Solving Binary Classification Problems with Carbon Nanotube / Liquid Crystal Composites and Evolutionary Algorithms

E. Vissol-Gaudin, A. Kotsialos, M.K. Massey, 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
m.k.massey1@gmail.com

***Abstract*—This paper presents a series of experiments demonstrating the capacity of single-walled carbon-nanotube (SWCNT)/liquid crystal (LC) mixtures to be trained by evolutionary algorithms to act as classifiers on linear and nonlinear binary datasets. The training process is formulated as an optimisation problem with hardware in the loop. The liquid SWCNT/LC samples used here are un-configured and with nonlinear current-voltage relationship, thus presenting a potential for being evolved. The nature of the problem means that derivative-free stochastic search algorithms are required. Results presented here are based on differential evolution (DE) and particle swarm optimisation (PSO). Further investigations using DE, suggest that a SWCNT/LC material is capable of being reconfigured for different binary classification problems, corroborating previous research. In addition, it is able to retain a physical memory of each of the solutions to the problems it has been trained to solve.**

I. Introduction

Evolution-in-Materio (EiM) is a research area in the domain of unconventional computing (UC). EiM aims at finding alternative approaches for developing computational devices, exploiting intrinsic properties of materials by use of evolutionary algorithms (EAs) [14].

A similar notion was described by G. Pask experimenting on the evolution of an electrochemical ear [2]. Interest in EiM was revived after EAs were used for configuring field-programmable-gate-arrays (FPGA) into frequency classifiers [28], making implicit use of the FPGA's material properties for selecting solutions. Observations on the resulting circuits suggested that, in a search for an optimum solution an EA exploits the intrinsic analogue properties of the FPGA's components [28], [29].

EiM has demonstrated the possibility of creating devices where the information processing is done within a previously un-configured material, which does not contain any silicon or metal-oxide-field-effect-transistors (MOFSETs) and without resorting to the carefully designed components and architecture of conventional electronic circuits.

In general, EiM makes use of a computer, an evolvable motherboard and a suitable hardware interface between the two. The evolvable motherboard hosts the material whose physical properties are manipulated and evolved. The computer is used to run the EAs, and the hardware interface to translate signals between computer and material, which effectively acts as a black box. This black box combines the hardware interface and the material whose properties are exploited. There is no general model available for predicting the material's property given a set of incident inputs.

An iterative process is performed by the computer, gradually configuring the material until it reaches a state where a pre-specified scheme of interaction is uniquely translated as a computation input/output relationship. Configuration of the material is induced by a combination of incident signals which are either controlled by an EA through a combination of hardware and software or independently through the influence of the environment. In effect, the iterative process is the material training or evolution. A set of post training tests reusing the optimal evolved solution, which consists of *both* the optimal incident signals and the material's current state, are performed for verification purposes.

Training and verification require the selection of two distinct finite sets of data. Both consist of known input/output pairs from the computational problem's domain of definition and range, respectively. The training process requires the repetitive application of computation inputs sent to the material and measurement of its corresponding response. Measured responses are translated into computation outputs allowing for the definition of an error function. Specific physical properties of the material are measured for a given EiM implementation, which in our case are measurements of electrical current. The interpretation scheme of the materials response used for translating these measurements into a computation output are pre-specified and fully known *before* the training process starts.

As a whole concept, EiM has a broad scope and can be delineated along five inter-dependent dimensions: (1) the material used, (2) the hardware interface, (3) the type of problems to be solved, (4) the problem formulation and (5) the algorithms used for training.

Different organic and inorganic media have been used as materials, such as slime moulds [7], bacterial consortia [1],

978-1-5090-4601-0/17/$31.00

cells (neurons) [22], liquid crystal (LC) panels [5] and nano-particles [3]. Single-walled carbon-nanotube (SWCNT) based materials have shown the potential to solve computational problems [8], [12], [32], [15], [18], [23]. In [26] it is argued that non-biological materials make a better medium for un-conventional computing exploration. Following this argument, as well as results in [32], [11], [31], [30], a mixture of SWCNT and LC in liquid form is used here. The SWCNT/LC material has a non-linear relationship between voltage and current, which is exploited during EiM. When deposited on an electrode array, the structure of the SWCNT matrix changes under an applied electric field. Research suggests that this is due to the re-orientation of the SWCNT molecules in the direction of the field and the formation of physical percolation paths between electrodes formed by the SWCNTs [32], [11]. During the learning process, an EA controls voltage levels applied to the sample, thereby modifying the orientation of these paths and the material's conductivity.

The types of materials listed have a very complex structure and the development of analytical or stochastic models of their behaviour is very difficult. In their absence, they are treated as black boxes in the EiM process. Derivative free population based stochastic search algorithms are used for solving the training problem. Here, a particle swarm optimisation (PSO) [9] and an implementation of differential evolution (DE) [21], are used.

