A hardwired machine learning processing engine fabricated with submicron metal-oxide thin-film transistors on a flexible substrate

DOI:
10.1038/s41928-020-0437-5

Document Version
Accepted author manuscript

A hardwired machine learning processing engine fabricated with sub-micron metal-oxide thin-film transistors on a flexible substrate

Emre Ozer[*], Jedrzej Kufel[*], James Myers[*], John Biggs[*], Gavin Brown[*^],

Anjit Rana[‡], Antony Sou[‡], Catherine Ramsdale[‡] and Scott White[‡]

[*]Arm, [‡]PragmatIC and [^]University of Manchester

Corresponding Author: Emre Ozer (emre.ozer@arm.com)

Abstract: *Flexible electronics can create lightweight, conformable components that could be integrated into smart systems for applications in healthcare, wearable devices and the Internet of Things. Such integrated smart systems will require a flexible processing engine to address their computational needs. However, the flexible processors demonstrated so far are typically fabricated using low-temperature poly-silicon thin-film transistor (TFT) technology, which has a high manufacturing cost, and the processors that have been created with low-cost metal-oxide TFT technology have limited computational capabilities. Here, we report a processing engine that is fabricated with a commercial 0.8 µm metal-oxide TFT technology. We develop a resource-efficient machine learning (ML) algorithm (termed univariate Bayes feature voting classifier) and demonstrate its implementation with hardwired parameters as a flexible processing engine for an odour recognition application. Our flexible processing engine contains around 1,000 logic gates and has a gate density per area that is 20–45 times higher than other digital integrated circuits built with metal-oxide TFTs.*

Flexible electronic devices are built on substrates such as paper, plastic and metal foil, and use active materials such as organics, metal oxides and amorphous silicon. They offer a number of advantages over traditional silicon devices, including thinness, conformability and low manufacturing costs, and various commercial systems are already available, including organic light emitting diodes, flexible displays and organic photovoltaics. The integration of different flexible components — for instance, printed sensors, organic displays, printed batteries, energy harvesters, memories, antennas, and near field communication or radio frequency identification (RFID) chips — could lead to innovative products such as flexible integrated smart systems [1] for logistics, fast moving consumer goods (FMCG), healthcare, wearables, and the Internet of Things

(IoT) [2]. However, to address the computational requirements of such integrated systems, a flexible processing engine, which operates as a central processing unit (CPU) or a domain-specific processing engine, is required.

CPUs are general-purpose (i.e. programmable) processors that can be used for multiple applications. As a result, when an application is run, parts of the hardware inside a general-purpose processor remain unused, and become an overhead (mainly in terms of area and power consumption) for the application running on it. This observation — called the Turing tax [3] — defines the compromise of universal computing. In contrast, domain-specific processing engines [4][5][6] are specialised hardware designed for a class of applications within a single domain, such as graphics, signal processing, machine learning, augmented/virtual reality, and security. They make the computation more efficient in terms of energy consumption, area, cost, and performance.

One approach is to integrate conventional silicon-based CPUs onto flexible substrates as processing engines. This is called hybrid integration [7][8][9] in which the silicon wafer is thinned and dies from the wafer are integrated onto a flexible substrate. However, this approach requires an expensive packaging process because the thinning process makes silicon more fragile. Thus, it is not a viable long-term solution for high-volume, low-cost, flexible integrated smart systems. Alternatively, a processing engine (either general-purpose or domain-specific) can be built exclusively with flexible electronic fabrication techniques, an approach we term a natively-flexible processing engine (NFPE).

Thin-film transistors (TFTs) can be fabricated on insulating substrates, such as glass or flexible polymeric substrates, and have a lower processing cost than metal–oxide–semiconductor field-effect transistors (MOSFETs) on silicon substrates [2][10]. A flexible CPU has, for example, been developed using a transfer process from a glass substrate onto a flexible one [11]. Furthermore, a flexible 8-bit CPU based on the integration of a flexible RFID controller and an antenna has been reported [12], as well as an asynchronous flexible 8-bit CPU [13] and an 8-bit ultra-high frequency radio frequency CPU (UHF RFCPU) on a flexible substrate [14]. However, all of these flexible 8-bit CPUs were developed using low-temperature poly-silicon (LTPS) TFT technology,

which has a high manufacturing cost and poor lateral scalability (limiting the complexity of the integrated circuits). More recently, a 16-bit RISC-V processor [15] built from complementary carbon nanotubes transistors was developed, though this used a conventional wafer rather than a flexible substrate.

Metal-oxide TFTs [16] are, in contrast, low-cost and can also be scaled down to the much smaller geometries required for large scale integration [17]. To date, only basic 8-bit arithmetic logic units (ALU; part of the CPU) fabricated with metal-oxide TFTs on a flexible substrate [18][19] have been demonstrated; these are proof-of-concept prototypes with limited computational capabilities. To develop an NFPE that can perform meaningful computations, a sufficient number of metal-oxide TFTs needs to be integrated.

In this Article, we report a domain-specific NFPE that is fabricated using a 0.8 µm metal-oxide TFT technology and implements a machine learning (ML) algorithm. We develop an algorithm, termed Univariate Bayes Feature Voting Classifier (UB-FVC), and implement it in hardware for an odour classification application (e-nose). The UB-FVC algorithm achieves a prediction accuracy of 90%, and its implementation as a NFPE contains 1,024 logic gates, which has a higher gate density (by 20–45 times) compared to other flexible processing circuits based on metal-oxide TFT technology.

Table 1 Process technology parameters. The table shows the FlexLogIC® fabrication technology information and lists the statistical variations of TFT parameters.

“Technology information and parameters”	**“Values/Types”**
Semiconductor material in metal-oxide TFTs	Indium-Gallium-Zinc Oxide (IGZO)
Flexible substrate	Polyimide
Channel length (µm)	0.8
Minimum supply voltage (V)	3
Wafer diameter (mm)	200
Total thickness (µm)	< 15
Number of material layers	13
Number of routable metal layers	4
TFT V_{th} (V)	Mean:0.685, St dev:0.057
TFT sub-threshold swing (V/dec)	Mean:0.119, St dev:0.017

TFT linear on-current (μA)	Mean: 2.23, St dev:0.25
TFT saturation on-current (μA)	Mean: 32.7, St dev:4.3
TFT hysteresis (V)	Mean: 0.126, St dev:0.023

FlexIC technology

Our NFPE is based on a flexible integrated circuit (flexIC) fabricated using a commercial 'fab-in-a-box' manufacturing line, FlexLogIC® [20]. The process uses an n-type metal-oxide TFT technology based on indium-gallium-zinc-oxide (IGZO) and generates the flexIC design on a 200 mm diameter wafer by running several sequences of material deposition, patterning and etching. The details of the fabrication methodology can be found in the *Methods* section.

The IGZO TFT circuits are made using conventional semiconductor processing equipment configured to produce devices on a flexible substrate - polyimide with less than 15 μm thickness - that can be bent to a radius of curvature of 5mm without damage to circuitry. The TFTs have a channel length of 0.8μm, and a minimum supply voltage of 3V. Process parameters and statistical variations of TFT parameters are summarised in **Table 1**.

