Computing Based on Material Training: Application to Binary Classification Problems

E. Vissol-Gaudin, A. Kotsialos, C. Groves, C. Pearson, D.A. Zeze, M.C. Petty
School of Engineering and Computing Sciences, Durham University, Durham, United Kingdom
E-mail:{eleonore.vissol-gaudin, apostolos.kotsialos, chris.groves,
christopher.pearson, d.a.zeze, m.c.petty}@durham.ac.uk

***Abstract*—Evolution-in-materio is a form of unconventional computing combining materials' training and evolutionary search algorithms. In previous work, a mixture of single-walled-carbon-nanotubes (SWCNTs) dispersed in a liquid crystal (LC) was trained so that its morphology and electrical properties were gradually changed to perform a computational task. Material-based computation is treated as an optimisation problem with a hybrid search space consisting of the voltages used for creating the electrical field and the material's infinitely possible SWCNT arrangements in LC. In this paper, we study solutions using synthetic data with a non-linear separating boundary. In addition, results for two real life datasets with partly merged classes are presented. The training process is based on a differential evolution (DE) algorithm, which subjects the SWCNT/LC material to repeated electrical charging, leading to progressive morphological and electric conductivity modifications. It is shown that the material configuration the DE algorithm converges to form a non-negligible part of the solution. Furthermore, the problem's complexity is relevant to the properties of the resulting physical solver. The material structures created when training for a problem allow the retraining for a less complex one. The result is a doubly-trained material that keeps the memory of the original more complex problem. This is not the case for doubly-trained materials where initial training is for the less complex problem. The optimal electric field found by the DE algorithm is also a necessary solution component for the material's output to be interpreted as a computation.**

I. Introduction

Within the broader framework of unconventional computing methods, Evolution-in-Materio (EiM) is an approach based on the idea of exploiting one or more of the physical properties of a piece of material for performing a computation task. The selected physical properties must be manipulable in the sense that it should be possible to adapt and mould them using a set of well defined external stimuli. By embedding the material within an evolutionary search algorithm, consisting of both hardware and software, the selected physical properties are evolved towards a computation inducing state. There are five elements in an EiM based computation: (a) the material properties used; (b) the hardware interfacing with the material for applying inputs to the appropriate material to be shaped and for reading responses that can be interpreted as a computation; (c) the interpretation scheme used for transforming the material's response to a computation; (d) the algorithm used for configuring the material and evolving it; and finally (e) the computation problem itself. For a review of EiM see [13].

Depending on the material used and the physical properties to be exploited, different EiM computing devices can be developed. Liquid Crystals (LCs) from a display screen have been used as the material part of EiM for evolving robot controllers [5], a tone discrimination device [3] and logic gates [4]. In [9], [11], and [17] a dry mix of Single-Walled Carbon Nanotubes (SWCNT) with a polymer were used as the computational material and its electrical conductance was used as the manipulated property for solving the problem of calculating Boolean functions using a threshold interpretation scheme; the same material is used in [14] and [15] for solving optimisation problems. In [12], [22], [21] and [20] the material used is a solution of SWCNT with LC in liquid rather than in dry solid state. The physical property used for evolving the material is its electrical conductivity and the ability of the SWCNT to form percolation paths within the LC. The problems addressed are variations of binary classification problems using artificially created datasets. This paper's first contribution is to follow the same approach but instead of artificial datasets to use real life medical data.

The amorphous state of the SWCNT/LC liquid solution does not allow for a deterministic input/output model. Hence, the material system is treated as a black box and the binary classification problems are solved by following a supervised learning approach. The learning problem is formulated as an optimisation problem splitting the available dataset into training and verification sets. Hence, there is no explicit algorithm for solving the classification problem, but rather the training of the material for the specific computation task. Before training, the material is not able to produce a response that can be translated into a computation. A statement declaring the class a particular datum belongs to is available post training, after having created the conductive network structures of SWCNT within the material during the process of evolution. It is in this sense that the material is trained, i.e. it has its morphology changed, for solving a particular problem and not by being able to execute a number of discrete algorithmic steps towards providing an answer to the problem. This is different from the classical computation approach, which can be implemented by digital or analogue devices. Although the computation can be considered as analogue in nature, it is the macro-behaviour of the emerging material properties that is used. The alignment and formation of percolation paths of SWCNT within the LC host is enforced by the iterative nature of the evolutionary search conducted, until a notion of computation error is minimised or becomes zero. Just as in the analogue computing case, there is a persistent issue with

the computation's accuracy. This is a problem which can be addressed by improving the quality of the hardware used and the efficiency of the training algorithm.

The training algorithm searches a hybrid space of solutions, which consists of a normal subspace spanned by the configuration inputs or stimuli that make the material change its morphology, and of the formed network of SWCNT within the body of the material. The space of possible network configurations and associated percolation paths is infinitely dimensional. The search algorithm has only implicit access to it through the configuration inputs. The repetitive application of such inputs together with the computation inputs is expected to create a suitable internal structure. Hence, a solution is described by the optimal configuration inputs and the resulting evolved conductive network of SWCNT.

The second contribution of this paper, is a preliminary investigation of the contribution of the two parts of the solution, i.e. the configuration inputs and the material itself, to the optimum solution identified by the search algorithm. The question posed concerns the contribution of the post evolution material state and the optimal configuration inputs where the best result was obtained, to the overall computing device.

These two contributions continue the work reported in [12], [20], [21] and [22]. The hardware platform and material are discussed in section II. The classification problems are described in section III and the optimisation problem formulation is briefly outlined in section IV. Section V discusses the new results obtained and section VI concludes this paper with some future directions of work.

II. Hardware Implementation and Material

Three main hardware components are used in an EiM experimental set-up; a computer, an evolvable mother board (EM) and a material.

