Confidence Measures for Carbon-Nanotube / Liquid Crystals Classifiers

E. Vissol-Gaudin, A. Kotsialos, C. Groves, C. Pearson, D.A. Zeze, M.C. Petty
Department of Engineering, 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
N. Al Moubayed
Department of Computer Sciences, Durham University, Durham, United Kingdom
E-mail: noura.al-moubayed@durham.ac.uk

Abstract—**This paper focuses on a performance analysis of single-walled-carbon-nanotube / liquid crystal classifiers produced by evolution in materio. A new confidence measure is proposed in this paper. It is different from statistical tools commonly used to evaluate the performance of classifiers in that it is based on physical quantities extracted from the composite and related to its state. Using this measure, it is confirmed that in an un-trained state, ie: before being subjected to an algorithm-controlled evolution, the carbon-nanotube-based composites classify data at random. The training, or evolution, process brings these composites into a state where the classification is no longer random. Instead, the classifiers generalise well to unseen data and the classification accuracy remains stable across tests. The confidence measure associated with the resulting classifier's accuracy is relatively high at the classes' boundaries, which is consistent with the problem formulation.**

I. Introduction

One direction of research within the framework of Unconventional Computing (UC) methods is Evolution in Materio (EiM) [1].The latter combines software and hardware -based investigations, with the aims of exploring and exploiting the computational properties of materials using evolutionary algorithms (EAs). Contrary to traditional computing with metal-oxide-silicon-field-effect-transistor (MOFSET) technology, where everything is designed, produced and programmed very carefully, EiM uses a bottom-up approach where computation is performed by the material without having explicit knowledge of its internal properties.

This idea has been explored in the work of G. Pask [2] which was concerned with evolving an electrochemical ear. The field was revived following experiments reported in [3] where EAs were used to design a frequency classifying circuit in a field-programmable-gate-arrays (FPGAs). It was observed that the resulting circuit topologies chosen by the EAs had been influenced by some inherent analogue properties of the material used in the FPGA's components. EiM has replaced the FPGAs with suitable material systems.

In a typical EiM experiment, a search algorithm selects a set of configuration signals to be applied to the material, with the aim of changing its physical properties. The state the material is brought in is tested against a number of known input/output pairs. A response is recorded for each of those test inputs and a global error function is evaluated. The algorithm manipulates the state of the material within an iterative search, varying the values of the configuration signals. This evolutionary process brings the material into a state where it can perform the desired computation, in the sense that its outputs can be interpreted according to a pre-specified scheme.

EiM has a broad scope, and can be delineated along four dimensions: (a) the type of material used, (b) the physical property manipulated for obtaining a computation, (c) the computation problem and (d) the evolutionary algorithm used for solving the corresponding training problem.

Different materials have been used, biological and non-biological. Examples include slime moulds [4], bacterial consortia [5] and cells (neurons) [6]. In [7] it is argued that non-biological materials make a better medium for unconventional computing exploration. 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 [8], a tone discrimination device [9] and logic gates [10]. In [11] and [12] 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 [13] and [14] for solving optimisation problems. A mixture of SWCNT and LC in liquid form has also shown potential to solve classification problems. 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 three artificially created datasets. Due to the lack of stochastic and numerical model of the material's behaviour a population based derivative free stochastic algorithms, differential evolution [15] is used.

The main contribution of this paper is to propose a novel approach to analyse the performance of the physical SWCNT-based classifiers resulting from the EiM experiments. It is observed that a confidence measure can be calculated, based exclusively on the solution produced by the evolutionary algorithm, that is, a combination of the material's electrical state and a threshold that belongs to the problem's set of decision variables. This is an important contribution in that it will allow further investigations to be undertaken with complex datasets where a high confidence in the result is crucial. For such problems, the measure can be calculated throughout the training process and included in the problem formulation to be optimised along with the classifier accuracy.

A short description of the physical parts of the experimental set-up is presented in Sections II and III. The three datasets used in experiments are described in Section IV, whilst the problem formulation and algorithm parameters are detailed in Section V. The new confidence measure is proposed in Section VI and the analysis of results using this tool is reported in Section VII. Finally Section VIII concludes this paper and proposed avenues for future work.