Development of hardwired ML NFPE

The specific domain of our NFPE is ML where the training phase of an ML algorithm is performed offline. After training, the learned parameters remain fixed or hardwired in the inference phase so that the inference phase of an ML algorithm can be efficiently implemented in hardware. We develop an NFPE implementing an ML inference algorithm with hardwired parameters for an odour classification in sweat application that uses a flexible e-nose sensor array consisting of multiple organic field-effect transistors (OFETs) [21].

The e-nose sensor array model used in the Article is based on OFET sensors similar to the flexible OFET sensor reported previously in [22] [23]. As shown in **Fig. 1a**, each OFET sensor has an organic semiconductor between the source and drain electrodes that is sensitive and selective to volatile organic compounds (VOCs) in odour, and generates a current when exposed to odour. An array of OFET sensors each of which

has a different organic semiconductor material and/or geometry will respond to a number of VOCs. Each sensor is not tuned to detect a specific VOC, so all sensors can respond to VOCs in odour in a different manner because of their different sensing material and geometry. The combined behaviour of the sensor array makes the difference to separate one odour type from another.

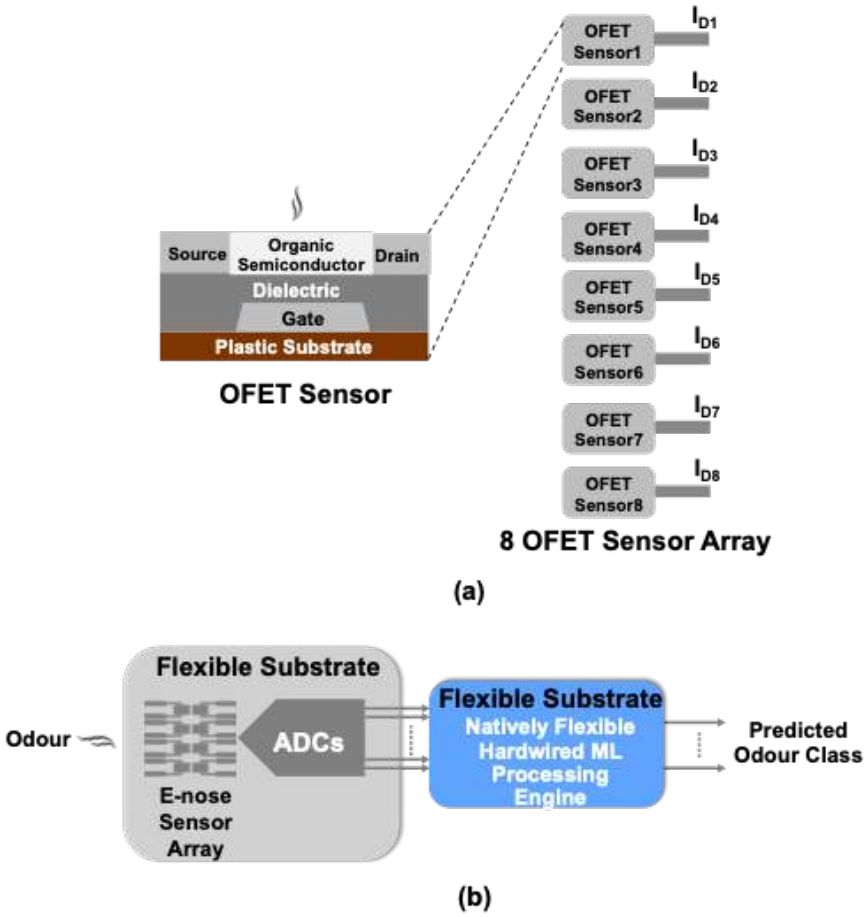


Fig. 1 OFET sensors and system architecture of the flexible smart system. a) A single OFET sensor and an e-nose sensor array consisting of eight OFET sensors. **b)** System architecture of the flexible smart system consisting of the e-nose sensor array with ADCs on a flexible substrate and the natively flexible hardwired ML processing engine on a flexible substrate

Each sensor generates an output current that will be converted into digital data by an analog-to-digital converter (ADC), which will then be processed by the NFPE in order to classify the odour as shown in **Fig. 1b**. The focus of this Article is the design, implementation, fabrication and test of the NFPE. The NFPE development methodology is generic enough to be adapted to other odour-based applications such as food packaging, wound dressing, room air quality detection etc. Each application has

different input, output and performance requirements, and the best performing ML algorithm can vary from application to application but the methodology to develop it remains the same.

A number of standard ML algorithms will need to be explored in order to meet the prediction accuracy requirement of the application. Once the best performing ML algorithm is found, a thorough analysis is required to assess the hardware implementation constraints of the ML algorithm. This is because the ML algorithm will be implemented as a domain-specific processing engine using the flexible electronics fabrication technology that is not as mature as the conventional silicon technology in terms of large-scale integration. If the hardware of the algorithm cannot reasonably be fabricated, then either the hardware design needs to be further optimised to reduce its complexity or the ML algorithm needs to be modified to have simpler hardware implementation given the fabrication constraints.

In this Article, we focus on the application of “odour classification in sweat” for which a 90% prediction accuracy is acceptable. In order to develop an ML hardware to classify odour in this application, we investigate a number of standard ML algorithms such as Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Decision Tree (DT), k-Nearest Neighbour (k-NN) and Gaussian Naïve Bayes (GNB). We run each ML model on the e-nose sensor array data generated by the OFET sensor model. There are eight e-nose sensors in the sensor array, and the ML engine will classify their response into five different odour classes at the output. The full precision of the sensor data is 9 bits but we also quantise the sensor data from the full precision down to 2 bits using dynamic data range scaling to understand the effects of using fewer data bits on the performance of the ML algorithms. Quantised data are used both in training and inference stages for all models.

The prediction accuracy results are shown in **Fig. 2a**. When the sensor data are in full precision, the best performing ML algorithm is GNB with a prediction accuracy of 92%. We also observe that quantising the sensor output down to 5 bits does not impact the classification accuracy for GNB and other ML models. This implies that a 5-bit ADC conversion would be sufficient for the ML inference hardware running the ML algorithm.