Several candidate computational problems can be used in the context of EiM. A more comprehensive review of potential problems can be found in [19]. The problem considered here is simple binary data classification with different degrees of separation and data distributions. Compared with the work reported in [11], [30], [31], here, a slightly modified objective function is used, where the material's response invariance is not penalised. Furthermore, results are presented for a new nonlinear artificial binary classification problem, referred to as NLC. Finally, an investigation of the properties of doubly–trained materials is introduced.

II. Hardware Implementation and Material

An EiM experiment comprises three communicating levels of hardware; a computer, an evolvable motherboard (EM) and the material itself. The computer is used to run an EA. Signals from which are sent via serial connection to a *mbed* microcontroller which is part of the EM. Those signals are converted into voltages by the *mbed*, via a set of digital-to-analogue converters (DACs), and transferred to a gold micro-electrode array. The material sample was drop-cast into a 2.5mm diameter nylon washer fixed on the electrode array.

Outputs from the material are current levels which are transferred back to the microcontroller via analogue-to-digital converters (ADCs). An interpretation scheme, described in section IV, is used to map the material's measurements to a uniquely defined computational output. This allows the material's computation result for a given set of inputs to be sent back to the computer, via the serial connection. A

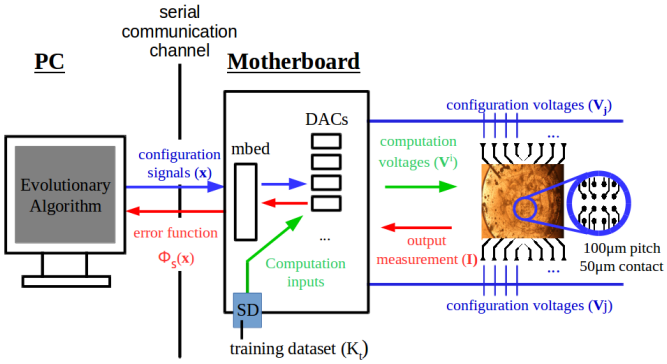
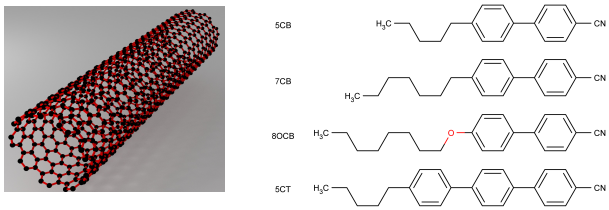


Fig. 1. EiM hardware implementation.



(a) (b)

Fig. 2. (a) Single-Walled-Carbon-Nanotube (b) E7 Liquid Crystal molecules.

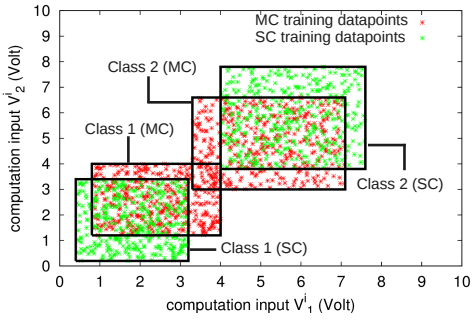
graphical representation of the implementation and process is presented in Figure 1.

A 0.05 weighted (wt) % SWCNT/LC material sample is produced by adding nanotubes (dry powder) to LCs and subsequently mixing with an ultrasonic probe. This ensures a near-uniform distribution of SWCNT within the medium. The SWCNT used contains 2/3 semiconducting (figure 2(a)), 1/3 metallic nanotubes and 15 % impurities. A blend of four LC molecules presented in 2(b) constitute the E7 nematic LC used here. The SWCNTs and LC were purchased from Carbon Nanotechnologies Inc. (Houston,TX, USA) and Merk Japan, respectively.

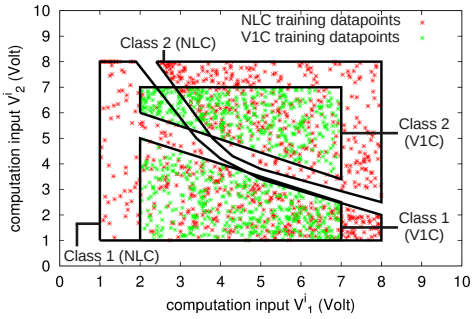
It has been observed in [32] that SWCNTs dispersed in LC tend to aggregate in line with an electric field, forming percolation paths between electrodes. LCs are used as a solvent in the material, as LC-only samples present very little conductivity [32] and the variation of their outputs under the EiM process cannot be interpreted as a computation [30], [31].

The purpose of LCs is therefore to provide a medium within which SWCNTs are able to form variable and reconfigurable percolation paths resulting in complex electrical networks as argued in [11]. This adds an extra level of complexity and reconfigurability compared to the experiments performed on solid SWCNT composites, [8], [12], [18], [17], [16], [23].

Prior to training, a material sample is not configured. Contrary to conventional electronics design methods, the whole system is not designed *a priori* to perform a specific operation. Instead of designing circuits with specific components for solving the problem, the classification operation is performed by iteratively modifying the un-configured material's state.



