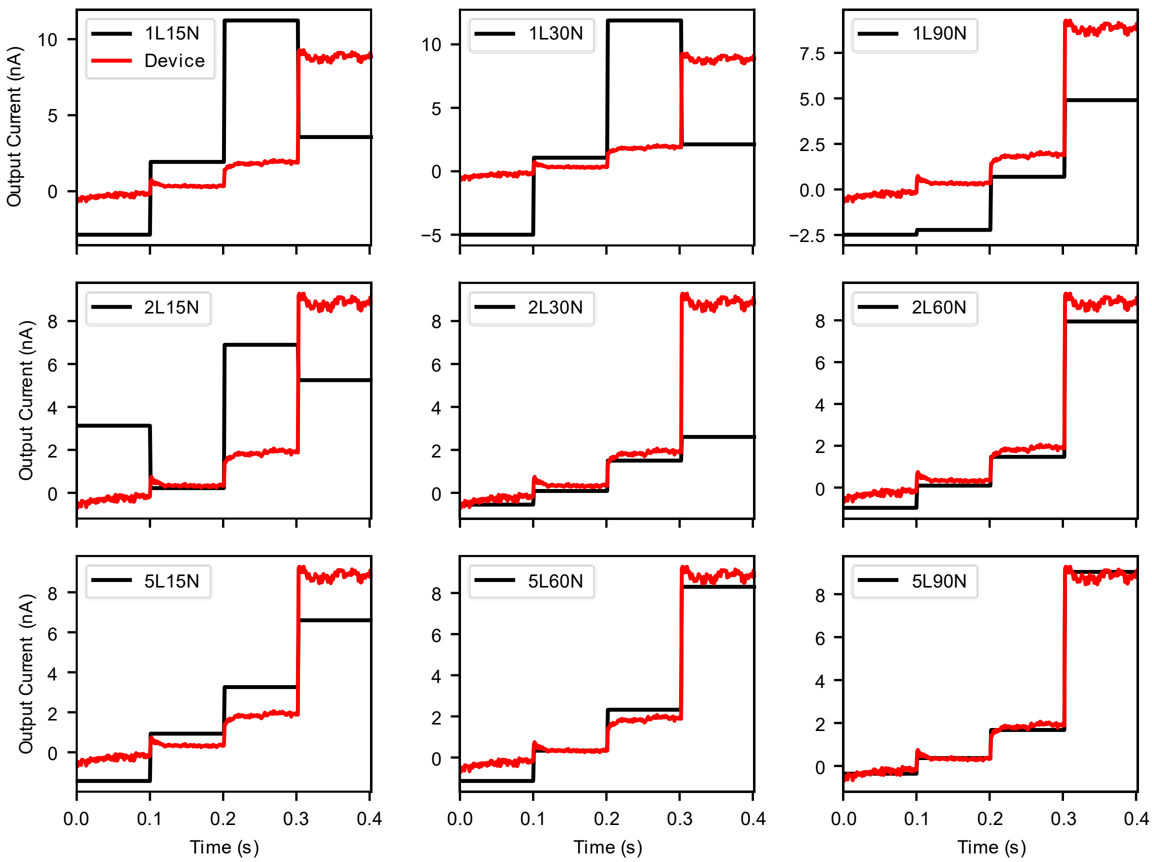
2. DNN architecture

In our optimisation of the deep neural network (DNN) architecture, we consider DNNs with different numbers of hidden layers (1, 2, 5 and 6) and nodes per layer (15, 30, 60 and 90). Figure S1 shows the root-mean-square error (RMSE) in the current predicted by the trained DNN as compared to the measured validation data on the physical device for the architectures that we considered. We observe a similar performance for DNNs with 5 and 6 hidden layers, which both show a significantly better performance than DNNs with less hidden layers. Increasing the number of nodes per layer from 60 to 90 decreases the performance of the DNN with 6 hidden layers, possibly due to a slight overfitting of the training data. The best performing DNN is the one with 5 hidden layers and 90 nodes per layer, and we therefore used this DNN in the modelling of our device.



Supplementary Figure S1 | Performance of different DNN architectures. The root-mean-square error (RMSE) in the currents predicted by the trained DNNs as compared to the measured validation data. Errors are plotted versus the number of nodes per layer in the case of 1 to 6 layers. The inset shows the difference between DNNs with 5 and 6 hidden layers for 60 and 90 nodes per layer.