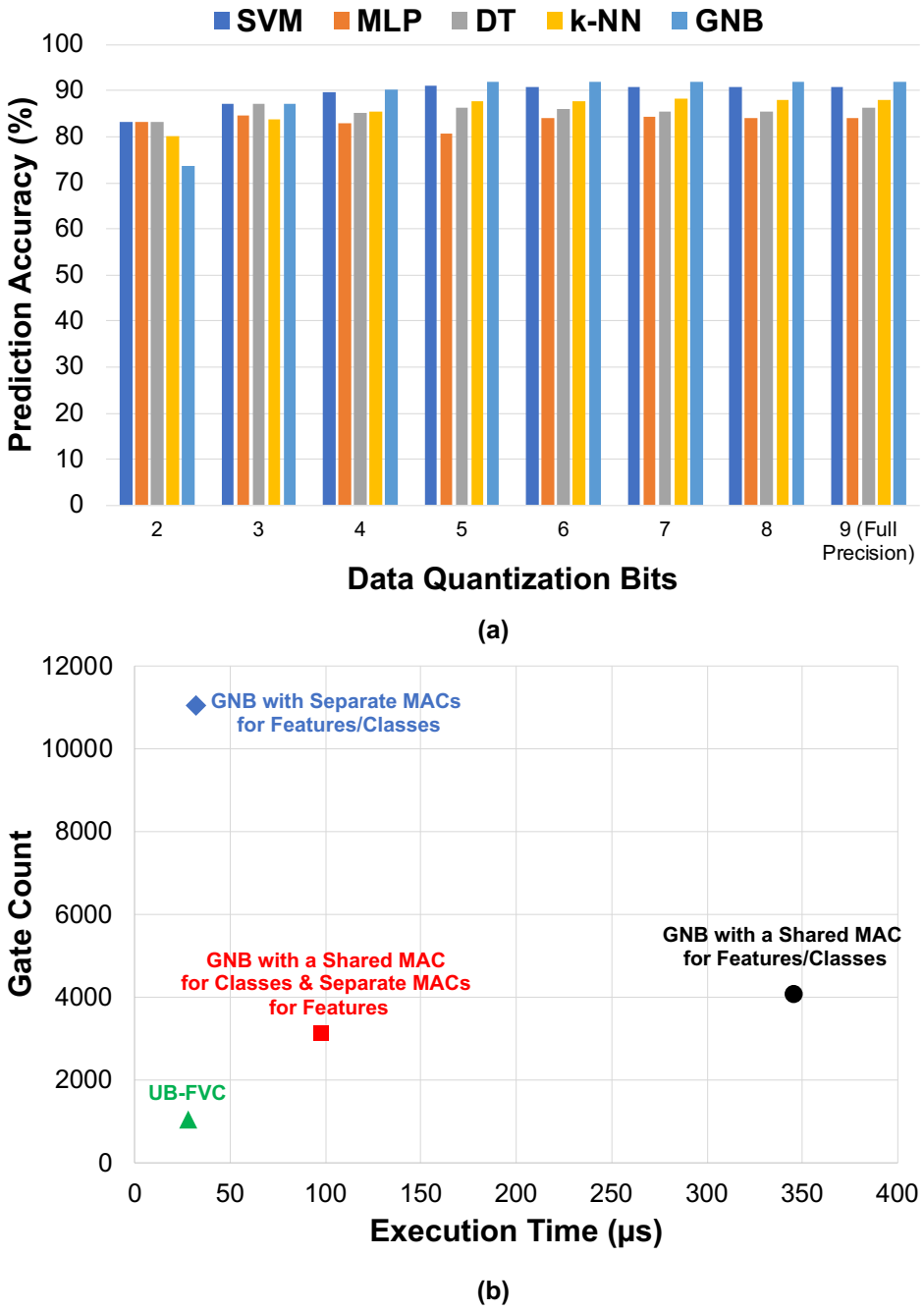


Fig. 2 Design Space Exploration with Various ML Algorithms. a) Prediction accuracies are shown for various standard ML algorithms on the odour classification application varying data quantisation levels from 2 bits to 9 bits (full precision). The ML training and performance evaluation methodology follows the standard ML practice: The dataset is split into training and test datasets. Then, the ML algorithms are trained offline using the training datasets. Once the training is complete, the performance of the ML algorithms with learned parameters are evaluated with the test datasets. We use a 5-fold cross-validation methodology to avoid overfitting. Classification prediction accuracy is used as a metric that is defined as how accurate the prediction is with respect to the ground truth. No visible difference is observed between 5-bit and full precision data representations. The best performing ML algorithm is GNB with a prediction accuracy of 92%. **b)** The 5-bit GNB design variants are compared in terms of gate count and execution time. The three GNB variants are created by either sharing or duplicating the multiply-accumulate (MAC)

units for features (i.e. sensor inputs) and classes (i.e. outputs). Sharing a MAC among classes and features reduces the number of gates while increasing the execution time. On the other hand, separate MACs will increase the number of gates while improving the execution time by doing computations in parallel. The smallest GNB implementation is the one with a shared MAC for classes and separate MACs for features and is comprised of over 3000 gates.

We pick GNB as the best performing ML algorithm among all ML algorithms. Then, we design and implement the GNB inference algorithm as a NFPE using the generic methodology described in our earlier work [24]. **Fig. 2b** compares three variants of the GNB hardware using 5-bit data quantisation in terms of total gate count and execution time. The smallest GNB hardware implementation has over 3000 gates.