(a)



(b)

Fig. 3. Four binary classification problems represented by (a) separable and merged classes (b) diagonally and non-linearly separable classes.

At each iteration of an evolutionary optimisation algorithm, the material's internal structure is changed and its electrical properties adapted in order to favour the interpretation of the measurements as a computation based on the pre-specified interpretation scheme.

III. Classification Datasets

The experiments presented in this paper have focused on simple binary classification problems. Such problems have a number of applications and are typically researched within the domain of machine learning. Applications include medicine [6], [25], [13], meteorology [6], [10], [20] and security [24]. The problems considered here, however, were artificially created and are intentionally simple, as they are used as proof-of-concept for the technique.

Each of those artificial problems is defined by a number of training instances K_{tot} and attributes. For our investigations, $K_{tot} = 4,800$ and each class datum is a 2-dimensional point whose coordinates are electrical voltages. Following a typical training and verification scheme, the instances are divided into a training set, containing $K_t = 800$ members and a verification set containing $K_v = 4,000$ members. Figure 3 is a graphical representation of the problems' data classes. A class point is defined by two voltage levels, V_1^{in} and V_2^{in} and belongs to either class 1 ($C((V_1^{in}, V_2^{in})) = C_1$) or class 2 ($C((V_1^{in}, V_2^{in})) = C_2$). Training and verification data are distributed randomly within each class's boundary (in solid lines).

The problems shown in Fig. 3 are of increasing difficulty. The datasets of two similar problems are shown in 3(a); the first one, SC, is linear with two separable classes easily discerned; the second, MC, is similar but with more uncertainty in the classes' data sets, since there in an overlap between the two over an area where they are partially merged. The overlapping area contains around 6.6% of the total number of data points.

Two other problem data sets are shown in Figure 3(b). A linear problem where classes are separated diagonally resulting to the V1C dataset and a nonlinear separable problem labelled NLC, where the classes are separated by a curve, rather than a straight line.

IV. Problem Formulation and Implementation

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.

By considering as a decision variable only the possible locations where the two components of $\mathbf{V}^{in}$ are applied and 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: $V_1 \rightarrow 1$, $V_2 \rightarrow 2$, $V_3 \rightarrow 4$ $V_4 \rightarrow 6$, $V_5 \rightarrow 7$, $V_6 \rightarrow 8$, $V_7 \rightarrow 9$ $V_8 \rightarrow 10$) 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 is defined as

$$\mathbf{x} = [V_1 \ldots V_8 \ R \ p]^T \tag{1}$$

where R is a scaling factor. It is for a specific electrode assignment p and set of configuration voltages V_j, that the material's response to an input $\mathbf{V}^{in}$ is recorded. The response is a pair of measurements $\mathbf{I} = (I_1, I_2)$ (A) of the direct current at the two output terminals, which are 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. They can be based on an arbitrary but suitable combination of the computational and configuration inputs and

the corresponding induced material responses. A functional form of $C_M\left(\mathbf{V}^{in}, \mathbf{x}\right)$ must 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.

For the SC and MC problems

$$C_M\left(\mathbf{V}^{in}(k), \mathbf{x}\right) = \begin{cases} C_1 & \text{if} \quad I_1(k) > RI_2(k) \\ C_2 & \text{if} \quad I_1(k) \leq RI_2(k). \end{cases} \tag{2}$$

For the V1C and NLC problems

$$C_M(\mathbf{V}^{in}(k), \mathbf{x}) = \begin{cases} C_1 & \text{if} \quad I_1(k)V_1^{in}(k) + I_2(k)V_2^{in}(k) > R \\ C_2 & \text{if} \quad I_1(k)V_1^{in}(k) + I_2(k)V_2^{in}(k) \leq R \end{cases} \tag{3}$$

For every training data point $\mathbf{V}^{in}(k)$, $k = 1, \ldots, K_t$ the error from translating the material response according to rules (2) or (3) is

$$\epsilon_{\mathbf{x}}(k) = \begin{cases} 0 & \text{if} \quad C_M(\mathbf{V}^{in}(k), \mathbf{x}) = C(\mathbf{V}^{in}(k)) \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

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

$$\Phi_e(\mathbf{x}) = \frac{1}{K_t} \sum_{k=1}^{K_t} \epsilon_{\mathbf{x}}(k). \tag{5}$$

In order to avoid the application of large voltages, which may disrupt the material structure formed by evolution, a penalty term $H(\mathbf{x})$ is added to (5), given by

$$H(\mathbf{x}) = \frac{\sum_{j=1}^{8} V_j^2}{8V_{max}^2}. \tag{6}$$

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 favourable to the problem formed during evolution.

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

$$\Phi_s(\mathbf{x}) = \Phi_e(\mathbf{x}) + H(\mathbf{x}) \tag{7}$$

The optimisation training problem to be solved is that of minimising (7) for a population of size S, subject to voltage bound constraints $V_j \in [V_{\min}, V_{\max}]$, $R > 0$, electrode assignment p and classification rule (2) or (3). $V_{\min} = 0$ Volts and for all problems $V_{\max} = 4$ Volts.