During an experiment, Evolutionary Algorithms (EAs) are run on the computer, producing signals that are transmitted to an *mbed* microcontroller fixed on the EM. These signals are translated into voltage levels by the *mbed* and the resulting configuration voltages are sent to the material, along with the voltages defining the computational problem, through a set of digital to analogue converters (DACs). Currents flowing through the material are measured and sent back to the microcontroller via an analogue to digital converter (ADC). Finally, the interpretation of the material's output is transmitted to the computer and used by the EAs for evaluating the objective function value and subsequently producing new signals.

The material sample is a 0.05% SWCNT/LC mixture obtained by dispersing SWCNTs in a LC using an ultrasonic probe. The nanotubes are 1/3 metallic, 2/3 semiconducting whilst the E7 nematic liquid crystal molecule presents no comparative conductivity. It was reported in [21] [22] that using the experimental set-up described, it was not possible to train a stand-alone LC solution in liquid state to perform a computation.

The SWCNT/LC blend has a non-linear relationship between voltage and current which is exploited by the EiM process. It has been observed in [23] that the nanotubes tend to bundle and form complex percolation paths between electrodes under an applied electric field. The LC provides a liquid medium within which the resulting conductive networks can be modified [12], adding a level of complexity to the search space as compared to solid CNT-based samples.

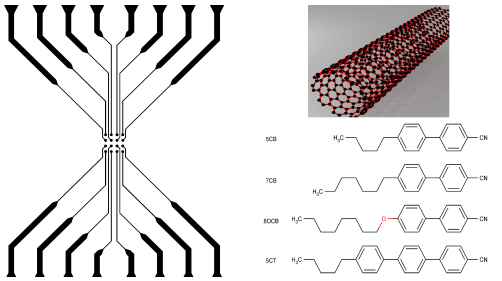


Fig. 1. Gold micro-electrode array with $50\mu m$ contacts and $100\mu m$ pitch, SWCNT and E7 LC molecule.

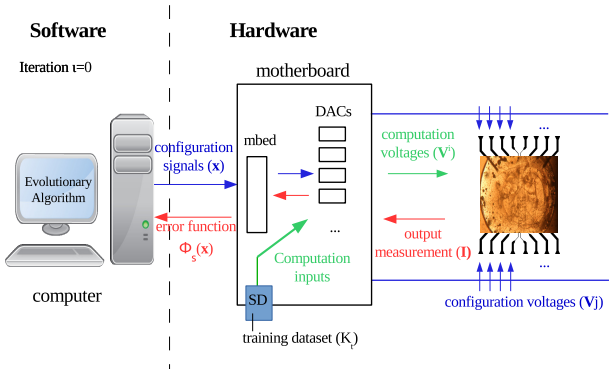


Fig. 2. Representation of the hardware implementation of EiM.

At the start of an experiment, the un-configured SWCNT/LC mixture is drop-cast within a 2.5 mm washer. The latter is fixed onto the gold micro-electrode array, itself deposited on a microscope slide using photolithography. The characteristics of the array are shown in Figure 1. It is connected to the hardware and used as a means to transfer voltages and measure currents to and from the material respectively.

The schematic representation of the experimental set up and procedure is shown in Figure 2.

III. Classification problems

Four binary classification datasets have been used in this paper. The choice of datasets has been partly directed by two experimental constraints; the number of input and output pins on the micro-electrode array, respectively fourteen and two, and the maximum voltage levels, $8\ Volts$, allowed in the hardware components and evolvable material. A dataset is composed of K instances which are translated into computation inputs, or voltage levels $\mathbf{V}^{in} \in [0, 8]^m$ (Volts) in the physical space, where m is the number of class features.

Two out of the four problems are represented by two-dimensional ($m = 2$), separable datasets, created artificially. One is diagonally separable, the V1C dataset, and the other, the

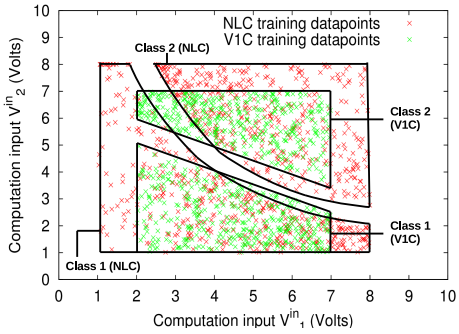


Fig. 3. Two binary classification problems represented by diagonally and non-linearly separable classes.

NLC dataset, has two classes separated by a hyperbola. The complexity of these problems is measured using the Fisher criterion f_1 [7] defined for a binary problem as

$$f_1 = \frac{(\mu_1^j - \mu_2^j)^2}{(\sigma_1^j)^2 + (\sigma_2^j)^2} \tag{1}$$

where μ_i^j and σ_i^j is the mean and the standard deviation of feature j for classes 1 and 2, respectively. Taking the maximum f_1 over all attributes being, for V1C $f_1 = 10.078$ and for NLC $f_1 = 2.097$, i.e. V1C is less complex than NLC.

During an experiment, the datasets are separated into training and verification subsets of size $K_t = 800$ and $K_v = 4000$, respectively. Figure 3 shows the distribution of training data in the 2D feature space.