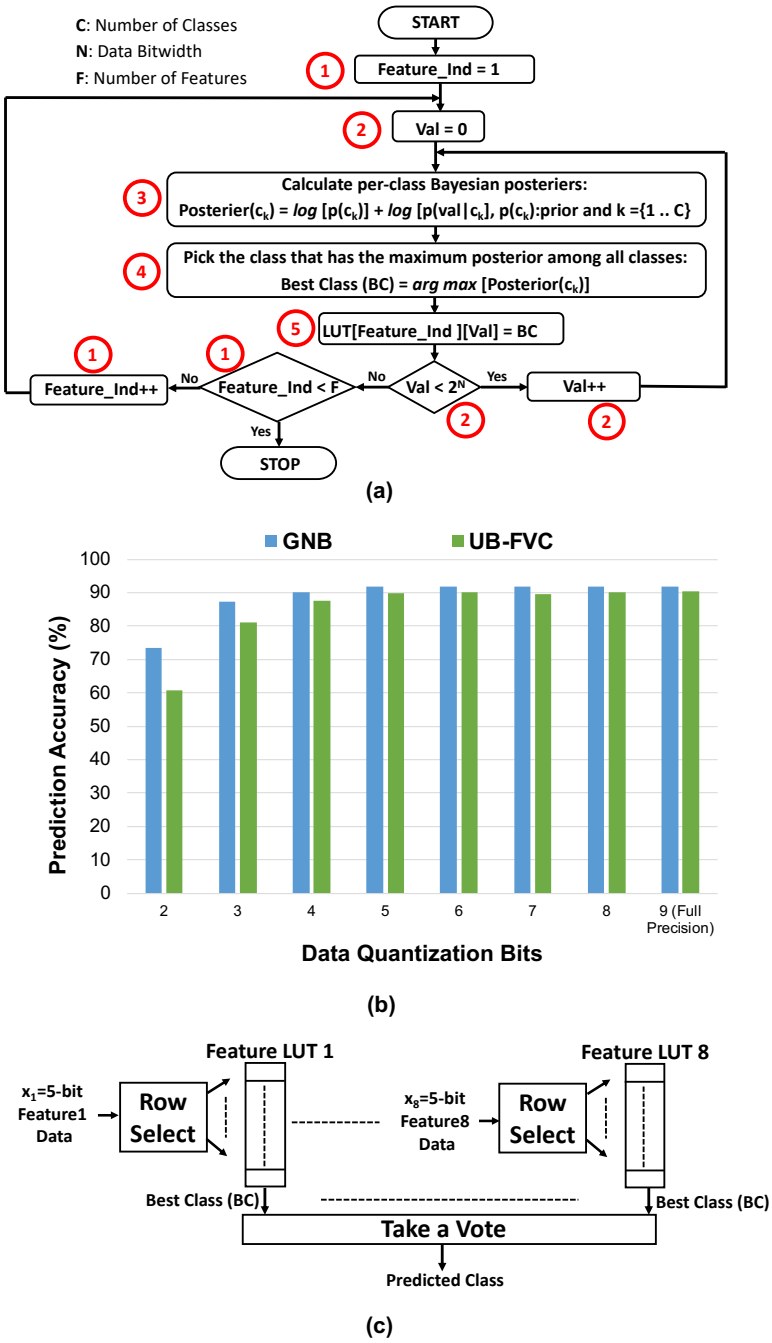


Fig. 3 Univariate Bayes feature voting classifier (UB-FVC). a) The training algorithm of UB-FVC computes the class posterior probabilities for each feature (i.e. sensor) independently **(Step 3)**, and picks the best class (BC) for the feature **(Step 4)**. Because feature values are quantised values from 0 to 2^n-1 where n is the data bitwidth, the algorithm computes the BC for each value of a feature **(Step 2)** and stores them in a look-up table (LUT) per feature and value **(Step 5)**. These steps are repeated for all features **(Step 1s)**. **b)** The performance of UB-FVC is compared with GNB from 2 bits to 9 bits (full precision). UB-FVC stabilises at the 5-bit quantisation level beyond which no performance improvement is observed, achieving 90% prediction accuracy. **c)** In the inference stage of UB-FVC, when new sensor values are received, each 5-bit sensor value is used to query its own sub-LUT denoted as *Feature LUT X* to retrieve its BC, which becomes its vote. The most frequent class (i.e. statistical mode) is selected among all votes or BCs, which becomes the predicted class.

Univariate Bayes feature voting classifier

Metal oxide TFTs are at much earlier stage in the development cycle than silicon and consequently, to date, the most complex digital designs achieved with metal oxide TFTs have been less than a thousand (NAND2 equivalent) gates [19] [25].

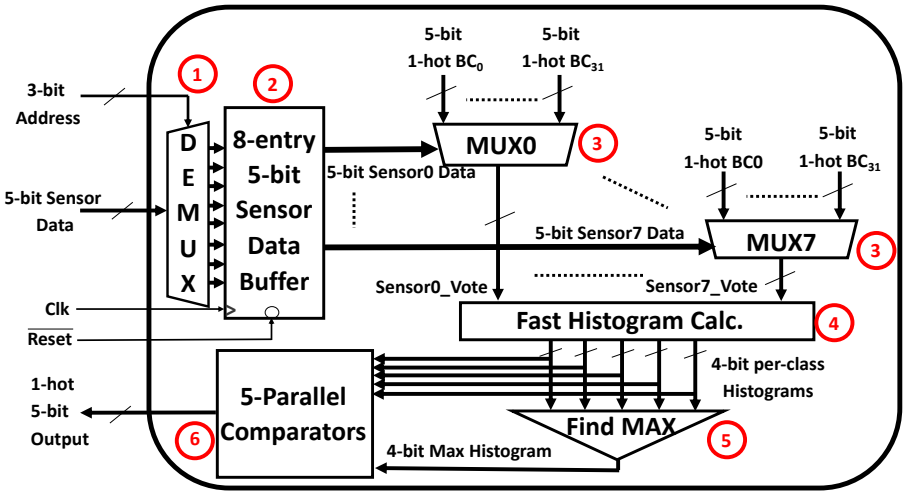
To build a more resource-efficient ML NFPE for our application, we develop a new ML algorithm termed “Univariate Bayes Feature Voting Classifier” or UB-FVC. The training stage of the UB-FVC is similar to the training stage of other ML algorithms where training is performed offline. The training algorithm of the UB-FVC is described in **Fig. 3a**. It is inspired by the GNB algorithm that accumulates the log-likelihood functions of all the features for each class and picks the best class (BC) with the maximum posterior probability as the predicted class, and stores the BC information in a Look-up Table (LUT) the LUT contents become the learned coefficients of the UB-FVC after the training stage completes.

We compare the performance of UB-FVC to GNB (which was the best ML algorithm) for our application, and show the results for varying levels of data quantisation in **Fig. 3b**. At the 5-bit quantisation level, the prediction accuracy of UB-FVC reaches 90%, which is only 2 percentage points behind GNB but still provides an acceptable prediction accuracy for our application.

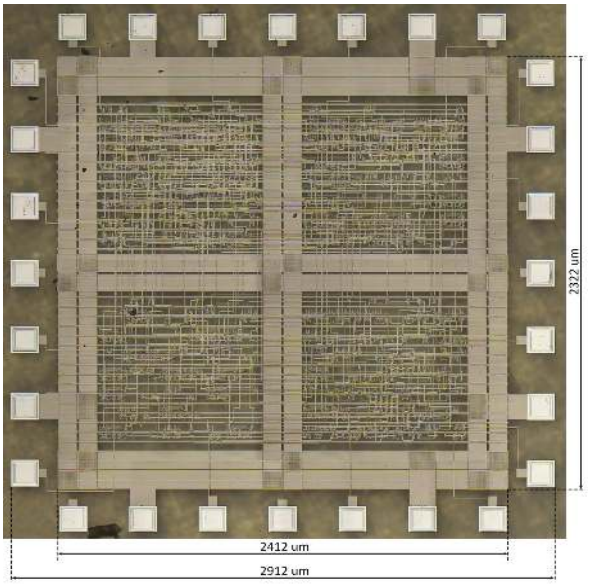
At the inference stage of UB-FVC as shown in **Fig. 3c**, only the LUT is used to make classifications. All the information needed to make a prediction are stored in the LUT. The sufficiency of the 5-bit data quantisation level for our application allows us to build a 32-entry LUT per feature. 5-bit feature data are received from eight sensors, and each 5-bit feature data is used to access the LUT associated with the feature to read out its BC. Then, a voter selects the most frequent class (i.e. statistical mode) among all eights BCs as the predicted class. The UB-FVC approach simplifies the hardware implementation for the odour classification application into table lookups and statistical mode computation.

Fig. 4a shows the microarchitecture of the UB-FVC inference stage described in **Fig. 3c**. The 5-bit sensor data values are received serially from the ADC, and are demultiplexed and stored in the sensor data buffer selected by the 3-bit sensor address

input. The five odour classes are encoded in a 5-bit one-hot format so that the position of the hot bit determines the class. Because the design is custom for the specific application, the LUTs are not stored in memory. Instead, the LUT entries (i.e. one-hot 5-bit predetermined BC values) are hardwired as inputs to the multiplexors to simplify the hardware complexity. Thus, the 5-bit sensor data is used as an address to select one of the thirty-two hardwired BC values, which becomes the vote of a sensor. After finding each vote per sensor, the statistical mode among eight sensor votes is computed by histogram calculation, maximum value determination and a set of parallel comparators. Except for the 8-entry sensor data buffer, everything else in the design is combinational logic.



