

Article

Precision Modeling: Application of Metaheuristics on Current–Voltage Curves of Superconducting Films

Syed Rameez Naqvi ¹, Tallha Akram ^{1,*}, Sajjad Ali Haider ¹, Muhammad Kamran ¹,
Aamir Shahzad ², Wilayat Khan ¹, Tassawar Iqbal ³ and Hafiz Gulfam Umer ⁴

¹ Department of Electrical Engineering, COMSATS University Islamabad, Wah Campus, Wah Cantonment 47040, Pakistan; rameeznaqvi@ciitwah.edu.pk (S.R.N.); sajjadali@ciitwah.edu.pk (S.A.H.); dr_m_kamran@ciitwah.edu.pk (M.K.); wilayat@ciitwah.edu.pk (W.K.)

² Department of Electrical Engineering, COMSATS University Islamabad, Abbottabad Campus, Abbottabad 22040, Pakistan; aamirsardar@gmail.com

³ Department of Computer Science, COMSATS University Islamabad, Wah Campus, Wah Cantonment 47040, Pakistan; tassawar@ciitwah.edu.pk

⁴ Department of Computer Science & IT, Ghazi University, D.G. Khan 32200, Pakistan; gulfambise@hotmail.com

* Correspondence: tallha@ciitwah.edu.pk

Received: 13 July 2018; Accepted: 1 August 2018; Published: 3 August 2018



Abstract: Contemplating the importance of studying current–voltage curves in superconductivity, it has been recently and rightly argued that their approximation, rather than incessant measurements, seems to be a more viable option. This especially becomes bona fide when the latter needs to be recorded for a wide range of critical parameters including temperature and magnetic field, thereby becoming a tedious monotonous procedure. Artificial neural networks have been recently put forth as one methodology for approximating these so-called electrical measurements for various geometries of antidots on a superconducting thin film. In this work, we demonstrate that the prediction accuracy, in terms of mean-squared error, achieved by artificial neural networks is rather constrained, and, due to their immense credence on randomly generated networks' coefficients, they may result in vastly varying prediction accuracies for different geometries, experimental conditions, and their own tunable parameters. This inconsistency in prediction accuracies is resolved by controlling the uncertainty in networks' initialization and coefficients' generation by means of a novel entropy based genetic algorithm. The proposed method helps in achieving a substantial improvement and consistency in the prediction accuracy of current–voltage curves in comparison to existing works, and is amenable to various geometries of antidots, including rectangular, square, honeycomb, and kagome, on a superconducting thin film.

Keywords: superconducting film; Shapiro steps; artificial neural networks; genetic algorithms; entropy

1. Introduction

Precisely measuring critical current density in superconducting materials and systems requires understanding of the fundamentals of current–voltage (IV) characteristics, also called the transport measurements [1]. Recent studies have shown that these IV curves may show sudden jumps, which resemble Shapiro Steps, in voltage around critical current (I_c) and/or critical temperature (T_c) in superconducting thin films. These steps usually appear when the vortex lattice is formed, which may lead to instability at high vortex velocities. These instabilities have been studied in different systems including nanotube Josephson Junctions [2], superconducting nanowires [3], low temperature thin films [4], and a square array of periodic antidots on an Nb film [5]. Studying such behavior in

superconducting films is important since Shapiro Steps are exploited to make flux Qubits that are essential for superconducting quantum computers [6]. Other mechanisms that may lead to such jumps include thermo-magnetic instabilities of vortex matter [7], thermally-assisted flux flow [8], or just overheating of the superconducting film on SiO₂ substrate.

Fabrication of various geometries by electron-beam lithography, to obtain transport measurements via a physical properties measurement system (PPMS), is an expensive, tedious and cumbersome process [9]. It has been recently pointed out that these curves, especially when they are needed for a wide range of temperature and magnetic field values, may not be necessarily measured incessantly; instead, they may be obtained using some approximation technique applied on a finite amount of curves already obtained via PPMS [10–12]. Artificial Neural Networks (ANN) have been used as the approximation method in each of these solutions to extrapolate the IV curves for unforeseen values of critical parameters. It shall be discussed in Section 2.2 that training the ANN requires two *randomly* generated networks' coefficients called *weights* and *biases*, which are updated in each iteration, until an optimal solution (with the smallest mean-squared error, MSE) is obtained. Mainly due to this randomness in the coefficient generation, ANN tends to converge to one local minima from a pool of possible solutions, which makes training of the ANN a nondeterministic process. The latter, in turn, may lead to inordinately varying values of MSE, number of iterations, and time to converge. In this way, two rounds of training on the same data may result in different prediction accuracies.