II. Single-Walled-Carbon-NanoTube/Liquid Crystal Composite

The material used in experiments is a 0.05 wt % single-walled-carbon-nanotubes (SWCNT) / liquid crystals (LC) composite. The composite is produced by mixing nanotubes in dry powder form with liquid crystals using an ultrasonic probe. The powder was purchased from Carbon Nanotechnologies Inc. (Houston,TX, USA). It contains 2/3 semiconducting and 1/3 metallic nanotubes and less than 15% impurity. The E7 nematic liquid crystal blend was purchased from Merk Japan. Figure 1 presents a general structure of SWCNT and the four molecules contained in the E7 blend.

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 [16] that SWCNTs dispersed in LCs tend to aggregate along an applied electric field, forming percolation paths between electrodes. Varying this electric field results in modification of these percolation paths and the formation of complex SWCNT structures in the composite which favour computation [17, 18]. At the start of each experiment, a sample of the SWCNT/LC composite was drop-cast within a nylon washer of 2.5 mm internal diameter. The washer was fixed on a microscope slide upon which an array of gold electrodes had previously been deposited using etch-back photolithography. The left-hand-side of Figure 1 presents the micro-electrode array's design and its scale.

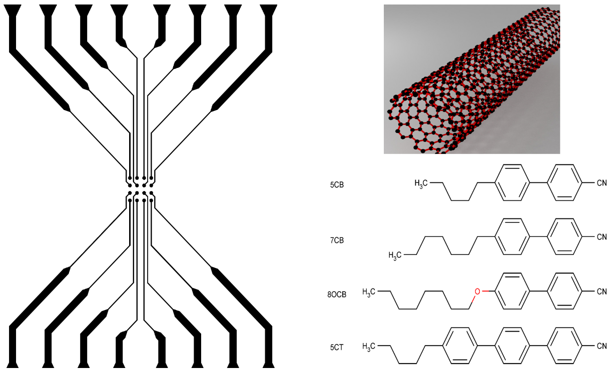


Fig. 1. Gold micro-electrode array with $50\mu m$ contacts and $100\mu m$ pitch, SWCNT and E7 LC molecule.

III. Hardware Implementation

The hardware can be divided into three main components (PC), a computer, an evolvable motherboard (EM) and the material. A simple illustration of the experimental set up and procedure is presented in Figure 2.

Evolutionary algorithms are run on the computer. They control the value of signals modifying the state of the evolvable material. A *mbed* microcontroller placed on the motherboard is connected to the computer via serial port. Signals transfered over this connection are translated into voltage levels by the *mbed*. They are subsequently sent through a set of digital-to-analogue converters (DACs) before reaching the micro-electrode array upon which the material has been deposited. The state of the material under the influence of the input signals is evaluated through output current measurements that are sent back to the micro-controller via a set of analogue-to-digital converters (ADCs). The currents are translated into a signal using an interpretation scheme and sent back to the computer via serial connection. This signal is subsequently used by the EAs to produce a new set of inputs and the process is repeated until a termination criterion is reached.

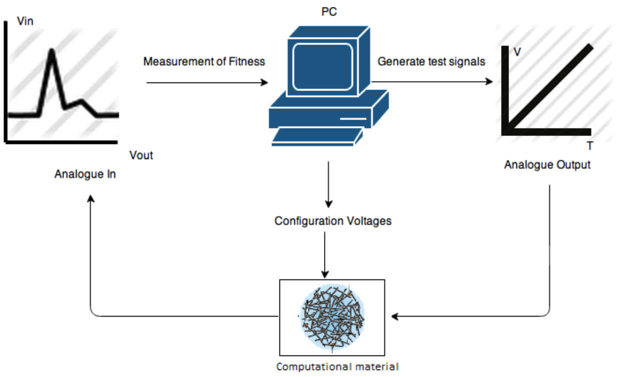
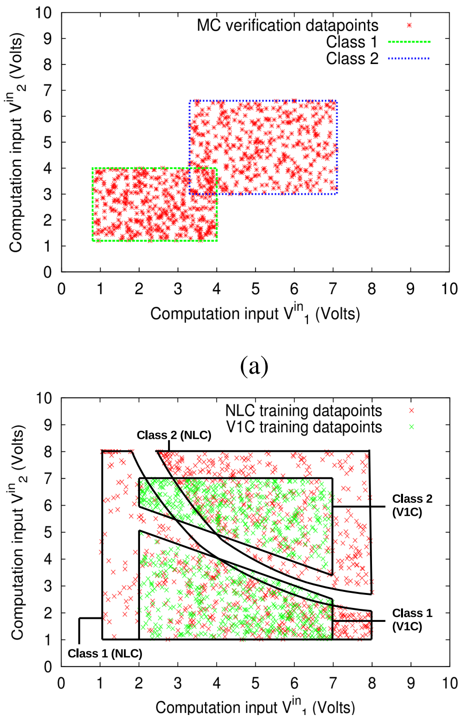