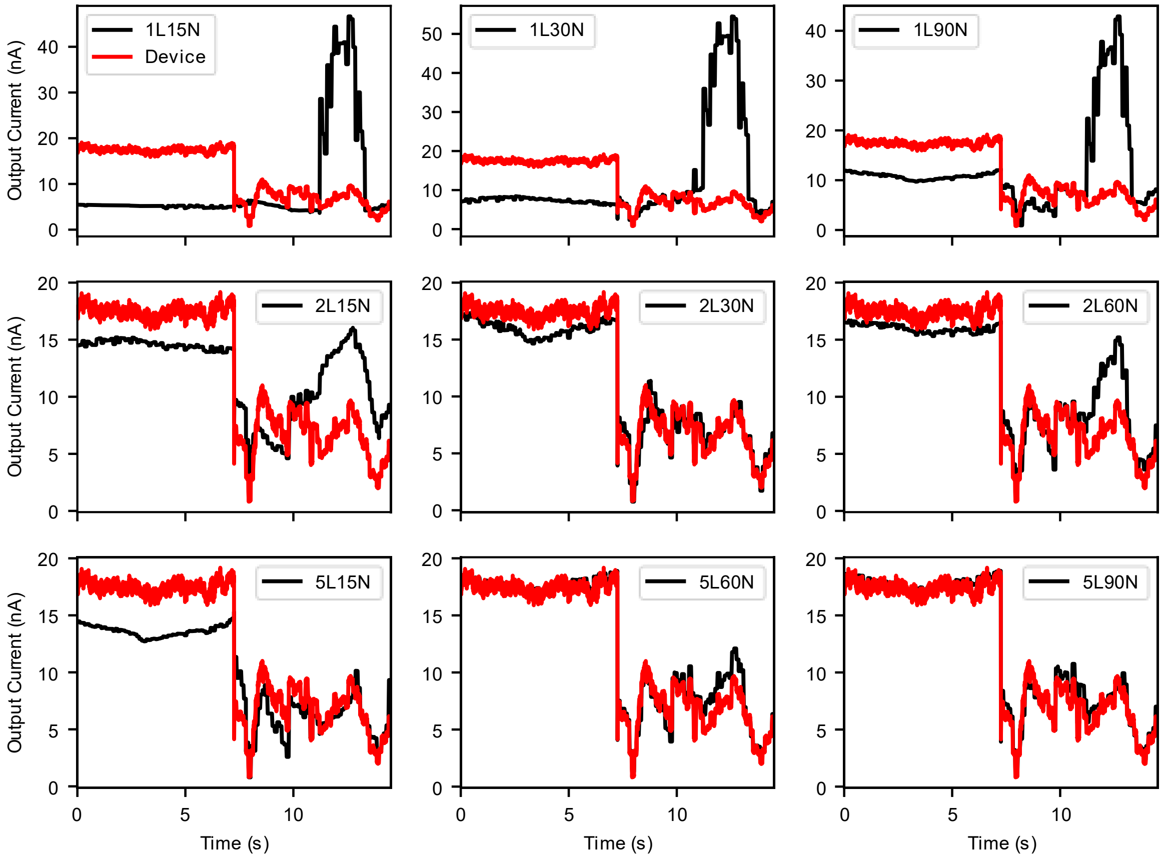
If computational efficiency is a concern, one could trade predictive accuracy (here quantified by the RMSE) for computational efficiency (here quantified by the DNN size). One could, for example, take the smallest architecture for which the RMSE levels off. In our specific case, we could choose a less computationally demanding model with 5 hidden layers and 60 nodes per layer without any significant performance loss (see Fig. S1). However, in general predictive accuracy should have priority, since failing to predict the device's behaviour correctly significantly increases the probability of false negatives and false positives when searching for functionality.