The proposed work intends to address the aforesaid problem by controlling the randomness in coefficient generation, such that the ANN always converges (close) to the global minima, by means of a novel *entropy based genetic algorithm*. Our case study is the prediction of IV curves for four different geometries of antidots, including rectangular, square, honeycomb, and kagome, on an Nb film. However, the proposed approach is equally applicable to other superconducting films as an alternative approximation technique for measuring other properties at the same time. The main contribution of the proposed work, therefore, is the increased accuracy in the prediction of IV curves by means of *Entropy*, which in a nutshell is the uncertainty measurement associated with initialization of the weights and biases. Since ANN are highly dependent on their initial conditions for their fast convergence and accurate approximation, the selected vector of weights and biases should have a minimum entropy. To the best of our knowledge, entropy, especially in conjunction with genetic algorithm (GA), has never been adopted for the prediction of IV curves for a superconducting film, which, we show, outperforms the conventionally used predictors.

The rest of the paper is organized as following: Section 2 covers our experimental setup and a brief overview of existing methodologies and relevant works. We present the problem formulation in Section 3. Section 4 presents the proposed methodology, and an overview of GA. In Section 5, we present simulation results and a comparative analysis of the used algorithms, before we conclude the paper in Section 6.

2. Related Work and Experimental Setup

In order to be able to draw a fair comparison between the proposed methodology and the existing works, we decided to adopt the same experimental setup and the superconducting film of the same thickness. In what follows, we give an overview of the experimental setup and fundamentals of the ANN, which is mainly an extract of the benchmark works [10–12].

2.1. Experimental Setup and Measurements Using PPMS

For our experiments, we deposited a high quality 60 nm Nb film on a SiO₂ substrate, followed by fabrication of micro-bridges by ultraviolet photolithography and dry etching in order to obtain transport measurements. The desired geometries of circular antidots were obtained by applying e-beam lithography on a photo-resist layer. Finally, magnetically enhanced dry etching transferred the patterns to the film. A commercially available PPMS was used to perform the transport measurements. Figure 1 presents the scanning electron microscopy (SEM) of various geometries.

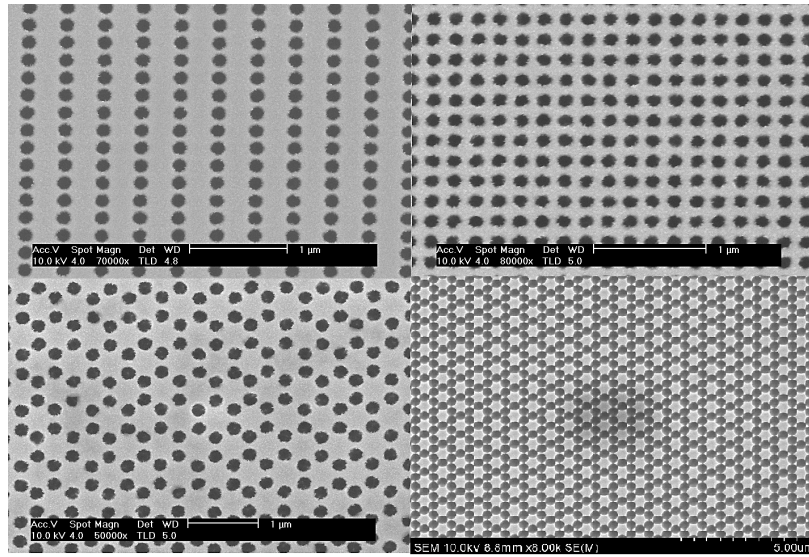


Figure 1. Scanning Electron Micrograph of superconducting Nb film with four geometries: **top left-right** (rectangular-square), **bottom left-right** (honeycomb-kagome).