The other two problems relate to the Bupa liver disorder (BPC) and mammographic mass (MMC) datasets, with parameters given in Table I. These are real-life problems retrieved from the UCI repository [10]. They are spread across more than two dimensions, with both non-linear boundaries and overlapping areas, as seen in Figure 4. A metric related to the overlap of classes along given features is given by

$$f_2 = \prod_{j=1}^{m} \frac{\min\{U_{1,j}, U_{2,j}\} - \max\{L_{1,j}, L_{2,j}\}}{\max\{U_{1,j}, U_{2,j}\} - \min\{L_{1,j}, L_{2,j}\}} \tag{2}$$

where

$$U_{1,j} = \max\left\{V_j : \mathbf{V}^{in} \in C_1\right\}, U_{2,j} = \max\left\{V_j : \mathbf{V}^{in} \in C_2\right\}$$
$$L_{1,j} = \min\left\{V_j : \mathbf{V}^{in} \in C_1\right\}, L_{2,j} = \min\left\{V_j : \mathbf{V}^{in} \in C_2\right\}.$$

From the two criteria, we can rank the different problems according to their complexity, with V1C being the simplest and BPC the most complex. The Fisher criterion over all attributes, as well as the measure of overlap defined in eqn. 2 are reported in Table I.

A number of papers using the BPC dataset, such as [1, 8], split it into $K_t = 276$ and $K_v = 69$, which are the values taken in our experiments. The MMC dataset on the other hand has been split arbitrarily into $K_t = 200$ and $K_v = 630$. In each case, prior to implementation, the datasets were normalised to fit the maximum voltage constraint.

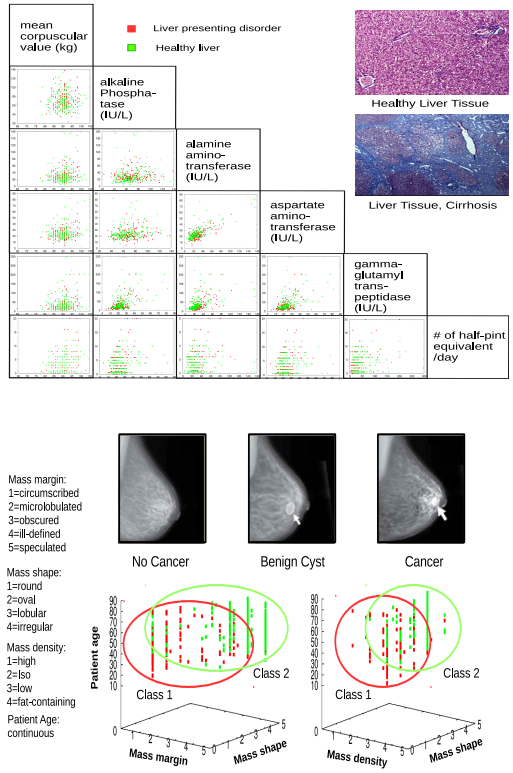


Fig. 4. Representation of the bupa liver disorder dataset top and mammographic mass dataset (bottom) with their attributes (photo taken from [6] and [19] respectively).

TABLE I
V1C, NLC, BPC AND MMC PROBLEMS' PARAMETERS.

Name	number of attributes	number of instances	Fisher criterion	volume of overlap (%)
V1C	2	4800	10.078	0
NLC	2	4800	2.097	0
MMC	4	831	0.809	1.4
BPC	6	347	0.081	4.4

IV. Problem Formulation and Algorithms

The process of evolving a piece of material into a computing device is formulated as an optimisation problem. There are sixteen connections on the micro-electrode array, twelve of which are used. Two of those are used for sending computation inputs as voltage pulses of amplitude $\mathbf{V}^{in} = (V_1^{in}, V_2^{in})$ and eight are used for sending configuration voltages as pulses within the range $V_j \in [V_{\min}, V_{\max}]$, $j = 1, \ldots, 8$. The remaining two connections are reserved for measuring output currents $\mathbf{I} = (I_1, I_2)$ (A) when the material has been sent $\mathbf{V}^{in}$ and is under charge of the V_j's.

The possible locations where the two components of $\mathbf{V}^{in}$ are applied is considered as a decision variable. Using a simple increasing index scheme for assigning configuration voltages (e.g. if V_1^{in} is assigned to electrode 3 and V_2^{in} is assigned to 5, then the following assignment for the configuration inputs takes place: $V_1 \to 1$, $V_2 \to 2$, $V_3 \to 4$ $V_4 \to 6$, $V_5 \to 7$, $V_6 \to 8$, $V_7 \to 9$ $V_8 \to 10$) then there are ${}^{10}P_2 = 90$ possible

connection assignments. A continuous variable $p \in [1, 90]$ is defined and updated by the EA used rounded to the nearest integer during the iterations.

The optimisation problem's vector of decision variables, spread over a total of $D = 10$ dimensions, is defined as

$$\mathbf{x} = [V_1 \dots V_8 \ R \ p]^T \tag{3}$$

where R is continuous over $[R_{min}, R_{max}]$. For a specific electrode assignment p and set of configuration voltages V_j, the material's response to an input $\mathbf{V}^{in}$ is recorded. The response is the pair of direct current, $\mathbf{I} = (I_1, I_2)$ (A), measured at the two output terminals. It is the basis of a comparison scheme using R for deciding the class $\mathbf{V}^{in}$ belongs to.

Let $\mathbf{I}^{(k)}$ denote the pair of direct current measurements taken when input data $\mathbf{V}^{in}(k)$ from class C_i, $i = 1$ or $i = 2$, are applied *while* the material is subjected to configuration voltages V_j. $\mathbf{V}^{in}(k)$ and V_j are applied according to electrode assignment number p and scaling factor R is used. Also, let $C(\mathbf{V}^{in}(k))$ denote $\mathbf{V}^{in}(k)$'s real class and $C_M(\mathbf{V}^{in}(k), \mathbf{x})$ the material's assessment.