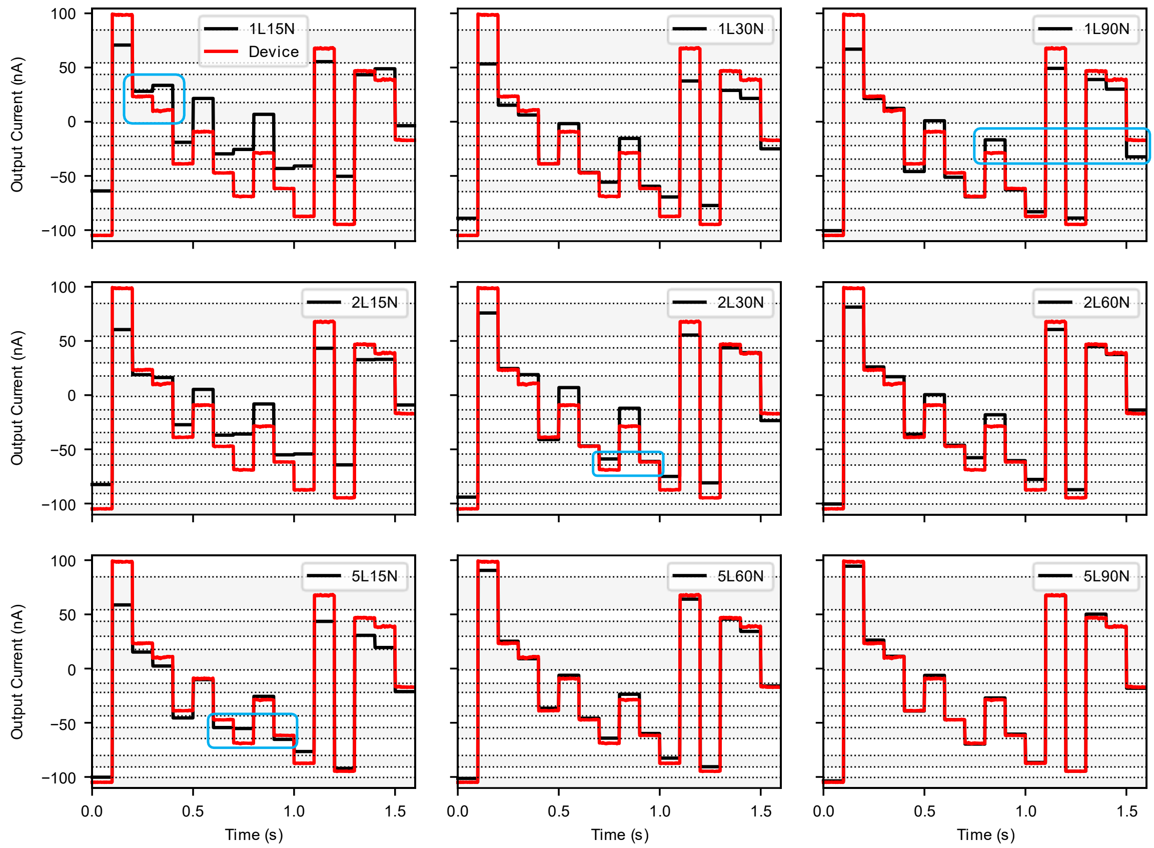
Supplementary Figure S2 | AND gate predictions of DNN models with different architectures. Black curves: predicted output from DNN models with architectures given in Fig. S1 designated as xLyN, with x layers and y nodes. Red curve: measured output current (nA) of the device with the AND gate *control voltages* as in the main text (same for all panels).

Figure S2 shows the AND gate solved by the device from the main text (red curve, always the same) compared with different current predictions using models with architectures as in Fig. S1. Although the same input and control voltages are used for all models and the device, we see large variations in the predictions. The simplest models (1L15N, 1L30N and 2L15N) fail to predict the trends in the output current, giving a false negative in the search for functionality. Hence, this control voltage configuration would be completely ignored. By chance, the model 1L90N does give the AND functionality, however, with a large error in the current prediction. We observe a similar behaviour for XNOR, where all models up to 2L15N fail to predict functionality (not shown). Similarly, large errors in the prediction of the output current could result in false positives, generating control voltage configurations that solve the task in the model but fail the validation in the device. Both tasks are solved in model 2L30N with a separation smaller than 2 nA, however, these control voltage configurations could be ignored by the searching procedure since the loss function optimizes for separation and false positive solutions with larger separations could be prioritised. All other tasks are solved with all models, so those voltage configurations happened to be robust to model errors due to the large differences in the “high” and “low” levels.

Naturally, if tasks require higher accuracy, the performance of the model becomes more relevant. Figure S3 shows the prediction of the output current given the same control voltages that solve the ring classification task as in the main text. We observe again a failure of the simpler models (up to 2L15N) to predict functionality with these control voltages. Coincidentally, the model 2L30N has low error for these inputs, predicting functionality with a relatively large gap. However, the models 2L60N and 5L15N fail to predict the correct separation of the classes, increasing the probability that this solution would be ignored if false positives exist with larger separation between classes, i.e. only the models with the lowest RMSE correctly and reliably predict the functionality with large separability between classes.