Two different stochastic optimisation algorithms are used for solving this problem, differential evolution (DE) [27] and particle swarm optimisation (PSO) [4]. A constricted version of PSO with parameters taken from [9] is implemented. The DE algorithm implementation uses the parameters suggested in [21]. A population size of $S = 10$ is used for both.

V. Result and Discussion

A number of experiments have been conducted for training material samples using the experimental apparatus shown in Figure 1. Except if otherwise stated, the SWCNT/LC solution drop-casted onto the washer is replaced between each experiment.

TABLE I
Training and verification errors for SC, MC, V1C and NL problems.

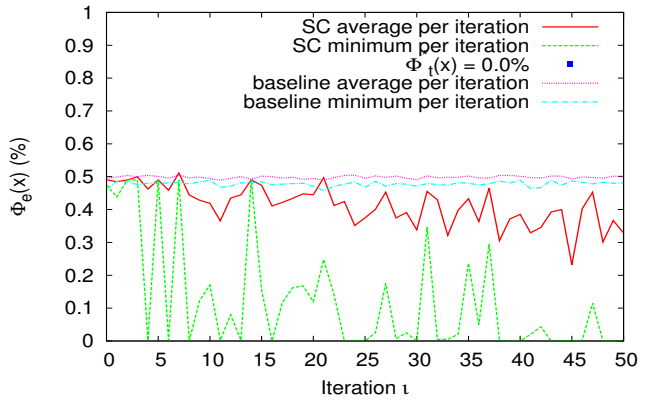
SC	$\Phi_e^*(\%)$	$\Phi_{e,v}^*(\%)$	$\Phi_{e,v}^w(\%)$	$\overline{\Phi}_{e,v}$ (%)	$\sigma^2_{\Phi_{e,v}}$
DE SC	0.000	0.000	0.125	0.0525	0.0013
PSO SC	0.000	0.525	1.050	0.8325	0.0225
MC	$\Phi_e^*(\%)$	$\Phi_{e,v}^*(\%)$	$\Phi_{e,v}^w(\%)$	$\overline{\Phi}_{e,v}$ (%)	$\sigma^2_{\Phi_{e,v}}$
DE MC	3.1	3.95	4.15	4.01	0.0044
PSO MC	5.7	7.825	9.025	8.5975	0.1184
V1C	$\Phi_e^*(\%)$	$\Phi_{e,v}^*(\%)$	$\Phi_{e,v}^w(\%)$	$\overline{\Phi}_{e,v}$ (%)	$\sigma^2_{\Phi_{e,v}}$
DE V1C	0.000	0.30	0.825	0.61	0.0285
PSO V1C	1.1	2.525	3.375	2.915	0.063
NL	$\Phi_e^*(\%)$	$\Phi_{e,v}^*(\%)$	$\Phi_{e,v}^w(\%)$	$\overline{\Phi}_{e,v}$ (%)	$\sigma^2_{\Phi_{e,v}}$
DE NLC	0.300	0.85	1.800	1.2325	0.1151
PSO NLC	0.6	4.275	7.325	6.06	0.8451

Baseline experiments were performed for MC, SC, V1C and NLC datasets prior to any other experiment. They consist of the same training and verification procedure performed on LC-only materials and on an array of fixed resistors. The minimum and average training and verification errors in these two cases are about 50%, i.e. the optimisation does not have an effect on conductive networks and on SWCNT–free LC samples. For these experiments, evolution of the error function's value during training is shown in left column subfigures of Figure 4.

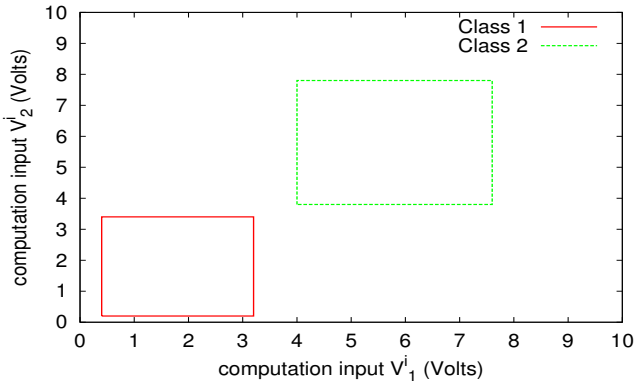
The first column of Table I presents the minimum error Φ_e^* achieved during training. Once training is terminated, verification is performed on the trained material by applying back the optimal solution achieved along with the previously unused verification data. The same verification procedure is repeated ten times and the outcome is given from the rest four columns of Table I. $\Phi_{e,v}^*$ is the error of the best verification result, $\Phi_{e,v}^w$ the worst, $\overline{\Phi}_{e,v}$ the average and $\sigma_{\Phi_{e,v}}$ is the variance. For both DE and PSO, the penalty term $H(\mathbf{x})$ is not included for the sake of brevity. Although the material is liquid and each sample presents differences, these results are reproducible.