Fig. 2. Representation of the hardware implementation of EiM.

IV. Classification datasets

Three artificial binary datasets of increasing complexity are used in this paper. The three datasets comport $K = 4800$ instances. Due to the training and verification procedure followed to solve the optimisation problem which will be defined in the next Section, K is split into $K_t = 800$ training and $K_v = 4000$ verification data. Each instance is defined in $m = 2$ dimensions. The two attributes are translated into direct voltage inputs in the experimental implementation. These inputs are referred to as computation voltages, $\mathbf{V}^{in} = [V_1^{in}, V_2^{in}]$. Their values are limited by the the maximum voltage levels, $8\ Volts$, allowed in the hardware components and evolvable material. For each problem, instances from the training and verification sets are initialised independently within the boxes defining the classes. Three datasets referred to as V1C, NLC and MC are used in experiments and presented in Figure 3. The two classes of the V1C dataset are separated by a diagonal whilst those of the MC dataset are merged. The overlapping area containing approximatively 6.6% of all instances which



(a)

(b)

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

are effectively undistinguishable from one another. NLC has two classes which are separated by a hyperbola. Figure 3 shows the distribution of training data, for (a) the merged and (b) the two separable datasets, in the 2D feature space.

The complexity of the three problems is measured using the Fisher criterion f_1 [19] 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 $f_1 = 2.198$ and for NLC $f_1 = 0.962$, i.e. V1C is less complex than NLC.

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 NLC the most complex. The number of attributes, m, instances contained in the datasets, K as well as the Fisher criterion over all attributes, f_1, and the measure of overlap, f_2, defined in eqn. 2 are reported in Table I.

TABLE I
PROBLEM PARAMETERS FOR THE V1C, MC AND NLC DATASETS.

Name	number of attributes	number of instances	Fisher criterion	volume of overlap (%)
V1C	2	4800	2.198	0
MC	2	4800	1.908	6.6
NLC	2	4800	0.962	0

V. PROBLEM FORMULATION AND ALGORITHMS

The process of transforming the state of an unconfigured material in such a way that it is able to classify data is formulated as an optimisation problem. A training and verification procedure is followed to solve this problem.

The number of gold terminals in the micro-electrode array limits the number of inputs, outputs and configuration voltages that can be used in an experiment. In the case of the simple 2D binary datasets described in Section IV, it is possible to use only twelve of the array's sixteen connections: ten as inputs and two as outputs. Data to be computed by the material $\mathbf{V}^{in}$, is sent to two input terminals in the form of voltages amplitudes. The remaining eight connections are used for a set of configuration inputs. The configuration inputs are voltage levels $V_j \in [V_{\min}, V_{\max}]$, $j = 1, \ldots, 8$ which are part of the set of decision variables controlled by the algorithm during training.