Supplementary Figure S3 | Model comparisons for the ring classification. Same representation as in Fig. S2. For this task, more complex models are required to correctly predict the control voltage configuration.

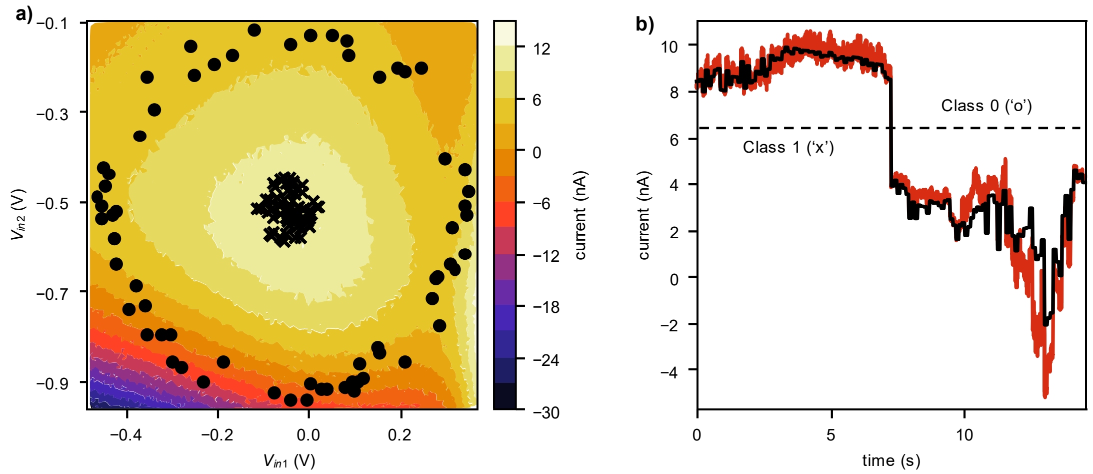


Supplementary Figure S4 | Model comparison for the 2×2 pixel feature mapping task. Same representation as in Fig. S2, with the addition that the red curve represents the mean output over 10 measurements. Even the mid-sized models 2L30N and 5L15N fail in the functionality given the control voltage configuration obtained in the main text. The error in the models creates situations (examples of these are indicated by blue windows) where there is no meaningful separation predicted, which makes the voltage configuration a false negative, or there is an output current from the device that transposes the order of the predicted feature mapping, see e.g. 1L90N.

The need for more complex models in tasks with even higher accuracy requirements is shown in Figure S4. Here, we compare the predicted outputs for the feature mapping task solved in the main text. The models used for the prediction of the output are those used to estimate the RMSE curves in Figure S1. We observe that in this example even mid-sized models like 2L30N and 5L15N fail (blue windows). There are two non-exclusive ways in which a model can fail in this task. First, if there is an incorrectly predicted separation of the features, the search will result in a false negative for this specific control voltage configuration. Second, even if the predicted features are well separated, the order of the features can be exchanged, for instance as in model 1L90N. Although technically we would accept this solution because we only optimise for separability, the order of the features in the output would be permuted. Hence, the assignment of the predicted output after training a naive Bayes classifier would not correspond to the correct feature, making the functionality search rely more heavily on the naive Bayes classifier and the validation of the functionality obtained a posteriori.

3. Alternative ring classification functionality.

Figure 5 in the main paper shows a ring classification functionality using V_4 and V_5 of the device as data inputs. To demonstrate that other electrode choices are also possible, Fig. S5 shows an alternative ring classification functionality using V_6 and V_7 as inputs. The RMSE is in this case 0.83 nA. The control voltages for this alternative ring classification functionality are given in Table S3 and the scaling factor and offsets of the two input voltages in Table S4.



Supplementary Figure S5 | Alternative ring classification functionality. Ring functionality as in Fig. 5 of the main text, but now with V_6 and V_7 from Fig. 3b in the main paper as data input voltages $V_{\mathrm{in},1}$ and $V_{\mathrm{in},2}$. **(a)** 66 outer points (class '0', discs) and 66 inner points (class '1', crosses) to be classified, superimposed on a heat map of the device's output current. **(b)** Prediction of the current by the trained DNN (black curve) and verification measurement (red curve). The dash black line indicates a possible threshold value for the class assignment.

Supplementary Table S3 | Control voltages for the alternative ring classification. Control voltages predicted by the DNN, applied to the device to verify the alternative ring classification in Fig. S5.