In the IV curves measured at different values of temperature and magnetic field, shown in Figure 2, we observed Shapiro steps around T_c . However, these steps continued to weaken with increasing temperature until it completely vanished. These temperature dependent curves might be divided into three regions: between two regions—in each of which the curve showed some slope—there was a region that comprised the Shapiro steps. We repeatedly varied magnetic fields and kept the current constant, and vice versa to collect 25,600 samples arranged in a $4 \times 4 \times 1600$ matrix for each geometry, where 4×4 refers to the 4 constant values of current and magnetic fields. In our proposed work, we isolated one sample from each matrix and subjected the rest for training the ANN model. Once trained, the isolated sample should be compared with the predicted curve for the same values of magnetic field and temperature.

2.2. Approximation Using ANN

ANN imitate a human brain and are supposed to perform learning in a given situation and repeat in another [13]. They tend to establish relationships between independent variables of a small subset, and are widely used for approximation on a larger sample space by utilizing those relationships. An ANN architecture, mainly, is an interconnection of several neurons [14] in a directed graph, which provide mapping from an input layer to an output layer via a few hidden layers sandwiched in between. The neurons in each layer carry some real *weights* and *bias*—together called networks' coefficients. A different set of weights, which are updated in each iteration of the training process, leads to a different network's response. A number of training algorithms exist in literature—each of which updates the coefficients in a unique manner, but, for most of them *backpropagate*, their errors from the output layer to input layer in order to minimize the objective function. The goal of this training or learning process is to achieve the smallest mean squared error (MSE) between the target and the actual system's response [15]. The network's response carrying only one hidden layer is given by Equation (1):

$$y_k = \delta_o \left(\sum_{j=0}^M w_{kj}^y (\delta_H (\sum_{i=0}^N w_{ji}^H x_i + b_i)) + b_j \right), \quad (1)$$

where δ_o and δ_H are the *threshold* or *activation* function in the output and hidden layer, respectively, w and b represent weights and bias terms, respectively, and x_i represents the i th element in the input layer.

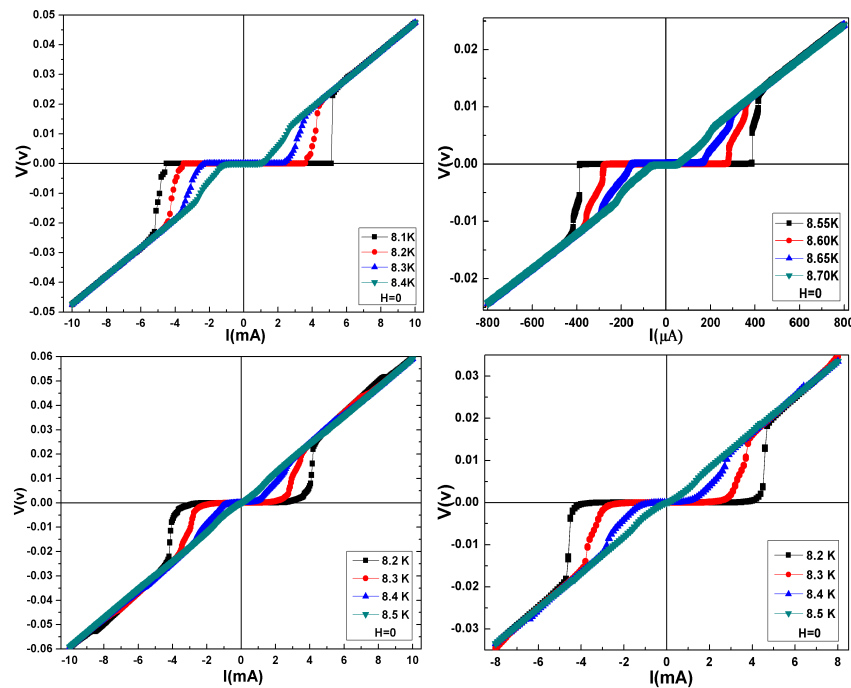


Figure 2. The IV characteristics of four geometries at different temperatures and magnetic fields: **top left-right** (rectangular-square), **bottom left-right** (honeycomb-kagome).