In addition to the configuration inputs, the possible locations where the two components of $\mathbf{V}^{in}$ are applied is considered as a decision variable. It has been observed experimentally that allowing the algorithm to modify the input location throughout the search tends to reduce the number of iterations needed to reach an optimum result and that the latter is obtained with more consistency accross tests. 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, spread over a total of $D = 10$ dimensions, is defined as

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

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. Two direct current outputs, $\mathbf{I} = (I_1, I_2)$ mA are a measured across the two output terminals. They are used to assess the state of the material under the influence of the electric field combining $\mathbf{V}^{in}$ and V_j and are at 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, which considers the computational inputs, the corresponding induced material responses and the continuous decision variable R used as threshold; for the MC problem

$$C_M\left(\mathbf{V}^{in}(k), \mathbf{x}\right) = \begin{cases} C_1 & \text{if} \quad I_1(k) > RI_2(k) \\ C_2 & \text{if} \quad I_1(k) \leq RI_2(k). \end{cases} \tag{4}$$

and for the V1C and NLC problems

$$C_M(\mathbf{V}^{in}(k), \mathbf{x}) = \begin{cases} C_1 & \text{if} \quad I_1(k)V_1^{in}(k) + I_2(k)V_2^{in}(k) \leq R \\ C_2 & \text{if} \quad I_1(k)V_1^{in}(k) + I_2(k)V_2^{in}(k) > R \end{cases} \tag{5}$$

Variations of these schemes have been tested experimentally. It was observed that Eqs 4 and 5 produced the better results than these variations for the datasets and algorithm used in this paper. For every training data point $\mathbf{V}^{in}(k)$, $k = 1, \ldots, K_t$ the error from translating the material response according to rules (5) is

$$\epsilon_{\mathbf{x}}(k) = \begin{cases} 0 & \text{if} \qquad C_M(\mathbf{V}^{in}(k), \mathbf{x}) = C(\mathbf{V}^{in}(k)) \\ 1 & \text{otherwise.} \end{cases} \tag{6}$$

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

$$\Phi_e(\mathbf{x}) = \frac{1}{K_t} \sum_{k=1}^{K_t} \epsilon_{\mathbf{x}}(k). \tag{7}$$

A penalty term $H(\mathbf{x})$ is added to (7), given by

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

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

$$\Phi_s(\mathbf{x}) = \Phi_e(\mathbf{x}) + H(\mathbf{x}) \tag{9}$$

The optimisation training problem to be solved is that of minimising (9) 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 (5). $V_{\min} = 0$ Volts, $V_{\max} = 4$ Volts, $R_{\min} = 0.05$ and $R_{\max} = 15$.

Population-base, derivative-free, stochastic 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) [15], 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 in the following eqn. 10,

$$x_d = \begin{cases} x_d^a + F(x_d^b - x_d^c) & \text{if } d = D \text{ or } r_d < CR \\ x_d & \text{otherwise.} \end{cases} \tag{10}$$

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 [20] and have been modified empirically for the problem undertaken.

VI. Confidence Measure

Training, and especially verification errors are both measures of the performance of the nanotube-based classifiers produced using the EiM implementation described in Sections II, III and V. A new measure proposed in this section is a confidence measure, which is designed to give an indication of the probability for the class assigned to a datapoint by the classifier to be incorrect. A number of papers report discussions regarding the best way of deriving confidence measures for classifiers produced using different machine learning approaches [21, 22]. Within the context of EiM literature, the subject is discussed in [23] for logic gates evolved using a cellular automata approach. Classification problems have also been investigated, with solutions analysed using sophisticated statistical tests [24]. In both cases, the material used is a solid SWCNT/polybuthyl(methacrylate) (PBMA) composite. No discussion on confidence measures for SWCNT/LC classifiers evolved with DE algorithms, has yet been reported.

Physical quantities related to the material state are used to calculate the confidence measure. The latter will be refered to as figure of merit (FoM). The class assigned to a datapoint defined by a set of computation inputs $\mathbf{V}^{in}$ is determined using an interpretation scheme defined by eq.5 in Section V, which is dependant on the dataset used. In this scheme, the output currents measured across the material are compared to the decision variable R. Based on this comparison, the FoM for the V1C and NLC datasets is given by

$$FoM = \left| \frac{I_1(k)V_1^{in}(k) + I_2(k)V_2^{in}(k) - R}{\max_k \left\{ I_1(k)V_1^{in}(k) + I_2(k)V_2^{in}(k) - R \right\}} \right| \tag{11}$$

and for the MC dataset it is

$$FoM = \left| \frac{(I_1(k)/I_2(k)) - R}{\max_k \left\{ (I_1(k)/I_2(k)) - R \right\}} \right| \tag{12}$$

which is effectively a measure of the distance between the output currents collected across the material, when it is sent information about an instance, and the configuration variable R used in the interpretation scheme. This distance is normalised using the maximum distance achieved in the dataset. The FoM is given as a percentage and points with 0% FoM are effectively classified at random.