It can be seen in Table I that in the case of the SC problem, both DE and PSO manage to bring the training error to zero. However, when the verification data set is used, i.e. when the material is tested against $K_v = 4,000$ points, a zero error occurs only for the solution found by the DE algorithm. As the mean and the variance indicate, not all verification runs yield a zero error. However, it is clear the DE solution is better than the one provided by PSO, as the average error is 0.0525 % whereas for PSO it is 0.8325 %. Variations between training and verification error exist, since the optimum is not always obtained at the end of the training process. Fig. 4 (a) shows the convergence trajectory for the best SC result, yielding $\Phi_e^* = 0$. The best objective function value achieved by the population as well as its average per iteration are shown along with the problem's baseline.

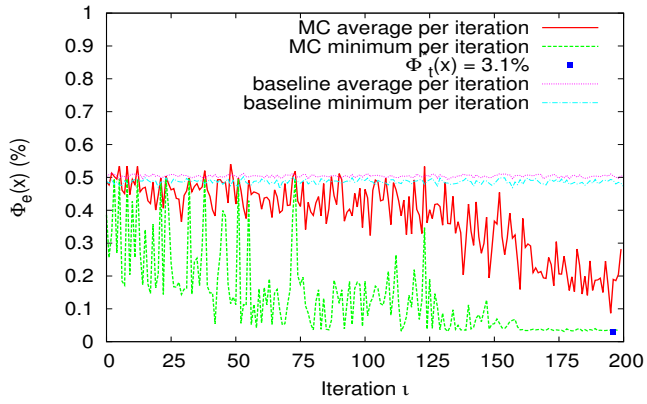
It can be seen that the optimal configuration voltages were



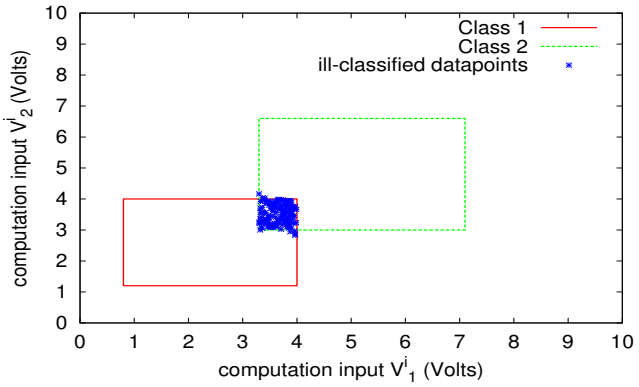
(a)



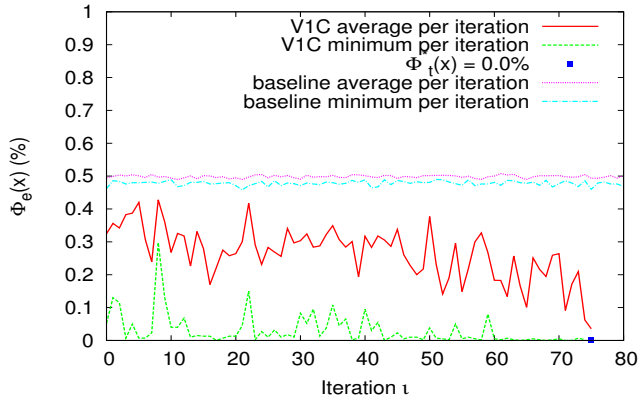
(b)



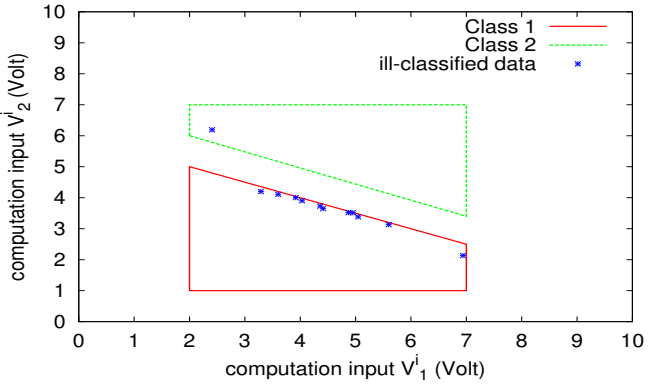
(c)



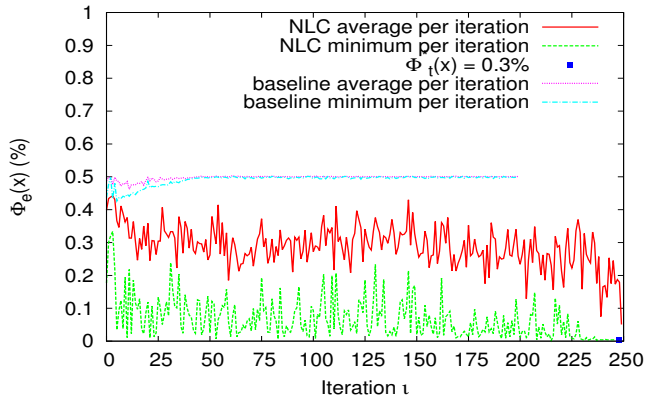
(d)