Without going into details of the available training algorithms, in this work, we will stick only to those that were used for the same purpose in the benchmark works. The first work [12] that proposed to approximate the IV curves for an Nb film made use of Bayesian Regularization (BR) algorithm for training the ANN for a square array of antidots. The algorithm converged in fifty-five iterations (epochs), and managed to achieve the best MSE of 2.08×10^{-8} . This work did not explore other available options—be it in terms of training algorithms or ANN architectures.

The second benchmark work [10] in contrast did the same for a diluted square lattice and presented a thorough comparison between three different ANN architectures, each with ten different configurations (number of neurons in each layer), and trained by six training algorithms—a total of 180 ANN models were developed, and the best MSE obtained was 4.55×10^{-8} . Figure 3 shows the MSE obtained by all the models, where cascaded, feedforward, and layer recurrent are the three ANN architectures; it is important to note the diversity in the obtained results, which is mainly due to random nature of the generated coefficients.

The last benchmark work [11] was an extension of the second benchmark, in which a comparison was drawn between four geometries of antidots on the same Nb film. The lowest MSB reported in this work was in the order of $\times 10^{-9}$ for a specific architecture and a specific training algorithm. However, the diversity in the obtained results was once again enormous, and nondeterministic.

Other than these benchmark works, there have been a few notable attempts on proposing formal models for analyzing IV curves for superconducting devices and films. For instance, a formal, self-consistent, model was presented for estimating critical current of superconducting devices [16]. The authors admitted that an array of antidots based thin film was very difficult to model mathematically. Although its computation burden was less than that of the ANN based models, it lacked accuracy—tolerance of 4–6% within actual values as compared to the ANN models having tolerance less than 1% with the actual values. Another attempt based on ANN was presented by Bonanno et al. [17]. The authors proposed a radial basis function neural network (RBFNN) and demonstrated that it had a prediction accuracy of about 10^{-1} (MSE), which is even lesser than the already existing techniques—let alone our current model based on a genetic algorithm, achieving MSE in the order of 10^{-9} .

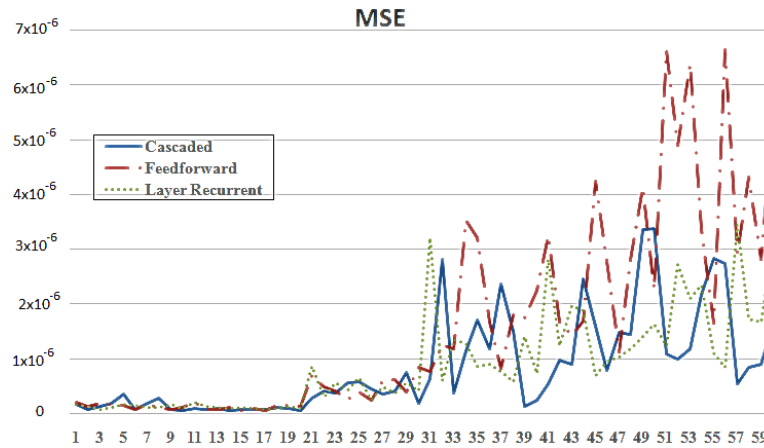


Figure 3. MSE achieved by a benchmark work for three ANN architectures. Reprinted from [10] with permission.

3. Problem Statement

Let $\Omega \subset \mathbb{R}^v$, where $v = 2; v \rightarrow \{R' \times D'\} | R' > D'$, be an experimental values. Let the selected feature set, $\phi = \phi(i) | i \in \Omega$ where $\phi_1(i), \dots, \phi_n(i) \subset \phi$ are associated with the ANN training process to predict output vector $\tilde{\phi}_{pred}$. The set of features ϕ are mapped to $\tilde{\phi}_{pred}: \phi \rightarrow \tilde{\phi}_{pred}$. The output vector $\tilde{\phi}_{pred}$ is calculated as:

$$\{\tilde{\phi}_{pred} \triangleq (\delta_{WB}^k, pop_t, \delta_R, \delta_s) \in \{-1 : 1\}, (\delta_E^k, \delta'_{WB}, X'_{ofs}, \phi_{pred}^y) \in \mathbb{R}^2\},$$

where the set of variables $\delta_{WB}^k, pop_t, \delta_R, \delta_s, \delta_E^k, X'_{ofs}$ and ϕ_{pred}^y have been described in the list of symbols. The objective function in case of ANN is selected to be MSE given in the following relation:

$$\Gamma(\tilde{w}) = \Psi \sum_{q=1}^Q (\Gamma_{tar} - \Gamma_{act})^T (\Gamma_{tar} - \Gamma_{act}) = \Psi \sum_{q=1}^Q e_q^2. \quad (2)$$