VII. Results and Discussion

In each experiment, a different SWCNT/LC sample has been used. Training begins with un-configured samples which, at the start of the process, randomly assign instances to the two classes C_1 and C_2. The minimum training error produced by SWCNT/LC samples during DE's search has been reported in [17] for the V1C, MC and NLC datasets. This is also the case of the ability of the resulting SWCNT/LC classifiers to generalised to unseen data. In verification tests, the optimum set of decision variables obtained by the algorithm during training is sent to the evolved samples along with the 4000 instances from the verification dataset, K_v. This process is repeated ten times. The contribution of the SWCNT structures transformed by the training process have been assessed by sending unseen instances to the evolved device, with no configuration voltages [17]. Finally, the stability of these structures when subjected to retraining was discussed in [18]. The first three columns of Table II present the optimal training Φ_e^* and mean verification $\overline{\Phi_e^v}$ errors averaged over 10 experiments for the three datasets, as well as the standard deviation σ of results across experiments.

The question addressed is concerned with the information that can be extracted from the material about its performance as a classifier. The confidence measure defined by the % Figure of Merit (FoM) is shown in the fourth column of Table II. Figure 4 illustrates the cumulated number of instances which have the same % FoM, where FoM has been rounded to zero decimals. For each dataset, measurements have been taken at the first iteration for ten evaluations with different sets of decision variables corresponding to the ten individuals contained in DE's population. In all three cases, the SWCNT/LC samples have not yet been trained, and the training error is around 50%, ie. the material randomly assigns instances to one class or the other. It can be observed that for the three datasets, the majority of intances are within 25% FoM. The outputs measured across the untrained SWCNT/LC samples do not give a good indication of the class a datapoint belongs to when used in the interpretation scheme. The distance between functions of the outputs and the threshold R is close to 0. In addition, correctly and incorrectly classified instances are indistiguishable in terms of the % FoM. The confidence interval effectively comprises all instances, which confirms that each point has a 50% chance to be assigned to the wrong class, as the device classifies randomly.

TABLE II
Performance measures for the V1C, MC and NLC datasets.

Dataset	$\Phi_e^*(\%)$	$\overline{\Phi}_{e,v}$ (%)	σ (%)	FoM (%)
V1C	0.02	0.41	0.3482	0.5375
MC	3.856	4.27	0.513	17.7
NLC	0.28	1.51	1.4161	8.75