(a)



(b)

Fig. 4 UB-FVC NFPE. **a)** The microarchitecture of the UB-FVC inference stage is shown. 5-bit sensor data are received serially and demultiplexed **(Block 1)** into the sensor data buffer **(Block 2)**. Each feature LUT is implemented as a multiplexor **(Block 3s)** where LUT entries are hardwired inputs. **Block 4** performs a fast histogram count calculation for the eight BCs or votes. Because classes are represented as one-hot values, the histogram count can be calculated very fast by adding the corresponding bits of the BCs (e.g. the most significant bits of the BCs are added together and so on). The fast histogram count calculation unit generates five histogram values (i.e. one per class) each of which has 4 bits to accommodate values from 0 count to 8 counts. The next step is to find the highest histogram value among the five classes in order to determine the class that has the highest count. The highest histogram value is calculated through a comparator reduction tree shown as the "Find MAX" block **(Block 5)**. Five parallel comparators **(Block 6)** take the five histogram values and compare each one with the highest histogram value from **Block 5** to find the statistical mode. It is possible to have more than one statistical mode in which case one class is picked from the leftmost order. **b)** Die photo of the NFPE implementing the UB-FVC microarchitecture.

Fabrication of UB-FVC based NFPE and measurement results

We fabricate the NFPE implementing the UB-FVC ML algorithm using *PragmatIC*'s 0.8μm process with n-type metal-oxide TFTs. To implement the NFPE, we need to build a standard cell library for the 0.8μm process. The standard cell library based on the metal oxide TFT technology contains 57 cells.

The micrograph of the NFPE flexIC is shown in **Fig. 4b**. It utilises 23 pins, which includes 8 power/ground and 15 input/output. The power and ground rails are routed through the combinational logic that gives the impression of having four symmetric blocks. The entire chip consists of combinational logic except for the 8-entry sensor data buffer that stores the sensor data at the interface. The clock is implemented as an unbuffered tree driven from an input pin. The nominal operating voltage is 4.5V. Output pins are driven by pseudo-CMOS buffers with a maximum driving capability of 1mA.

Table 2 Comparison between different complex digital circuits designed with metal-oxide TFTs on flexible substrates. The first column describes the figure of merit in terms of technology, design and implementation. The second column is our work while last two columns show the closest prior art.

"Figure of merit"	"NFPE"	"Flexible 8-bit ALU [19]"	"Flexible NFC Tag [25]"
Area (mm^2)	5.6	225.6	50.55
Technology (μm)	0.8 metal-oxide TFT	5 dual-gate organic + metal-oxide TFTs	1.5-2 metal-oxide TFT
Logic type	Unipolar n-type resistive load	Complementary oxide & organic	N-type pseudo-CMOS
Supply voltage (V)	4.5	6.5	3 & 6
Chip pin count	23	30	N/A
Number of devices	3132 (2084 TFTs + 1048 Resistors)	3504	1712
Max circuit clock frequency (kHz)	104	2.1	N/A
NAND2-equivalent gate count	1024	876	428

Power consumption (mW)	7.2	Not reported	7.5
Gate density (gates/mm^2)	183	4	9

We measure eight fully functional NFPEs, and all measurements are performed at room temperature whilst the flexible foil remains on its glass carrier. The implementation and fabricated chip measurement results are tabulated in **Table 2**, and are compared to the closest prior art that use metal-oxide TFTs on flexible substrates [19] [25] that developed complex digital circuits with metal-oxide TFTs on flexible substrates. The median power consumption among eight NFPEs is 7.2mW at 4.5V. The maximum circuit clock frequency is 104kHz. An NFPE comprises 2084 n-type TFTs and 1048 resistors with a core area of 2.32mm x 2.41mm. The NAND2 equivalent gate count is 1024 gates, which makes it the most complex digital circuit fabricated with metal-oxide TFTs. It has 20-45x higher gate density in terms of the number of gates per mm^2 area than the prior art. The chip simulation and measurement details can be found in the *Methods* section.

Conclusions

We have reported a domain-specific natively flexible processing engine (NFPE) fabricated with 0.8 μm metal-oxide TFT technology. We developed a resource-efficient ML algorithm, termed univariate Bayes feature voting classifier (UB-FVC), for sweat odour classification, and implemented the UB-FVC inference stage in hardware as an NFPE. The NFPE requires only 1,024, which is lower than the number required (3,000 gates) when implementing other ML algorithms like Gaussian Naïve Bayes. Furthermore, compared to other digital flexICs based on metal-oxide TFTs, our flexIC has a more complex design and a higher gate density per area by 20-45 times.

NFPEs are of potential use in emerging applications such as smart packaging, fast moving consumer goods (FMCG), and mass-market healthcare. The common characteristics of these markets are that the relevant products are low cost, high volume and have short lifetimes. For example, a smart label with a flexible e-nose sensor array

and ML NFPE could be attached to a meat package in order to monitor food quality and safety. The shelf life of such a product is normally a few days, after which the package (along with flexible electronics components) is disposed of or recycled.

Alternatively, a smart wound dressing that contains flexible temperature and e-nose sensors attached to an ML NFPE could perform real-time monitoring of the wound by processing sensor outputs and predicting the healing of the wound. The lifetime of the dressing is similar to the meat package (a few days), but here the predicted output could be a binary one, signalling the healing status as “healed” or “unhealed”. The performance metric would be the prediction accuracy of the healing status decision, which may be very high (over 95 %, for example) to avoid false positives, since the prediction outcome may be safety critical for the patient. A number of ML algorithms would need to be modelled on the training datasets in order to find the best performing ML model to meet the performance requirements of the application.

Like our UB-FVC algorithm, a large number of ML algorithms (e.g. GNBs, neural networks including the state-of-the-art deep learning neural nets) use offline training/learning. The parameters are learned during the offline training stage. These learned parameters do not change during inference, and can only change when the ML algorithm is re-trained offline with new datasets. After retraining, the parameters are updated in the rewritable memory of a system through a software/firmware update. A ML NFPE that is based on one of these ML algorithms and used, for example, in FMCG will be of single use and have short shelf lifetimes. Programmability may not be required for the ML NFPEs because the learned parameters do not need to change during the short lifetime of an FMCG product, so they can be hardwired instead of requiring a rewritable memory.

Finally, the development of CMOS technology is a vital step towards low-power circuit designs and larger scale integration of metal-oxide TFTs. To date, no commercially viable route to CMOS based on metal-oxide technology has been found due to the lack of an appropriate p-type material. Without CMOS, complex IC design will be constrained, but, as we have shown here, domain-specific NFPEs that have a reasonable gate and power budget can be built with n-type TFT logic.

Methods

FlexIC fabrication methodology

The forward transfer characteristic for an IGZO TFT is shown in **Extended Data Fig. 1**. The linear regime transfer curve plot is shown for an n-type metal-oxide TFT at logarithmic scale. The transistor has a drain voltage of 0.1V and a threshold voltage of 0.61V, a sub-threshold slope of 0.13V/dec, an on-current of 2.5μA and an off-current below the noise floor of the measurement equipment.

FlexLogIC® is based on a 200mm diameter wafer where repeated instances of the flexIC design are generated by running several sequences of material deposition, patterning and etching. For ease of handling and to allow industry standard tool to be used and sub-micron patterned features to be achieved, the flexible substrate is spin-coated onto glass at the outset of production. The process has been optimised to ensure that the thickness variation is significantly less than 3% over 20mm lateral distance. Substrate processing conditions have also been carefully optimised to minimise film stress and substrate bow. Feature patterning is achieved using a photolithographic stepper tool which images a shot that is repeated at multiple instances across the 200mm diameter wafer. Each shot is focussed individually which further compensates for any thickness variation within the spun-cast film. The measurements were carried out using process control monitoring structures. All measurements presented in this article were taken before release of the flexible foil from the glass carrier.