Different mapping schemes may be used for the calculation of C_M. A functional form of $C_M\left(\mathbf{V}^{in}, \mathbf{x}\right)$ must be specified for each problem before the training process and, since the material acts as a computing device, every $\left(\mathbf{V}^{in}, \mathbf{x}\right)$ must be mapped to one of the two possible classes. The mapping is performed by the interpretation scheme ansatz, which considers the computational inputs, the corresponding induced material responses and the continuous decision variable R used as threshold;

$$C_M(\mathbf{V}^{in}(k), \mathbf{x}) = \begin{cases} C_1 & \text{if} \quad I_1(k)V_1^{in}(k) + I_2(k)V_2^{in}(k) \leq R \\ C_2 & \text{if} \quad I_1(k)V_1^{in}(k) + I_2(k)V_2^{in}(k) > R \end{cases} \tag{4}$$

For every training data point $\mathbf{V}^{in}(k)$, $k = 1, \dots, K_t$ the error from translating the material response according to rules (4) is

$$\epsilon_{\mathbf{x}}(k) = \begin{cases} 0 & \text{if} \quad C_M(\mathbf{V}^{in}(k), \mathbf{x}) = C(\mathbf{V}^{in}(k)) \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

The mean error $\Phi_e(\mathbf{x})$ evaluated over the training data set for a particular solution $\mathbf{x}$ is

$$\Phi_e(\mathbf{x}) = \frac{1}{K_t} \sum_{k=1}^{K_t} \epsilon_{\mathbf{x}}(k). \tag{6}$$

A penalty term $H(\mathbf{x})$ is added to (6), given by

$$H(\mathbf{x}) = \frac{\sum_{j=1}^{8} V_j^2}{8V_{max}^2}. \tag{7}$$

The rationale behind this penalisation is that incremental and generally low levels of configuration voltages are preferable. Solutions where high V_j are applied can destroy material structures formed during evolution that contribute to the solution.

Hence, the total objective function $\Phi_s(\mathbf{x})$ for an arbitrary individual of the EA's population s is given by

$$\Phi_s(\mathbf{x}) = \Phi_e(\mathbf{x}) + H(\mathbf{x}) \tag{8}$$

The optimisation training problem to be solved is that of minimising (8) for a population of size S, subject to voltage bound constraints $V_j \in [V_{\min}, V_{\max}]$, $R \in [R_{\min}, R_{\max}]$, electrode assignment p and classification rule (4). $V_{\min} = 0$ Volts, $V_{\max} = 4$ Volts, $R_{\min} = 0.05$ and $R_{\max} = 15$.

Derivative-free, stochastic, population-base optimisation algorithms were considered to solve this problem, due to the complex and dynamic nature of the search space, as well as the fact that no analytical or stochastic model of the material's behaviour currently exists. Here, differential evolution (DE) [18], with a population size of $S = 10$ individuals has been implemented. The position of each individual over $d = 1, ...D$ dimensions, defined by the vectors of decision variables $\mathbf{x}$, is initialised using uniform distribution across the problem boundaries. It is then updated, dimension by dimension, at each iteration ι in the following eqn. 9,

$$x_d = \begin{cases} x_d^a + F(x_d^b - x_d^c) & \text{if } d = D \text{ or } r_d < CR \\ x_d & \text{otherwise.} \end{cases} \tag{9}$$

where the three vectors of decision variables, $\mathbf{x}^a$, $\mathbf{x}^b$ and $\mathbf{x}^c$ and randomly drawn from the population, $r_d \sim U(0, 1)$, the cross-over parameter is $CR = 0.7026$ and $F = 0.814$ is the differential weight. The value of these parameters are based on suggestions found in [16] and have been modified empirically for the problem undertaken.

V. Results and Discussion

A. Medical Datasets

Previous papers reported results for simple, proof-of-concept datasets [12, 21, 22]. The first part of our investigations focus on the training of un-configured SWCNT/LC mixture for medical datasets selected from the UCI repository. In order to assess the solutions currently obtained *in materio*, results to BPC and MMC problems, averaged over two tests, are presented in Table II. They are compared with those achieved over a range of neural network (NN) implementations. Results for both BPC and MMC are reported in [1], where a dentrite morphological NN (DMNN) is trained using DE. Further BPC solutions obtained with five different types of NN are reported in Table II. A description of the back propagation NN (BPNN), general regression NN (GRNN), radial basis function NN (RBFNN), probabilistic NN (PNN) and complementary NN (CMTNN) is presented in [8].

Finally, the SWCNT/LC results are compared with those obtained from a medical survey [2] where human accuracy on the MMC problem is investigated. Only two extremes are reported here, results for fellowship trained radiologists who are very good at distinguishing correctly the classes (12.00% error), and the other radiologists, who average 17.00% error. The difference between the two radiologist diagnosis is detailed in paper [2].

It can be seen that despite the problem's complexity, the SWCNT/LC material can be brought in a state where it is able to classify at best 18.85% and on average 20.51% of the 631 instances contained in the MMC verification dataset. This is less than half the error that would be obtained if the material was randomly assigning data to one class or another.

TABLE II
TRAINING AND VERIFICATION ERRORS FOR BUPA LIVER DISORDER AND MAMMOGRAPHIC MASS PROBLEMS.

MMC	$\Phi_e^*(\%)$	$\Phi_{e,v}^*(\%)$	$\Phi_{e,v}^w(\%)$	$\overline{\Phi}_{e,v}$ (%)
EiM, DE	20.5	18.85	21.39	20.51
DMNN, DE	15.8	N/A	N/A	10.40
Human	N/A	12.00	N/A	17.00
BPC	$\Phi_e^*(\%)$	$\Phi_{e,v}^*(\%)$	$\Phi_{e,v}^w(\%)$	$\overline{\Phi}_{e,v}$ (%)
EiM, DE	23.2	27.53	36.23	31.88
DMNN, DE	37.6	N/A	N/A	31.10
BPNN	N/A	20.29	39.14	30.00
GRNN	N/A	28.99	46.38	35.94
RBFNN	N/A	27.54	43.48	32.46
PNN	N/A	28.99	46.38	35.94
CMTNN	N/A	21.74	47.83	29.71

As mentioned in Section III, the MMC dataset was split into $K_t = 200$ and $K_v = 631$. Results for DE-trained DMNN were obtained using $K_t = 664$ and $K_v = 167$. The ratio of training to verification instances is nearly inverted. However, the evolved SWCNT/LC blend is able to produce a training error which is 4.3% higher, whilst the average verification error is 10.11% superior to the DE-trained DMNN. The best and average *in materio* solutions compare better to radiologists' opinion ($\overline{\Phi}_{e,v}$), with an error that is respectively 1.85% and 3.51% higher. The fellowship trained radiologists ($\Phi_{e,v}^*$), however, are more accurate in their diagnosis than both their non-fellowship trained counterpart and the SWCNT/LC.