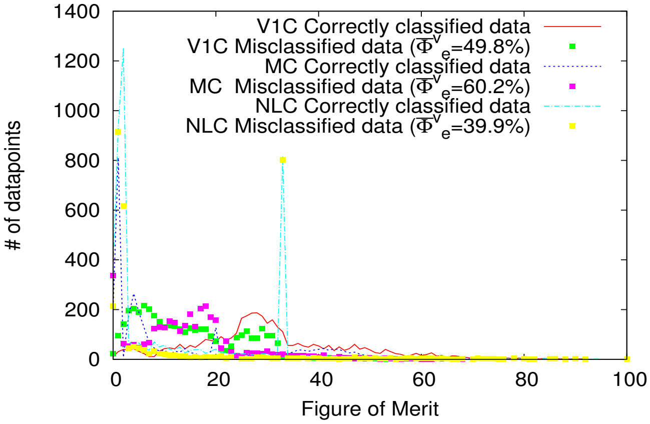
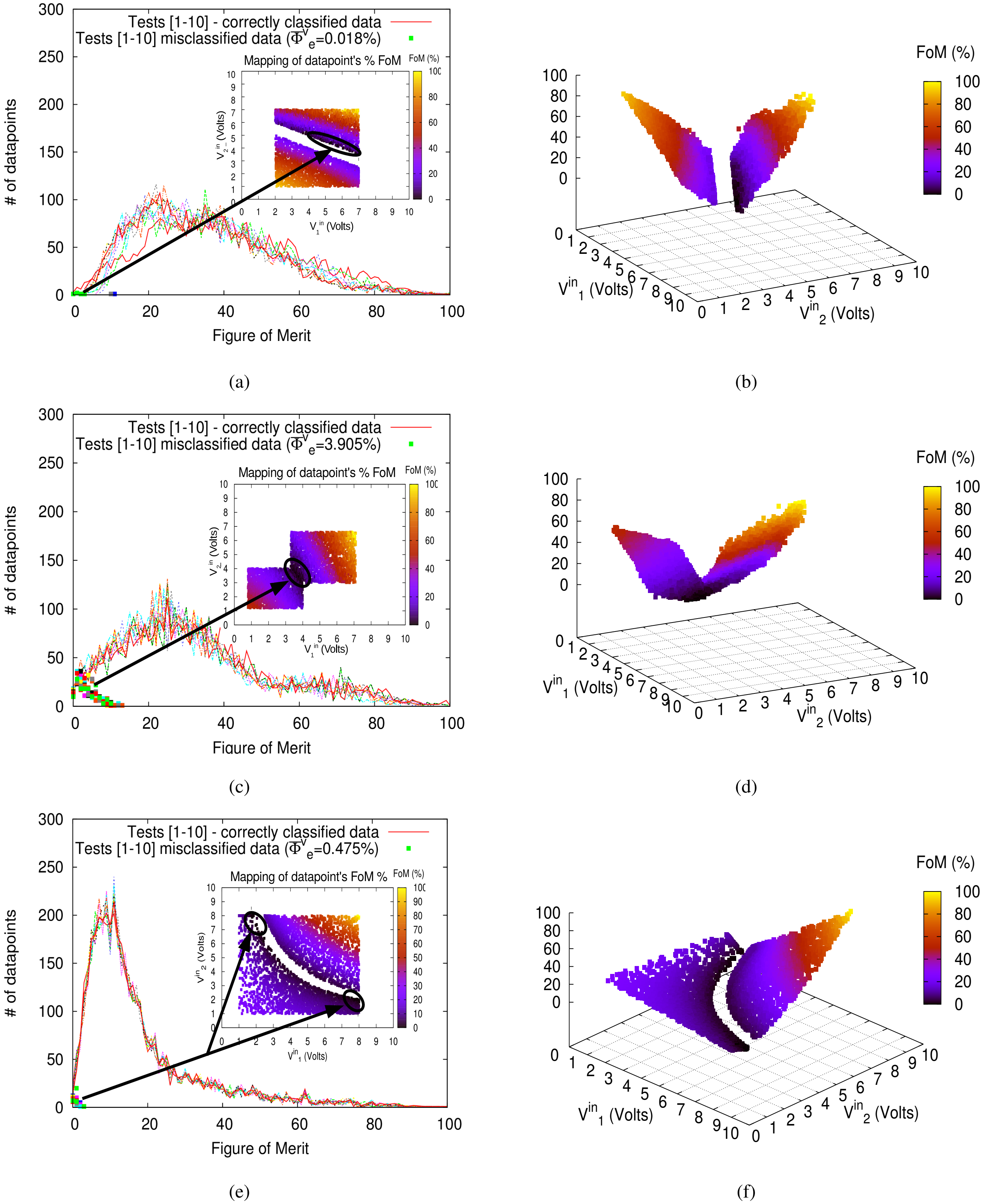


Fig. 4. Distribution of correctly and incorrectly classified training instances as a function of their distance to the separating threshold R for the three articial binary datasets.