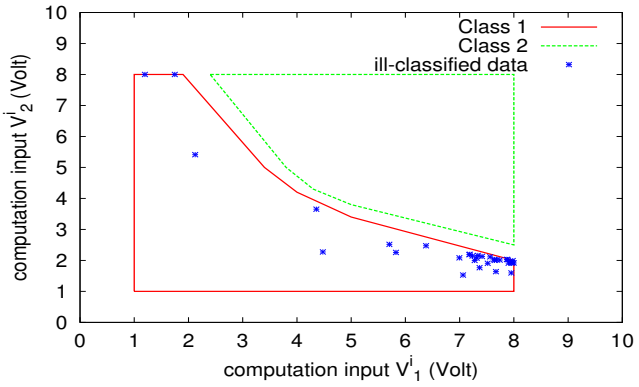
(e)



(f)



(g)



(h)

Fig. 4. (a) Convergence of the objective function - SC (b) Mapping of the ill-classified verification datapoints - SC (c) Convergence of the objective function - MC (d) Mapping of the ill-classified verification datapoints - MC (e) Convergence of the objective function - V1C (f) Mapping of the ill-classified verification datapoints - V1C (g) Convergence of the objective function - NLC (h) Mapping of the ill-classified verification datapoints - NLC

found by the end of iteration 50. This is the point where the confluence of the effect of the configuration voltages and the material's evolved structure result to zero error. After training finishes, the application of the optimal configuration voltages allow the material to act as a classifier. Fig. 4 (b) shows there are no ill-classified data points from the verification set shown in Fig. 3(a).

Figures 4 (c) and (d) show the objective function trajectory during training over the optimisation iterations and the ill-classified data points for the best of the verification runs of the MC problem, respectively. As expected, ill-classified points belong to the overlapping area of the classes, since it contains data points that are not distinguishable by design. This area contains about 6.6 % of the total data points, hence a fair random choice between the two will yield a 3.3 % error. This is the order of the training error achieved as shown in Table I. A similar verification error is achieved only by DE's solution, whereas the PSO's does not achieve as good a generalisation.

The V1C problem is a different version of SC, where the separation between classes is a small linear band at a constant angle, Fig. 3(b), resulting to diagonally arranged classes. Thus, C_1 cannot simply be defined as the class where the smaller amount of energy is sent to the material. This presents a more difficult challenge to the training algorithms, since the same amount of energy can be produced from points belonging to different classes . Table I shows that DE trains the material ending to a solution where all $K_t = 800$ points are classified correctly, i.e. $\Phi_e^* = 0.0\%$; the trajectory of $\overline{\Phi}_e$ and Φ_e^* over the number of iterations are shown in Figure 4 (e). The PSO algorithm does not achieve a zero training error, which is at 1.1%. The mean verification error for DE is 0.61% and for PSO 2.915% showing that the former consistently outperforms the latter. At any rate, although V1C is more difficult than SC and MC, it is still a linear problem with clearly defined separate classes.

A nonlinear problem with still separable classes is NLC, whose dataset is shown in Figure 3 (b), together with the V1C classes. The difference is that instead of having a linear band separating the classes, the separation band follows a hyperbolic curve. The training algorithms do not achieve a zero error any more. DE converges to a solution that yields $\Phi_e^* = 0.3\%$ outperforming PSO, which achieves $\Phi_e^* = 0.6\%$. The mean verification error is 1.2325% for DE and 6.06% for PSO. Fig. 4 (g) shows the convergence of the best and average Φ_e over the search iterations and 4 (h) the ill-classified data for the best verification result. It is the stochastic search and the material's behaviour that dictate the converge properties, which does not guarantee a stable global minimum.

Comparing the verification variance for each problem for DE and PSO, leads to the conclusion that DE converges to solutions that are more stable and generalise better than PSO solutions. This is partially conflicting with the results reported in [30], where DE yielded better training error but its solutions did not generalise as well as the ones provided by PSO. In this set of experiments, DE has achieved better generalisation as well, but it should be noted that more emphasis has been given to tuning DE in this round of experimentation.

VI. Retraining of materials

Currently there is only a very general understanding of the material's mechanism of adaptation for morphing into a state where the combination of configuration voltages and material state result to the input's classification. The nanotubes form bundles and percolation paths, ending in a very complex network of conducting and semi-conducting elements within the LC solution. The measured current at the two output terminals is the combined outcome of the material state and the optimal configuration voltages. The training process forms the nanotube network so as to favour the interpretation of resulting measurements for the classification problem. Hence, the question of interest is whether this property gained by the application of the optimisation process is retained in its structure. In other words, the question is whether the material retains a memory of its trainings. In order to investigate this, two-round training and verification experiments on the same material were conducted.