In the case of the BPC problem, the dataset was split in the same way for all implementations. It can be seen in the third row of Table II that the evolved carbon nanotube-based device compares with the different NN implementations. Optimal training is lower than that obtained using DMNN and average verification marginally higher, with a 0.78% difference. Similarly, the average verification error obtained using BPNN and CMTNN are respectively 1.88% and 2.17% lower than the *in materio* result. The trained SWCNT/LC outperforms the other methods, GRNN, RBFNN and PNN, with a worst and average verification error that is always lower by a minimum of 0.58% and a maximum of 4.06%.

Interestingly, when trained for the more complex BPC problem, the SWCNT/LC blend produces results that are either better or at least comparable to those obtained with the different NN implementations. On the other hand, for the less complex MMC problem, results are less comparable.

B. Double Training Experiments

The second set of results presented have been obtained using the two artificial datasets, NLC and V1C. It has been observed in our previous work [21, 22] that an un-configured SWCNT/LC mixture can be trained to solve these binary classification problems. Current investigations focus on whether the SWCNT structures formed in the material during training are stable, a non-negligible contribution to the solution and finally if they can be retained, to some extent, when the material has been retrained for a new problem. The effect of problem complexity in the multiple material training is also investigated; does training first for a complex problem make solving a simpler one easier or is the opposite true ?

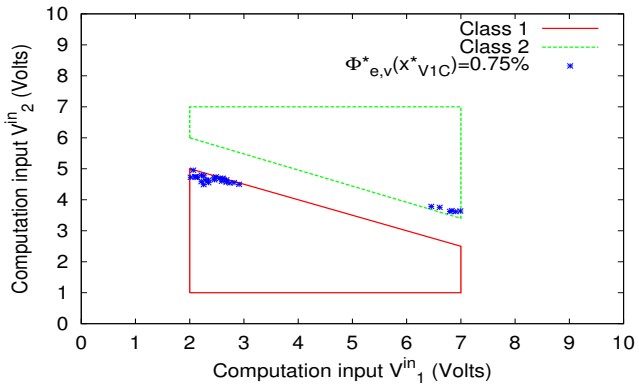
An experiment consists of a set of two consecutive training and verification procedures, using two different datasets, which are performed on the same material. The unconfigured material is drop-cast onto the micro-electrode array and subjected to a varying electric field produced by the DE algorithm until it is able to solve the first problem.

Following this first training, the evolved material is sent the corresponding verification data along with the optimum set of decision variables obtained during training $\mathbf{x}_1^*$. We thereby verify that the solution combining $\mathbf{x}_1^*$ and material state can generalise. In order to determine if the evolved material state itself is a non-negligible contribution to the solution, the sample is left for 300 seconds with no signals. A second verification test is then performed, where the configuration voltages are not applied. As seen in Section III, V1C is less complex than NLC, but the two problems are relatively similar. Verification data for the second problem is therefore sent along with $\mathbf{x}_1^*$ on the once trained material to determine if the solution is common to both problems, and if any further training would be redundant. If it is not the case, the SWCNT/LC material is subjected to a second set of training and verifications for the new dataset. The second problem's verification data are sent with new optimal set of decision variables, $\mathbf{x}_2^*$, then are re-sent with no configuration voltages.

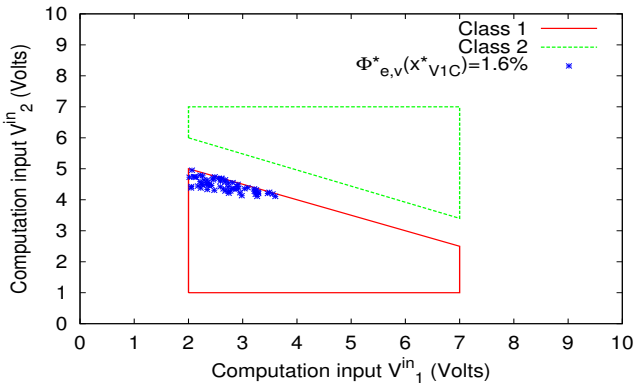
Finally, the ability of the doubly trained material to solve the original problem, with and without configuration voltages is assessed. It must be noted that each verification test is repeated ten times, with a 1s delay between each test. Results are presented in table III. The minimum training error is defined by $\overline{\Phi}_t$ and verification error by $\overline{\Phi}_{e,v}$. Values are averaged over three experiments. Distribution of the ill-classified verification data for the sequence of tests is presented in Figure 5.

Starting from an un-configured state, the DE algorithm is able, on average, to bring the SWCNT/LC mixture into a state where it is able to correctly assign 100% of $V1C$ training data and 98% of verification using the same set of decision variables, $\mathbf{x}_{V1C}^*$. There is no photographic evidence that the SWCNT structures have been retained after the first verification test. However, when V_j is set to zero and the verification data are sent to the trained material, the error increases by only 0.36%. From this difference, it can be observed on one hand that the contribution of the evolved material state is not negligible, and on the other, that the percolation paths have not relaxed to the original un-configured condition.