In Figure 5(a), the cumulated numbers of correctly classified verification instances as a function of the % FoM are represented by lines. The points represent the cumulated number of instances that have been incorrectly classified by the evolved material. It can be observed that the datapoints assigned to the wrong class all obtain a very low FoM. The majority of correctly classified instances have a FoM higher than the data incorrectly classified data. The area of random class assignment lies within 5% FoM. This area contains 0.5375% of all instances from the V1C dataset. This is reported in the last column of Table II. FoM values are also plotted on the dataset 2D map. The arrow (fig. 5(a)) points towards the location on this map where the incorrectly classified instances can be found. As expected, these areas are the darkest, ie: they have the lowest FoM and therefore the highest probability of being incorrectly classified. Figure 5(b) illustrates in three dimensions the % FoM values per instance on the map of the V1C dataset. Instances in Class 1 tend to be closer to the threshold than instances from Class 2. This can be explained by the fact that the voltage inputs that characterise instances from C_2 are higher than those from C_1. The material is such that the outputs can only be higher for C_2 instances. Multiplying by the computation voltage level further increases this difference. The average verification error for V1C, reported in the second column of Table II is calculated by averaging the number of instances misclassified by the evolved SWCNT/LC classifier. Adding to this value the % of instances classified correctly but contained within the 5% FoM, the classifier error becomes 0.9475%.

In the case of the MC dataset, the FoM is computed differently, following the interpretation scheme used for this problem. In addition, the data contained in the overlapping area between the classes represents 6.6% of all instances, which, if all instances lying outside of this area are correctly classified, means that the error will be around 3.3%. This can be observed in Figure 5(c), where the misclassified instances are represented by points on the graph. The arrow points towards the area on the dataset's FoM map where instances have the lowest value, and unsurprisingly, this is situated in the



(a)

(b)

(c)

(d)

(e)

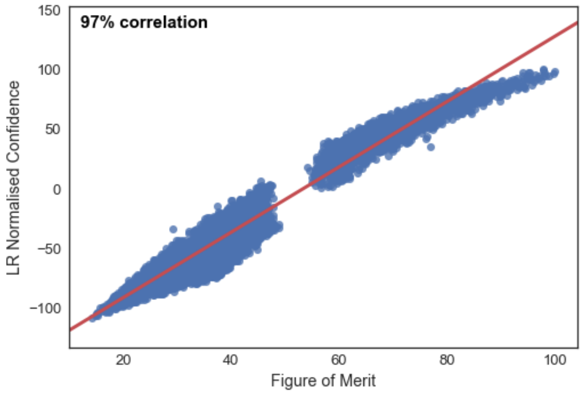
(f)

Fig. 5. Distribution of the correctly and incorrectly classified data as a function of their distance from the threshold R (LHS) and mapping of the verification error with associated confidence measure (% FoM) on the computation input space (RHS) for (a),(b) V1C, (c),(d) MC and (e),(f) NLC artificial binary datasets.

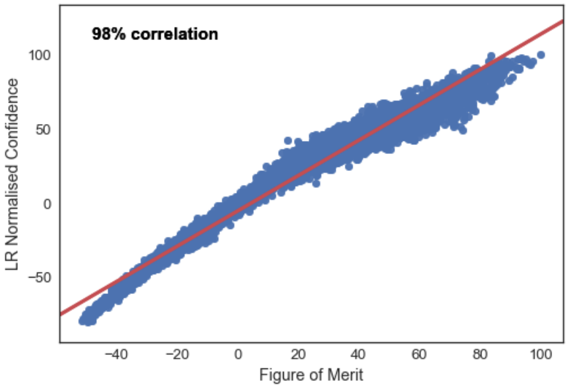
overlap. Similarly to the V1C classifier, however, the majority of correctly classified datapoints have a FoM % that is higher than the highest incorrectly-classified instance, which is 15% FoM. 7081 of all 40000 datapoints from the ten verification test share this area with the incorrectly-classified datapoints. This means that 17.7% of instances have a high probability of being randomly classified and the error increases to 21.97%.