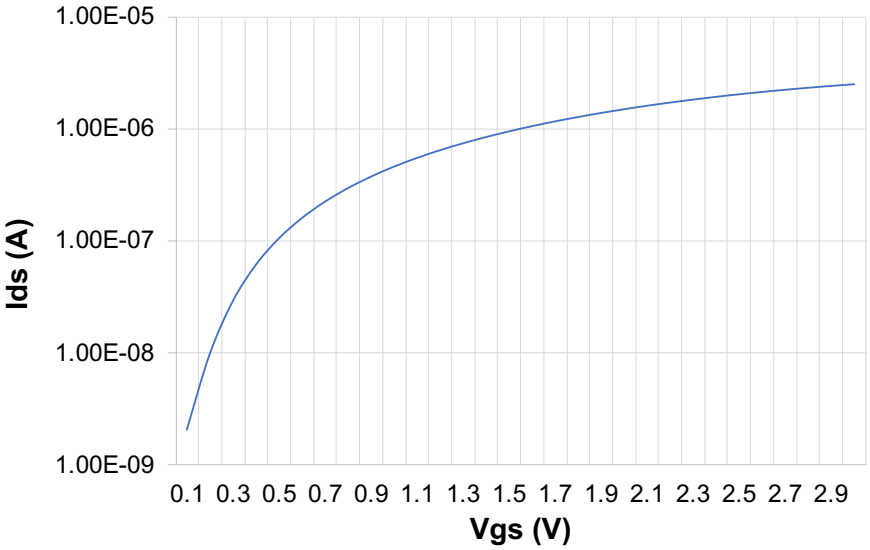
Chip simulation and measurement validation methodology

Extended Data Fig. 2 and **Extended Data Fig 4** depict simulation and chip measurement results of the UB-FVC based NFPE with a tester clock frequency of 104kHz and supply voltage of 4.5V. The input test vectors for both simulation and measurement results are the test datasets from our sweat odour classification application. We use over 500 test vectors (each test vector has eight 5-bit sensor values) to stimulate the simulation model and the fabricated chip, and the results of simulation match the results of the actual measurements for all test vectors.

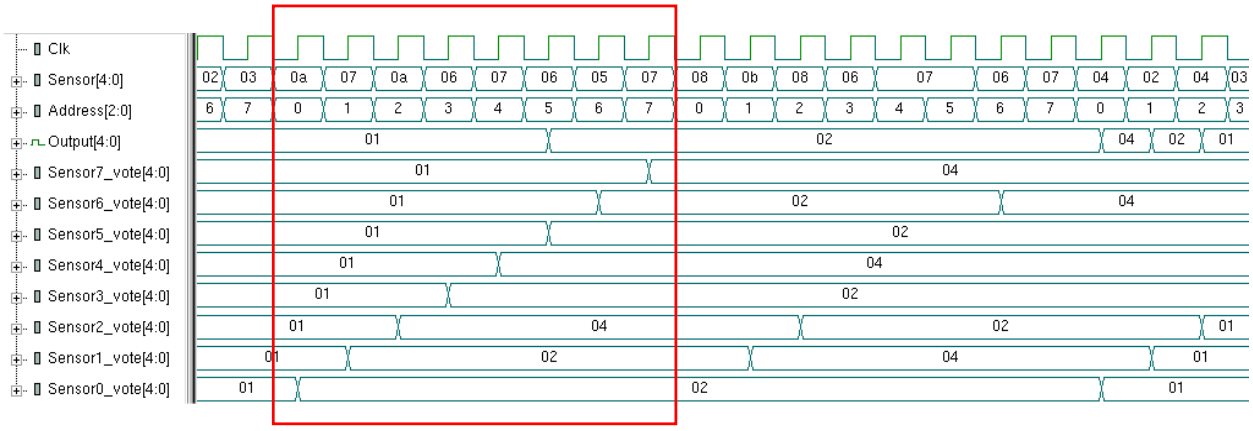
Simulation results in **Extended Data Fig. 2** demonstrate the functionality of the UB-FVC hardware. Eight 5-bit sensor data arrives serially at each cycle starting from address 0 to 7. Each 5-bit Sensor value is stored in the 8-entry 5-bit sensor data buffer selected by the 3-bit *Address* input from address 0 to address 7. For example, *Sensor0* stores the value of "0x0A" at address 0 in the buffer, and *Sensor1* stores "0x07" at address 1 and so on. This is shown inside the red rectangle drawn in the waveform. Then, each 5-bit Sensor data stored in the buffer is used to select one of the 32 5-bit hardwired best class (BC) coefficients. For example, Sensor0 has the value of "0x0A". The value will be used to access the 10^{th} BC coefficient for Sensor0. The predetermined one-hot encoded BC coefficients are hardwired in the microarchitecture and shown in **Extended Data Fig. 3.** The 10^{th} BC coefficient for Sensor0 is "2" in one-hot encoded format, which becomes the vote for Sensor0 as denoted by *Sensor0_vote* in **Extended Data Fig. 2**. Sensor1 has the value of "0x07", and the 7^{th} BC coefficient for Sensor1 is also "2" in one-hot encoded format, which becomes *Sensor1_vote*. After finding the BC values of all sensors, the eight votes are {2, 2, 4, 2, 4, 2, 2, 4}. The statistical mode is 2 among all these eight votes, so *Output* becomes 2.

The measurement results **Extended Data Fig. 4** confirm the correct functionality demonstrated in simulations with exact test stimulus. Each individual output bit in *Output_X* is shown in the waveform. The output settles at 2 after all sensor data are received. This can be seen in the waveform when *Output_1* becomes 1 and the remaining output bits are 0.

Additionally, the slow rising and falling edges can be observed on the *Output_X* signals. This is due to the experimental setup capacitive loading of the logic analyser and the limited drive strength capabilities of the output buffers. Furthermore, small glitches can be observed which correspond to the combinational nature of the histogram calculation.