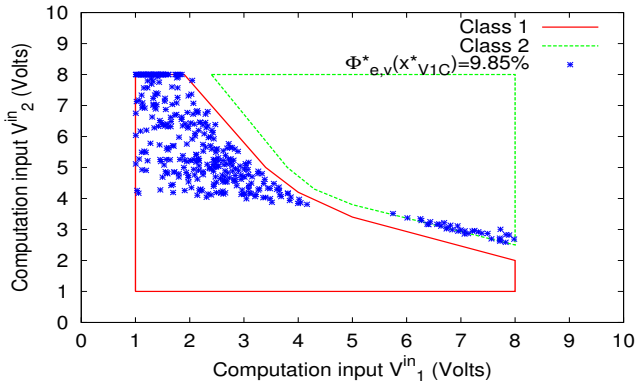
The result presented in the first row, fourth column, of Table III shows that a combination of material state and $\mathbf{x}_{V1C}^*$ does not generalise to the NLC verification dataset. The error increases by 12.18% and 14.54% compared to V1C, and to an optimum NLC solution, respectively. DE is used to subject the SWCNT/LC to a new training, using the NLC dataset. Results are near optimal, with $\overline{\Phi}_t = 0.0\%$ and $\overline{\Phi}_{e,v} = 0.2\%$. When no configuration voltages are applied the doubly trained material is able to classify the NLC verification dataset with an average of 21.75%. Finally, the verification results when V1C is reapplied have increased to become 7.85% and 9.28% which means that the original solution is not fully recovered. The second training has had an effect on the material state produced by the first training.



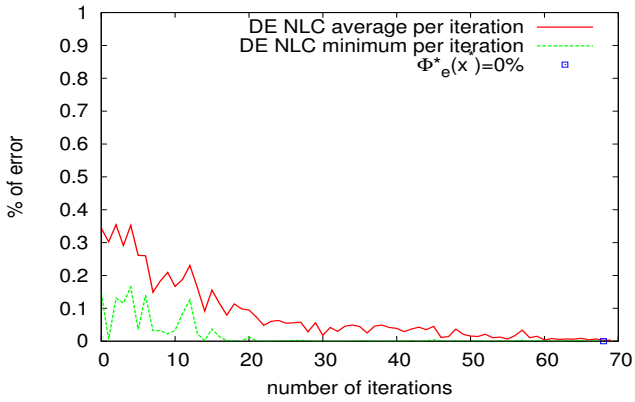
(a)



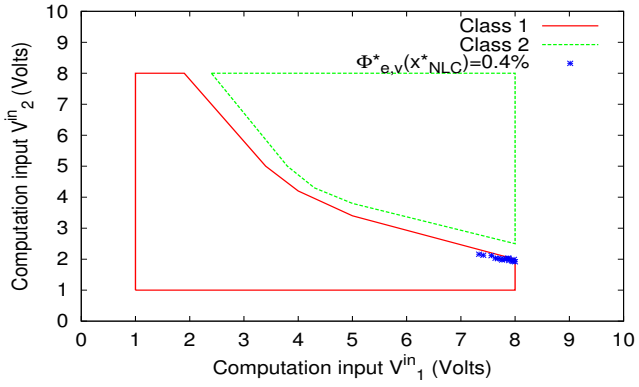
(b)



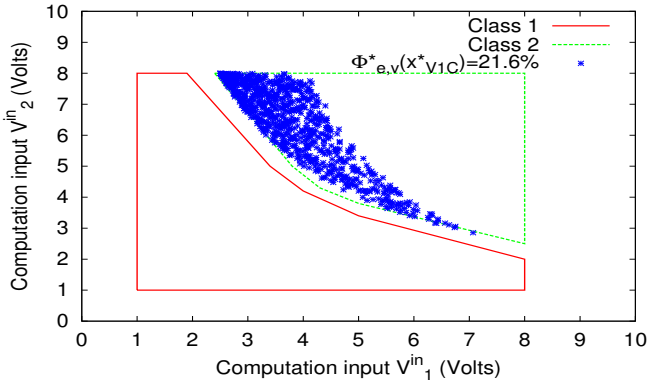
(c)



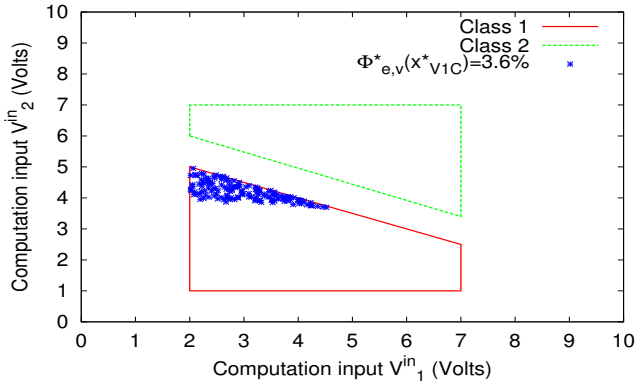
(d)



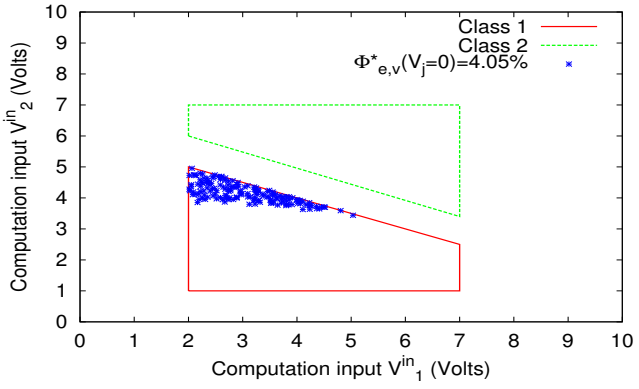
(e)



(f)



(g)



(h)

Fig. 5. Post first training, mapping of ill-classified verification data-points for (a) V1C, $\mathbf{x}^*_{V1C}$ applied (b) V1C, no configuration voltages (c) NLC, $\mathbf{x}^*_{V1C}$ applied. Second training, (d) NLC, convergence of the objective function and mapping of the ill-classified verification datapoints for (e) NLC, $\mathbf{x}^*_{NLC}$ applied. Finally ill-classified datapoints for (f) V1C, no configuration voltages (g) V1C, $\mathbf{x}^*_{V1C}$ applied and (h) V1C, no configuration voltages

TABLE III
AVERAGE TRAINING $\overline{\Phi}_t(\%)$ AND VERIFICATION $\overline{\Phi}_{e,v}(\%)$ ERROR FOR DOUBLE TRAINING EXPERIMENTS ON THE SAME SWCNT/LC SAMPLE.