The last two graphs in Figure 5 present the FoM results for the NLC dataset, over ten tests. The mean verification error is 0.475%, which is close to optimal and shows that the solution is relatively stable ie: sending one set of verification instances does not destroy the solution and the classifier can be re-used. In addition, there is a low standard deviation in results across tests, with $\sigma = 1.4161$. The FoM for NLC is calculated in the same way as for the V1C dataset since they share the same interpretation scheme. Compared to the unconfigured sample, there is a noticable separation in the FoM value between correctly and incorrectly classified instance. Training has produced a material state where 8.75% correctly classified instances are within the same 3% FoM as the ones assigned to the wrong class. The performance of the evolved device is not as good compared to the V1C problem, and this is irrespective of the value of the error. There is no overlapping area in this dataset, but it can be observed in the Figure 5(e) that the highest probability of error occuring is situated at the classes' boundaries. When mapped on 2D computation input space the threshold value is a hyperbola separating the two classes. And it can be seen in Figure 5(f) that, as for V1C, a higher confidence tends to be assigned to Class 2.

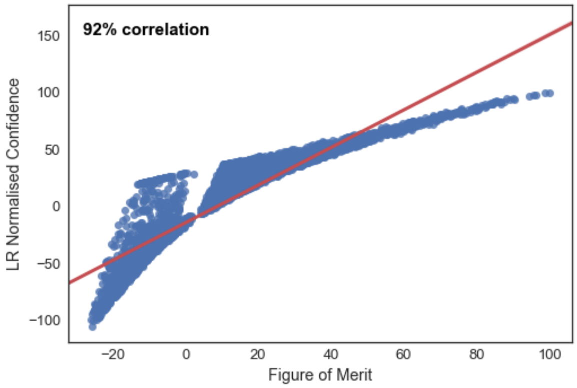
In order to compare our measure of confidence with a standard machine learning approach, Logistic Regression (LR) classifier was selected. Similar to our approach: given an input sample, LR assigns a weight to each possible class and then uses a distance-based measure to perform the classification task. The correlation between the output of the two classifiers is illustrated in Figure 6 with 98% Pearson correlation coefficient for MC, 92% for NLC, and 97% for V1C. This demonstrates the high correlation between the proposed classifier in this paper and an established classifier in machine learning.



(a)



(b)



(c)

Fig. 6. Pearson Correlation between LR normalised confidence and evolved SWCNT/LC FoM for the (a) V1C, (b) MC and (c) NLC datasets

VIII. 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 new measure of the performance of evolved SWCNT/LC classifiers is proposed. It is a confidence measure, calculated using the current outputs produced by the devices when solutions are tested against simple classification datasets. Results demonstrate that from a state where the devices classify instances randomly with very low confidence, evolution produces SWCNT/LC classifiers which are able to discriminate between two classes. The confidence associated with the evolved device's classification of instances is increased as compared to the non-trained material. The values reported for the different datasets present high correlations with the normalised confidence measure of an established machine learning classifier. Future work will extend to the use of more complex real life datasets, as well as modifying the problem formulation to make it a multi-objective optimisation problem, starting with two objectives; minimising classifier's error and maximising the confidence value.

References

[1] 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.

[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] 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.

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

[5] M. Amos, I. M. Axmann, N. Blüthgen, F. de la Cruz, A. Jaramillo, A. Rodriguez-Paton, and F. Simmel. Bacterial computing with engineered populations. *Philos. Trans. A Math. Phys. Eng. Sci.*, 373, 2015.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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 *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 5255–5261. IEEE, 2016.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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. *IEEE Congress on Evolutionary Computation (CEC)*, 2017.

[18] E. Vissol-Gaudin, A. Kotsialos, C. Groves, C. Pearson, D. A. Zeze, and M. C. Petty. Computing based on material training: Application to binary classification problems. *2017 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–8, 2017.

[19] 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.

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

[21] A. Mandelbaum and D. Weinshall. Distance-based confidence score for neural network classifiers. *arXiv preprint arXiv:1709.09844*, 2017.

[22] M. Poggi, F. Tosi, and S. Mattoccia. Quantitative evaluation of confidence measures in a machine learning world. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5238–5247, 2017.

[23] S. S. Farstad, S. Nichele, and G. Tufte. Towards standalone in-materio devices: Stable logic gates and elementary cellular automata in carbon nanotubes material. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 5246–5254. IEEE, 2016.

[24] M. Dale, S. Stepney, J. F. Miller, and M. Trefzer. Reservoir computing in materio: A computational framework for in materio computing. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2178–2185, 2017.