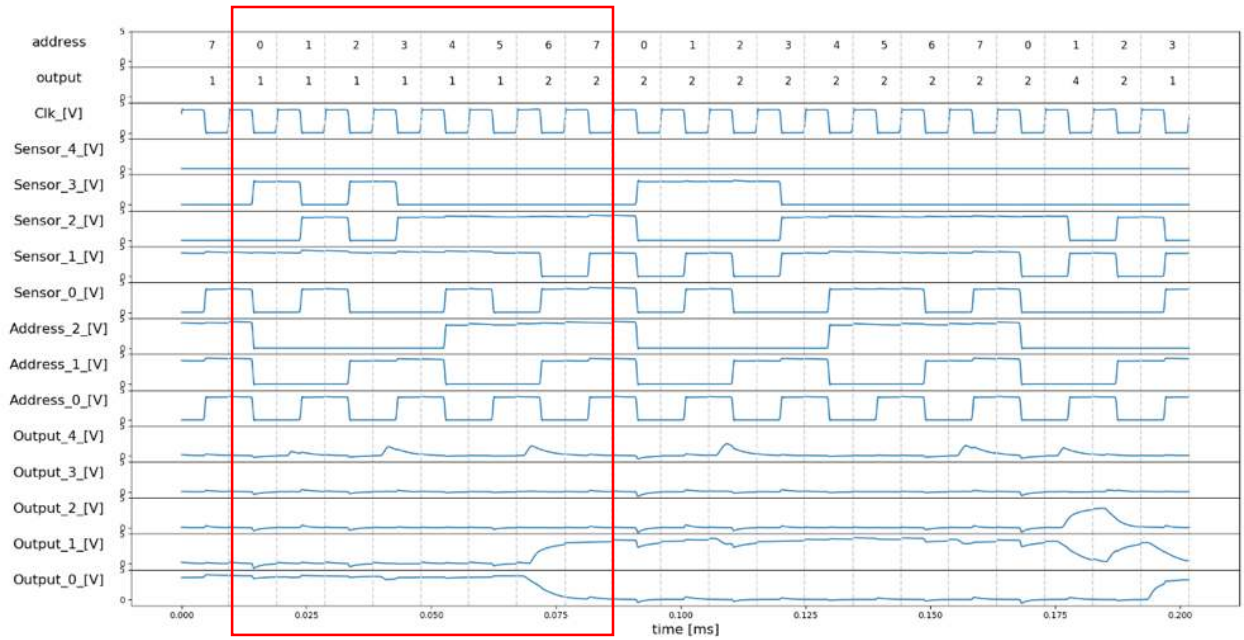
Extended Data Fig.1 Forward transfer characteristic of a metal-oxide TFT.



Extended Data Fig. 2 NFPE simulation results. The column on the left shows the list of input, intermediate and output signals. *Sensor[4:0]* and *Address[2:0]* are the inputs, and represent the 5-bit sensor data, and 3-bit sensor address, respectively. *SensorX_vote[4:0]* is intermediate signals, and represent the 5-bit BC coefficients (essentially votes) for each sensor. Finally, *Output[4:0]* shows the 5-bit one-hot predicted class as output.

Sensor Data Value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Sensor0_Vote	2	1	1	1	1	2	2	2	2	2	2	4	4	4	4	4	4	4	8	8	8	8	8	8	16	16	16	16	16	16	16	16
Sensor1_Vote	1	1	1	1	1	1	2	2	2	2	4	4	4	4	4	4	4	8	8	8	8	8	8	16	16	16	16	16	16	16	16	16
Sensor2_Vote	2	2	1	1	1	2	2	2	2	4	4	4	4	4	4	8	8	8	8	8	16	16	16	16	16	16	16	16	16	16	16	16
Sensor3_Vote	1	1	1	1	1	2	2	4	4	4	8	8	8	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Sensor4_Vote	2	2	1	1	1	2	2	4	4	4	8	8	8	16	16	16	16	16	16	16	16	16	16	16	16	16	16	8	8	8	8	8
Sensor5_Vote	2	2	1	1	1	2	2	2	4	4	4	8	8	8	8	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Sensor6_Vote	1	1	1	1	1	2	4	4	8	8	8	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Sensor7_Vote	1	1	1	1	1	2	2	4	4	4	8	8	8	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16

Extended Data Fig. 3 One-hot coefficients to represent BCs. The top row shows the sensor data values from 0 to 31. For each sensor value, the BC or vote of the sensor is predetermined and hardwired in the microarchitecture.



Extended Data Fig. 4 NFPE chip measurement results of a fabricated chip for the same setup as in the simulation. This is the waveform captured from the logic analyser. All inputs and outputs are shown as individual signals. *Sensor_X* and *Address_X* are input signals, and represent the sensor data and address. *Output_X* represents the 5-bit one-hot predicted class output signals.

Data availability

The data that support the plots within this paper and other findings of this study are available from the corresponding author upon reasonable request.

Code availability

The code used to generate the plots within this paper is available from the corresponding author upon reasonable request.

References

[1] *OE-A Roadmap for Organic and Printed Electronics* White Paper 8th Edn (OE-A, 2020).

[2] Nathan, A. et al. Flexible Electronics: The Next Ubiquitous Platform. *Proceedings of the IEEE* **100,** 1486-1517 (2012).

[3] Kelly, P.H.J. Architecture and Software for When There's no Longer Plenty of Room at the Bottom. *Dagstuhl Reports* **7**, 2 (2017).

[4] Lee, E.A. Programmable DSP architectures: Part I. *ASSP Magazine IEEE* **5,** 4-19 (1988).

[5] Fisher, J.A., Faraboschi, P. & Desoli, G. Custom-fit processors: letting applications define architectures. *Proceedings of the 29th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-29)* 324-335 (1996).

[6] Hennessy, J.L. & Patterson, D.A. A New Golden Age for Computer Architecture. *Communications of the ACM* **62,** 48-60 (2019).

[7] *Flex-ICs: Silicon-on-Polymer Products* (American Semiconductor, 2020); https://www.americansemi.com/flex-ics.html

[8] Gupta, S., Navaraj, W.T., Lorenzelli, L. & Dahiya. R. Ultra-thin chips for high-performance flexible electronics. *npj Flexible Electronics* **2,** 8 (2018).

[9] Harendt, C. et al. Hybrid Systems in Foil (HySiF) exploiting ultra-thin flexible chips. *44th European Solid-State Device Research Conference (ESSDERC)* 210-213 (2014).

[10] Khan, S., Lorenzelli, L. & Dahiya, R. Technologies for Printing Sensors and Electronics over Large Flexible Substrates: A Review. *IEEE Sensors Journal* **15,** 3164-3185 (2015).

[11] Takayama, T. et al. A CPU on a plastic film substrate. *Symposium on VLSI Technology* 230-231 (2004).

[12] Dembo, H. et al. RFCPUs on glass and plastic substrates fabricated by TFT transfer technology. *IEEE International Electron Devices Meeting (IEDM)* 125-127 (2005).

[13] Karaki, N. et al. A flexible 8b asynchronous microprocessor based on low-temperature poly-silicon TFT technology. *IEEE International Solid-State Circuits Conference (ISSCC)* 272-273 (2005).

[14] Kurokawa, Y. et al. UHF RFCPUs on Flexible and Glass Substrates for Secure RFID Systems. *IEEE Journal of Solid-State Circuits* **43,** 292-299 (2008).

[15] Hills, G. et al. Modern microprocessor built from complementary carbon nanotube transistors. *Nature* **572,** 595–602 (2019).

[16] Petti, L., et al. Metal oxide semiconductor thin-film transistors for flexible electronics. *Applied Physics Reviews* **3,** 021303 (2016).

[17] Myny, K. The Development of flexible integrated circuits based on thin-film transistors. *Nature Electronics* **1,** 30-39 (2018).

[18] Myny, K., van Veenendaal, E., Gelinck, G.H., Genoe, J. & Dehaene, W. An 8-Bit, 40-Instructions-Per-Second Organic Microprocessor on Plastic Foil. *IEEE J. Solid-State Circuits* **47,** 284-291 (2012).