Once trained material				Twice trained material				
V1C, $\overline{\Phi}_t$	V1C, $\mathbf{x}^*_{V1C}$	V1C, $V_j = 0$	NLC, $\mathbf{x}^*_{V1C}$	NLC, $\overline{\Phi}_t$	NLC, $\mathbf{x}^*_{NLC}$	NLC, $V_j = 0$	V1C, $\mathbf{x}^*_{V1C}$	V1C, $V_j = 0$
0.00	2.00	2.36	14.54	0.00	0.2	21.74	7.85	9.29
Once trained material				Twice trained material				
NLC, $\overline{\Phi}_t$	NLC, $\mathbf{x}^*_{NLC}$	NLC, $V_j = 0$	V1C, $\mathbf{x}^*_{NLC}$	V1C, $\overline{\Phi}_t$	V1C, $\mathbf{x}^*_{V1C}$	V1C, $V_j = 0$	NLC, $\mathbf{x}^*_{NLC}$	NLC, $V_j = 0$
0.5	1.19	2.35	10.76	0.00	0.00	0.76	3.39	5.78

From experiments where a simple binary diagonally separable dataset was used, followed by a more complex binary dataset with a hyperbolic separation, it can be observed that:
1) The contribution of the material state to the evolved classifier is not negligible.
2) The material can be retrained for a different, more complex problem, for which it obtains low error and
3) Important modifications in the material's state do not completely destroy original solutions. The untrained material produces a verification error for V1C, $\overline{\Phi}_{e,v}$= 49.53%, whilst the doubly trained material's solution achieves $\overline{\Phi}_{e,v}$ = 7.85%.

The second row of Table III presents results for the inverse experiment, where the hyperbola-separated dataset was used first, followed by its diagonally separated counterpart. First, DE is run on an un-configured SWCNT/LC sample, until it becomes possible to classify the NLC training data.

Verification tests with $\mathbf{x}^*_{NLC}$ produce an average of 1.19% error whilst verification increases to 2.35% when no configuration voltages are used. Sending $\mathbf{x}^*_{NLC}$ partly solves the V1C problem, but not well enough for further training to be redundant. Retraining the SWCNT/LC blend for the less complex V1C dataset results in 0% training and verification errors, with an increase of 0.76% when $V_j = 0$. The last results presented in Table III show that the NLC verification error has become 3.39% and 5.78% for $\mathbf{x}^*_{NLC}$ and $V_j = 0$ respectively.

It was observed that training for a simple problem first helped in finding very good solutions for the more complex problem. However, after finding a solution for the more complex problem, the state had change in such a way that the original solution had been partly destroyed, producing verification errors higher by 5.85% and 6.93% with and without configuration voltages respectively.

It was also possible to achieve good solutions for the more complex problem starting from an un-configured material. The difference in this case is that the second training had less effect on the original solution, for which the verification errors increased by 2.2% and 2.43% with and without configuration voltages respectively.

VI. CONCLUSION AND FUTURE WORK

This paper reported results on experimental investigations of an Evolution-in-Materio approach for the classification problem. The material is a mix of single-walled-carbon-nanotubes and liquid crystals. The method used does not follow conventional computation methods that have been proposed in the literature such as neural networks and K-nearest neighbours methods. Instead, a piece of material in liquid state is evolved until it reaches a computing inducing state where the computation task is a binary classification problem.

A first set of experiments was undertaken to assess the ability for a differential evolution algorithm to train the SWCNT/LC material to solve real life binary classification data. The mammographic mass and bupa liver disorder problems found on the UCI repository [10] were used. A training and verification process was followed and produced an evolved material which was able to classify the two datasets without any metal-oxide-field-effect-transistors components. Results obtained were compared to different types of neural network solutions found in literature [1, 8] as well as human diagnosis accuracy. In the case of the MMC dataset, the evolved carbon nanotube-based classifier produced average verification errors that were higher than both those obtained by human and neural networks. In contrast, results obtained for the BPC problem were more comparable with the neural network results.

The interpretation scheme and objective function used in the implementation were very simple. For example, the difference between true and false positives, an important parameter in medical applications, was not used in our problem formulation. The above considerations will be addressed in further investigations, along with modifying the split of the MMC dataset to be the same as that used in [1].

It has been observed that the contribution of the material state to the solution is not negligible. This had not been observed for simpler problems discussed in earlier papers [21, 22]. In addition, the material retains a memory of previous trainings. The problem complexity can be linked with this memory and the material state's contribution to a solution. Experiments indicate that after training for less complicated problems, the material can be retrained to address more complex ones. It is observed, however, that the second training tends to destroy the original solution. On the other hand, training the material for a complex problem first and then retraining it for a simpler one, results in a material state which is able to solve both problems well.

From these observations, we can formulate the following hypothesis; the SWCNT structures produced during DE training are relatively stable. When a more complex problem is used first, these structures favour the search for simpler problems, with little impairment to the original solution. Investigations using microscopy techniques are needed to confirm our hypothesis. It will be the focus of future work, along with the application of the retraining process to a larger number of datasets and to problems which present more differences than the ones used.