In the first experiment, an un-configured and un-trained sample was subjected to training for the SC problem. Let $V_{j,SC}^*$ denote the optimal configuration voltages. When training is finished, the verification procedure is repeated ten times and the mean verification error is 1.0%, Table II (different from the corresponding error in the round of experiments in Table I). Next, the same trained sample was sent the set of $V_{j,SC}^*$, i.e. the optimal set of configuration voltages for the SC problem, along with the $4,000$ instances of the MC verification data set. Ten such tests were repeated consecutively and the mean verification error was 50.125%. Hence, the material was acting as a fair coin experiment with respect to the MC when it is in the optimal state for SC and has not yet been trained for the MC problem. After that, the same sample was retrained using the MC data set and the optimisation converged to the optimal configuration voltages $V_{j,MC}^*$. Using $V_{j,MC}^*$ the MC verification data set was sent to the material for another set of verification tests. The average verification test error in this case was reduced to 6.255%, which is nearly double the 3.3% expected from a good solution for the MC problem. Having trained the material twice, the next step was to re-send $V_{j,SC}^*$ along with the SC verification data set to the doubly trained material, with no further training. The result was an average verification error equal to 0.2325%, i.e. an improvement over the original SC training, Table II. This improvement is achieved using the same configuration voltages that yielded an average error of 1.0%. Therefore the improvement observed should be attributed to the changes in the material structure forced by the second training.

In the second experiment, the SC and MC data sets were used again, but in reverse. A new un-trained sample was used for the MC problem resulting to optimal configuration voltages $V_{j,MC}^*$, different than the ones of the first experiment, and the average verification error achieved was 4.10%, which is slightly over the 3.3% lower bound. Sending as input the SC verification data in this state resulted to $\overline{\Phi}_{e,v} = 49.475\%$,

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

1^{st} training SC, $V^*_{j,SC}$	1^{st} verification MC, $V^*_{j,SC}$	2^{nd} training MC, $V^*_{j,MC}$	2^{nd} verification SC, $V^*_{j,SC}$
1.00	50.125	6.255	0.2325
1^{st} training MC, $V^*_{j,MC}$	1^{st} verification SC, $V^*_{j,MC}$	2^{nd} training SC, $V^*_{j,SC}$	2^{nd} verification MC, $V^*_{j,MC}$
4.10	49.475	0.0375	3.91
1^{st} training SC, $V^*_{j,SC}$	1^{st} verification NLC, $V^*_{j,SC}$	2^{nd} training NLC, $V^*_{j,NLC}$	2^{nd} verification SC, $V^*_{j,SC}$
0.7437	42.9337	0.5912	4.3225

which again means that in that state the material acts like a fair coin toss for SC. Re-training the material for SC results to an average verification error equal to 0.0375%, which is quite low and comparable to the results of Table I. Applying $V^*_{j,MC}$ and sending as input the MC verification data set, results to $\overline{\Phi}_{e,v} = 3.91\%$.

Given this marginal improvement on the MC, it can be supported that the original structure of the material formed while training for the first time on the MC has not changed significantly; the changes forced by the second training did not alter the material's behaviour radically, and where sufficient for making the whole system to behave as a classifier for SC as well.

The MC problem can be viewed as a superset of the SC, in the sense of the difficulty of the classification task. Hence, training the material to address a more complicated problem enables it to solve a simpler one. This is corroborated by the first experiment which resulted to improved performance for SC once it was retrained for MC. In the second experiment, the improvement of MC after retraining for SC was not in the same order as for SC in the first.

This is further supported by the result of the third, and final, experiment, shown in Table II, where the average verification error values for the SC and NLC problems are shown. Starting from the SC, the material and optimal configuration voltages converge to a very low average verification error. The material at that state and configuration voltages acts as a fair coin toss for the NLC data. Once retrained for NLC, the material is brought to a state where the application of the corresponding optimal configuration voltages $V^*_{j,NLC}$ result to an average verification error that is lower than 1%. Applying the optimal $V^*_{j,SC}$ to the retrained material and sending back the SC verification data set, results to an increased average, indicating that the material has undergone significant changes in order to be able to cope with the nonlinear behaviour of NLC, destroying in the process the original SC favouring structure.

VII. CONCLUSIONS

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-Nano-Tubes 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 towards a computing inducing state where the computation task is a binary classification problem.

Four different artificial data sets provided the background for problems of increasing difficulty. A typical training and verification experimental process was used for each problem and the results were reported. It is shown that the method followed is capable of evolving a binary classifier device out of an amorphous but evolvable piece of material, without using special and bespoke components such as MOFSETs.

Use of real life binary classification data sets is the immediate next step of our research. More experiments are under way and more detailed investigations will be conducted in an effort to produce more experimental results on real-life data such as those found in the UCI repository [10].

The investigations have shown that the material retains a memory of previous trainings. The problem's difficulty is associated with the effects of this memory. Preliminary experiments indicate that material structures resulting after training for less complicated problems are not sufficient for addressing more complex ones. Training the material for a difficult problem first and then retraining it for a simpler one, however, does not destroy the original solution obtained for the more challenging problem, leading this way to the conclusion that the material system does have a memory property. Future work will focus on studying the effect of retraining and on obtaining more conclusive results regarding the material's memory.