Terminal	**1**	**2**	**3**	**4**	**5**
Control voltage (V)	0.10	-1.20	0.40	0.48	0.08

Supplementary Table S4 | Input scaling and offsets for the alternative ring classification. Scaling factor and offsets of the input voltages predicted by the DNN, applied to the device to verify the alternative ring classification in Fig. S2.

Terminal	**6**	**7**
Scaling voltage (V)	0.42	0.42
Offset voltage (V)	-0.046	-0.517

4. Control voltage parameters of optimised functionalities in main text.

Supplementary Table S5 | Control voltages for Boolean logic. Control voltages (in V) predicted by the DNN, applied to the device to verify the Boolean logic gates in Figs. 3 and 4.

Terminal	**1**	**4**	**5**	**6**	**7**
AND	-1.19	0.15	-0.02	0.21	-0.36
NAND	-0.48	-1.06	0.23	-0.60	0.30
OR	-0.30	-1.20	-1.07	-0.27	-0.33
NOR	-1.14	-0.18	-0.05	0.30	0.28
XOR	-0.62	-0.17	-0.78	-0.63	0.26
XNOR	-1.20	0.11	-0.35	0.30	-0.27

Supplementary Table S6 | Control voltages for the ring classification. Control voltages predicted by the DNN, applied to the device to verify the ring classification in Fig. 5.

Terminal	1	2	3	6	7
Control voltage (V)	0.28	0.60	-1.02	-0.02	-0.69

Supplementary Table S7 | Affine transformation parameters for the ring classification. Scaling voltage and offsets of the input voltages predicted by the DNN, applied to the device to verify the ring classification in Fig. 5.

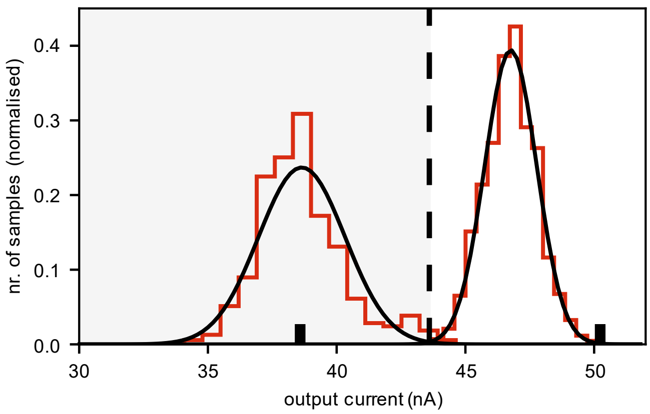
Terminal	4	5
Scaling voltage (V)	-0.80	-0.80
Offset voltage (V)	-0.167	0.309

Supplementary Table S8 | Control voltages for the pixel feature mapping. Control voltages predicted by the DNN, applied to the device to verify the pixel feature mapping in Fig. 6.

Terminal	1	6	7
Control voltage (V)	-1.20	0.10	-0.10

Supplementary Table S9 | Affine transformation parameters for the pixel feature mapping. Scaling voltages and offsets of the input voltages predicted by the DNN, applied to the device to verify the pixel feature mapping in Fig. 6.

Terminal	2	3	4	5
Scaling voltage (V)	0.85	0.45	-0.07	0.41
Offset voltage (V)	-0.35	-0.85	0.46	-0.46



Supplementary Figure S6 | Decision boundaries for the pixel feature mapping. Zoom-in to the current region 30-52 nA in the right panel of Fig. 6b. The red lines show the current histograms of two-pixel features (0010 and 1000), with Gaussian fits shown by the black lines. The black dashed line shows the decision boundary current obtained with a naive Bayes classifier. The DNN predictions of the current for the two features are shown as thick black marks.

Supplementary Table S10 | Decision boundary currents for the pixel feature mapping. Currents defining the decision between different 2×2 pixel features in Fig. 6b obtained with a naive Bayes classifier.

Pixel feature decision	**Decision boundary current (nA)**
1111 – 1011	-100.41
1011 – 0111	-90.74
0111 – 0011	-80.14
0011 – 1101	-64.51
1101 – 1001	-54.37
1001 – 0101	-43.59
0101 – 1110	-34.48
1110 – 0001	-22.06
0001 – 1010	-13.56
1010 – 0110	-1.19

0110 – 1100	17.68
1100 – 0010	29.81
0010 – 1000	43.59
1000 – 0100	54.34
0100 – 0000	84.51