As discussed previously, the electrical and physical charac-

teristics of SWCNT-based mixtures are modified when subjected to EiM. These modifications are extremely difficult to model. A better understanding of the modification mechanism is crucial for an optimisation of the technique, if it is to become competitive as an alternative to conventional technology. We believe that this is an exciting research work on unconventional and evolvable computing systems.

REFERENCES

[1] F. Arce, E. Zamora, H. Sossa, and R. Barrón. Dendrite morphological neural networks trained by differential evolution. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, pages 1–8. IEEE, 2016.

[2] J. G. Elmore, S. L. Jackson, L. Abraham, D. L. Miglioretti, P. A. Carney, B. M. Geller, B. C. Yankaskas, K. Kerlikowske, T. Onega, R. D. Rosenberg, et al. Variability in interpretive performance at screening mammography and radiologists characteristics associated with accuracy 1. *Radiology*, 253(3):641–651, 2009.

[3] S. Harding and J. F. Miller. Evolution in materio: A tone discriminator in liquid crystal. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1800–1807. IEEE, 2004.

[4] S. Harding and J. F. Miller. Evolution in materio: Evolving logic gates in liquid crystal. *Proc. Eur. Conf. Artif. Life (ECAL 2005), Workshop on Unconventional Computing: From cellular automata to wetware*, pages 133–149, 2005.

[5] S. Harding and J. F. Miller. Evolution in materio: investigating the stability of robot controllers evolved in liquid crystal. In *Evolvable Systems: From Biology to Hardware*, pages 155–164. Springer, 2005.

[6] J. J. Heidelbaugh and M. Bruderly. Cirrhosis and chronic liver failure: part i. diagnosis and evaluation. *Am Fam Physician*, 74(5):756–62, 2006.

[7] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):289–300, 2002.

[8] P. Jeatrakul and K. Wong. Comparing the performance of different neural networks for binary classification problems. In *Natural Language Processing, 2009. SNLP'09. Eighth International Symposium on*, pages 111–115. IEEE, 2009.

[9] A. Kotsialos, M. K. Massey, F. Qaiser, D. Zeze, C. Pearson, and M. C. Petty. Logic gate and circuit training on randomly dispersed carbon nanotubes. *International Journal of Unconventional Computing*, 10(5-6):473–497, 2014.

[10] M. Lichman. UCI machine learning repository, 2013.

[11] M. Massey, A. Kotsialos, F. Qaiser, D. Zeze, C. Pearson, D. Volpati, L. Bowen, and M. Petty. Computing with carbon nanotubes: Optimization of threshold logic gates using disordered nanotube/polymer composites. *Journal of Applied Physics*, 117(13):134903, 2015.

[12] M. Massey, A. Kotsialos, D. Volpati, E. Vissol-Gaudin, C. Pearson, L. Bowen, B. Obara, D. Zeze, C. Groves, and M. Petty. Evolution of electronic circuits using carbon nanotube composites. *Scientific reports*, 6, 2016.

[13] J. F. Miller, S. L. Harding, and G. Tufte. Evolution-in-materio: evolving computation in materials. *Evolutionary Intelligence*, 7(1):49–67, 2014.

[14] M. Mohid, J. F. Miller, S. L. Harding, G. Tufte, O. R. Lykkebo, M. K. Massey, and M. C. Petty. Evolution-in-materio: Solving bin packing problems using materials. In *Evolvable Systems (ICES), 2014 IEEE International Conference on*, pages 38–45. IEEE, 2014.

[15] M. Mohid, J. F. Miller, S. L. Harding, G. Tufte, O. R. Lykkebø, M. K. Massey, and M. C. Petty. Evolution-in-materio: Solving machine learning classification problems using materials. In *Parallel Problem Solving from Nature–PPSN XIII*, pages 721–730. Springer, 2014.

[16] M. E. H. Pedersen. Good parameters for differential evolution. Technical report, Technical report, Hvass Computer Science Laboratories, 2010.

[17] F. Qaiser, A. Kotsialos, M. Massey, D. Zeze, C. Pearson, and M. Petty. Manipulating the conductance of single-walled carbon nanotubes based thin films for evolving threshold logic circuits using particle swarm optimisation. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 5255–5261. IEEE, 2016.

[18] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[19] National Cancer Institute. https://www.cancer.gov/types/breast/understanding-breast-changes, 2015.

[20] E. Vissol-Gaudin, A. Kotsialos, M. K. Massey, C. Groves, C. Pearson, D. A. Zeze, and M. C. Petty. Solving binary classification problems with carbon nanotube/liquid crystal composites and evolutionary algorithms. In *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017.

[21] E. Vissol-Gaudin, A. Kotsialos, M. K. Massey, D. Zeze, C. Pearson, C. Groves, and M. Petty. Data classification using carbon-nanotubes and evolutionary algorithms. In *International Conference on Parallel Problem Solving from Nature*, pages 644–654. Springer, 2016.

[22] E. Vissol-Gaudin, A. Kotsialos, M. K. Massey, D. A. Zeze, C. Pearson, C. Groves, and M. C. Petty. Training a carbon-nanotube/liquid crystal data classifier using evolutionary algorithms. In *Unconventional Computation and Natural Computation Conference*, Lecture Notes in Computing Sciences, pages 130–141. Springer, 2016.

[23] D. Volpati, M. Massey, D. Johnson, A. Kotsialos, F. Qaiser, C. Pearson, K. Coleman, G. Tiburzi, D. Zeze, and M. Petty. Exploring the alignment of carbon nanotubes dispersed in a liquid crystal matrix using coplanar electrodes. *Journal of Applied Physics*, 117(12):125303, 2015.