Another line of investigation will focus on understanding the contribution of each of the two components of a solution, i.e. the material state and the optimal configuration voltages, to the performance of the material as a classifier device.

Finding a method for "freezing" the liquid material at the computation inducing state brought by the evolution process, thereby creating a solid and consequently more stable material with the desired properties, is also under investigation.

As discussed previously, the electrical and physical characteristics 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 and we are just at the beginning of this investigation.

REFERENCES

[1] M. Amos, I. M. Axmann, N. Blüthgen, F. de la Cruz, A. Jaramillo, A. Rodriguez-Paton, and F. Simmel. Bacterial computing with engineered populations. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 373, 2015.

[2] J. Bird and E. Di Paolo. Gordon pask and his maverick machines. In P. Husbands, O. Holland, and M. Wheeler, editors, *The mechanical mind in history*, chapter 8, pages 185–211. The MIT Press, 2008.

[3] S. Bose, C. Lawrence, Z. Liu, K. Makarenko, R. van Damme, H. Broersma, and W. van der Wiel. Evolution of a designless nanoparticle network into reconfigurable boolean logic. *Nature nanotechnology*, 2015.

[4] R. C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, volume 1, pages 39–43. New York, NY, 1995.

[5] S. L. Harding and J. F. Miller. Evolution in materio: Computing with liquid crystal. *Journal of Unconventional Computing*, 3(4):243–257, 2007.

[6] 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.

[7] J. Jones, J. G. Whiting, and A. Adamatzky. Quantitative transformation for implementation of adder circuits in physical systems. *Biosystems*, 134:16–23, 2015.

[8] A. Kotsialos, M. K. Massey, F. Qaiser, D. A. 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.

[9] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis. Particle swarm optimization for integer programming. In *2002 IEEE Congress on Evolutionary Computation (CEC)*, volume 2, pages 1582–1587. IEEE, 2002.

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

[11] 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.

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

[13] L. Mena and J. A. Gonzales. Symbolic one-class learning from imbalanced datasets: Application in medical diagnosis. *International Journal on Artificial Intelligence Tools*, 18(02):273–309, 2009.

[14] J. F. Miller and K. Downing. Evolution in materio: Looking beyond the silicon box. In *Proceedings of the 2002 NASA/DoD Conference on Evolvable Hardware*, pages 167–176. IEEE, 2002.

[15] J. F. Miller and M. Mohid. Function optimization using cartesian genetic programming. In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, pages 147–148. ACM, 2013.

[16] M. Mohid, J. F. Miller, S. L. Harding, G. Tufte, O. R. Lykkebø, M. K. Massey, and M. C. Petty. Evolution-in-materio: A frequency classifier using materials. In *Evolvable Systems (ICES), 2014 IEEE International Conference on*, pages 46–53. IEEE, 2014.

[17] M. Mohid, J. F. Miller, S. L. Harding, G. Tufte, O. R. Lykkebø, 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.

[18] 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.

[19] NASCENCE project (ICT 317662). Report on suitable computational tasks of various difficulties, 2013. Deliverable D4.2.

[20] R. Nayak, P. S. Patheja, and A. Waoo. *An Enhanced Approach for Weather Forecasting Using Neural Network*, pages 833–839. Springer India, India, 2012.

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

[22] S. Prasad, M. Yang, X. Zhang, C. S. Ozkan, and M. Ozkan. Electric field assisted patterning of neuronal networks for the study of brain functions. *Biomedical Microdevices*, 5(2):125–137, 2003.

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

[24] J.-M. Ramirez-Cortes, P. Gomez-Gil, V. Alarcon-Aquino, D. Baez-Lopez, and R. Enriquez-Caldera. A biometric system based on neural networks and svm using morphological feature extraction from hand-shape images. *Informatica*, 22(2):225–240, 2011.

[25] M. Srinivas, R. Bharath, P. Rajalakshmi, and C. K. Mohan. Multi-level classification: A generic classification method for medical datasets. In *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*, pages 262–267. IEEE, 2015.

[26] S. Stepney. The neglected pillar of material computation. *Physica D: Nonlinear Phenomena*, 237(9):1157–1164, 2008.

[27] 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.

[28] A. Thompson. An evolved circuit, intrinsic in silicon, entwined with physics. In *Evolvable systems: from biology to hardware*, pages 390–405. Springer Berlin Heidelberg, 1996.

[29] A. Thompson and P. Layzell. Analysis of unconventional evolved electronics. *Communications of the ACM*, 42(4):71–79, 1999.

[30] E. Vissol-Gaudin, A. Kotsialos, M. K. Massey, D. A. Zeze, C. Pearson, C. Groves, and M. C. Petty. Data classification using carbon-nanotubes and evolutionary algorithms. In *International Conference on Parallel Problem Solving from Nature*, volume 9921 of *Lecture Notes in Computer Science*, pages 644–654. Springer, 2016.

[31] 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, UCNC 2016*, volume 9726 of *Lecture Notes in Computing Sciences*, pages 130–141. Springer, 2016.

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