[19] Myny, K. et al. 8b Thin-film microprocessor using a hybrid oxide-organic complementary technology with inkjet-printed P^2ROM memory. *IEEE International Solid-State Circuits Conference (ISSCC)* 486-487 2014.

[20] *FlexLogIC* (PragmatIC, 2020); https://www.pragmatic.tech/technology

[21] Torsi, L., Magliulo, M., Manoli, K. & Palazzo, G. Organic field-effect transistor sensors: a tutorial review. *Chem Soc Rev*. **42,** 8612-8628 (2013).

[22] Tate, D.J., et al. Fully Solution Processed Low Voltage OFET Platform for Vapour Sensing Applications. *ISOCS/IEEE International Symposium on Olfaction and Electronic Nose* 1-3 (2017).

[23] Rahmanudin, A. et al. Robust High-Capacitance Polymer Gate Dielectrics for Stable Low-Voltage Organic Field-Effect Transistor Sensors. *Advanced Electronic Materials* **6,** 1901127 (2020).

[24] Ozer, E. et al. Bespoke Machine Learning Processor Development Framework on Flexible Substrates. *IEEE International Conference on Flexible and Printable Sensors and Systems (FLEPS)* 1-3 (2019).

[25] Myny, K. et al. A flexible ISO14443-A compliant 7.5mW 128b metal-oxide NFC barcode tag with direct clock division circuit from 13.56MHz carrier. *IEEE International Solid-State Circuits Conference (ISSCC)* 258-259 (2017).

Acknowledgements

This work is partially supported by the Innovate UK through the “PlasticArmPit: Accelerating the Development of Flexible Integrated Smart Systems (No 103390)” project.

Author contribution statement

EO and GB conceived the UB-FVC model. EO, JK and JB designed and implemented the model as an NFPE. AR, AS, CR and SW developed the fabrication process and methodology for the NFPE. All authors contributed to analysis of the data generated in the design, implementation and fabrication of the NFPE. EO, JK, JM, JB, CR and SW wrote the paper.

Competing interest statement

We have no financial or non-financial competing interests.

Figure captions

Fig. 1 OFET sensors and system architecture of the flexible smart system. a) A single OFET sensor and an e-nose sensor array consisting of eight OFET sensors. **b)** System architecture of the flexible smart system consisting of the e-nose sensor array with ADCs on a flexible substrate and the natively flexible hardwired ML processing engine on a flexible substrate

Fig. 2 Design Space Exploration with Various ML Algorithms. a) Prediction accuracies are shown for various standard ML algorithms on the odour classification application varying data quantisation levels from 2 bits to 9 bits (full precision). The ML training and performance evaluation methodology follows the standard ML practice: The dataset is split into training and test datasets. Then, the ML algorithms are trained offline using the training datasets. Once the training is complete, the performance of the ML algorithms with learned parameters are evaluated with the test datasets. We use a 5-fold cross-validation methodology to avoid overfitting. Classification prediction accuracy is used as a metric that is defined as how accurate the prediction is with respect to the ground truth. No visible difference is observed between 5-bit and full precision data representations. The best performing ML algorithm is GNB with a prediction accuracy of 92%. **b)** The 5-bit GNB design variants are compared in terms of gate count and execution time. The three GNB variants are created by either sharing or duplicating the multiply-accumulate (MAC) units for *features (i.e. sensor inputs) and* classes *(i.e. outputs)*. Sharing a MAC among classes and features reduces the number of gates while increasing the execution time. On the other hand, separate MACs will increase the number of gates while improving the execution time by doing computations in parallel. The smallest GNB implementation is the one with a shared MAC for classes and separate MACs for features and is comprised of over 3000 gates.

Fig. 3 Univariate Bayes feature voting classifier (UB-FVC). a) The training algorithm of UB-FVC computes the class posterior probabilities for each feature *(i.e.* sensor*)*

493 independently **(Step 3)**, and picks the best class (BC) for the feature **(Step 4)**. Because
494 feature values are quantised values from 0 to 2^n-1 where n is the data bitwidth, the
495 algorithm computes the BC for each value of a feature **(Step 2)** and stores them in a
496 look-up table (LUT) per feature and value **(Step 5)**. These steps are repeated for all
497 features **(Step 1s)**. **b)** The performance of UB-FVC is compared with GNB from 2 bits to
498 9 bits (full precision). UB-FVC stabilises at the 5-bit quantisation level beyond which no
499 performance improvement is observed, achieving 90% prediction accuracy. **c)** In the
500 inference stage of UB-FVC, when new sensor values are received, each 5-bit sensor
501 value is used to query its own sub-LUT denoted as *Feature LUT X* to retrieve its BC,
502 which becomes its vote. The most frequent class (i.e. statistical mode) is selected
503 among all votes or BCs, which becomes the predicted class.

504 **Fig. 4 UB-FVC NFPE**. **a)** The microarchitecture of the UB-FVC inference stage is
505 shown. 5-bit sensor data are received serially and demultiplexed **(Block 1)** into the
506 sensor data buffer **(Block 2)**. Each feature LUT is implemented as a multiplexor **(Block**
507 **3s)** where LUT entries are hardwired inputs. **Block 4** performs a fast histogram count
508 calculation for the eight BCs or votes. Because classes are represented as one-hot
509 values, the histogram count can be calculated very fast by adding the corresponding
510 bits of the BCs (e.g. the most significant bits of the BCs are added together and so on).
511 The fast histogram count calculation unit generates five histogram values (i.e. one per
512 class) each of which has 4 bits to accommodate values from 0 count to 8 counts. The
513 next step is to find the highest histogram value among the five classes in order to
514 determine the class that has the highest count. The highest histogram value is
515 calculated through a comparator reduction tree shown as the “Find MAX” block **(Block**
516 **5)**. Five parallel comparators **(Block 6)** take the five histogram values and compare
517 each one with the highest histogram value from **Block 5** to find the statistical mode. It is
518 possible to have more than one statistical mode in which case one class is picked from
519 the leftmost order. **b)** Die photo of the NFPE implementing the UB-FVC
520 microarchitecture.

521 **Extended Data Fig.1 Forward transfer characteristic of a metal-oxide TFT**.

Extended Data Fig. 2 NFPE simulation results. The column on the left shows the list of input, intermediate and output signals. *Sensor*[4:0] and *Address*[2:0] are the inputs, and represent the 5-bit sensor data, and 3-bit sensor address, respectively. *SensorX_vote*[4:0] is intermediate signals, and represent the 5-bit BC coefficients (essentially votes) for each sensor. Finally, *Output*[4:0] shows the 5-bit one-hot predicted class as output.

Extended Data Fig. 3 One-hot coefficients to represent BCs. The top row shows the sensor data values from 0 to 31. For each sensor value, the BC or vote of the sensor is predetermined and hardwired in the microarchitecture.

Extended Data Fig. 4 NFPE chip measurement results of a fabricated chip for the same setup as in the simulation. This is the waveform captured from the logic analyser. All inputs and outputs are shown as individual signals. *Sensor_X* and *Address_X* are input signals, and represent the 5-bit sensor data and 3-bit address. *Output_X* represents the 5-bit one-hot predicted class output signals.