4. Proposed Optimization of ANN's Coefficients

4.1. Genetic Algorithms

Genetic Algorithm (GA) is an evolutionary technique belonging to a class of stochastic search algorithms, which finds an optimal solution from a *pool of solutions* based on the principle of *survival of the fittest* [18]. In GA framework, each individual is represented by a string called a chromosome, whereas a group of chromosomes generate a population. For this architecture, weights and biases vector (chromosome) is generated, which is replicated multiple times to generate a population of some pre-defined size.

Binary representation of a GA is commonly used where each chromosome is a vector c , constituting a set of m genes from the set 0, 1:

$$C_i^k = \{G_1^k, G_2^k, \dots, G_m^k\} \quad G_m^k \in \{0, 1\},$$

where m is the length of a chromosome.

However, in practical optimization, it is more natural to represent a gene in real numbers for an optimized solution [19]. The continuous domain provides larger space and more convergence possibilities. Data range is normalized to the range of 0 $\rightarrow$ 1 prior to binary encoding, and, for this specific application, the chromosome is a floating point vector. In what follows, we present a few GA operators, called crossover and mutation, implemented for a thorough technical analysis.

4.1.1. Crossover Operators

Selecting a pair of chromosome $C_i^1 = (G_1^1, G_2^1, \dots, G_m^1)$ and $C_i^2 = (G_1^2, G_2^2, \dots, G_m^2)$ for a multiple crossover description.

1. Flat Crossover

An offspring O^i is generated where $O_i^k = (g_1^k, g_2^k, \dots, g_m^k)$ is a uniformly chosen random value from the interval $[\min(G_i^1, G_i^2), \max(G_i^1, G_i^2)]$, where m is the index for the number of genes and k is the index for the number of chromosomes.

2. Linear Breeder Genetic Algorithm (BGA) Crossover

Under the same consideration as above, let $O_i = G_i^1 \pm \text{rang}_i \cdot \gamma \cdot \Lambda$, where $\Lambda = (G_i^2 - G_i^1) / \|G_i^1 - G_i^2\|$ and $\gamma = \sum_{l=0}^{15} \alpha_l 2^{-l}$. An $\alpha \in \{0, 1\}$ is generated randomly with the probability of $p(\alpha_i = 1) = 1/12$. Usually, $\text{rang}_i = 0.5(\kappa_i - \tau_i)$ and $-$ sign is chosen with the probability of 0.8 [20].

3. Arithmetic Crossover

In arithmetic crossover, two Offsprings are generated, $O^k = (g_1^k, g_2^k, \dots, g_m^k)$, $k = 1, 2$ where: $g_i^1 = \lambda G_i^1 + (1 - \lambda) G_i^2$ and $g_i^2 = \lambda G_i^2 + (1 - \lambda) G_i^1$. Here, λ is a constant and user-defined value, which can vary with the number of generations.

4.1.2. Mutation Operator

Let the maximum number of generations be represented by R_{max} and R_t denotes the generation on which mutation operator is applied. As per mutation rule:

$$G'_i = \begin{cases} G_i + \delta(t, \kappa_i - G_i), & \text{if } \gamma = 0, \\ G_i + \delta(t, G_i - \tau_i), & \text{if } \gamma = 1. \end{cases}$$

4.1.3. Selection

Selection is a process of selecting chromosomes with the smallest value of the cost function. The selection rate ς_{rate} defines the survivors eligible for mating in the next generation. Generally, the $\varsigma_{rate} = 50\%$, and the population selection is defined as:

$$\eta_{rem} = \varsigma_{rate} \times \eta_{pop}, \quad (3)$$

$$\eta_{keep} = \eta_{pop} - \eta_{rem}. \quad (4)$$

The selection probability depends on the cost weight, calculated as:

$$P_k = \frac{C^k}{\sum_{j=1}^{\eta_{keep}} C^j}. \quad (5)$$

The chromosome with the lowest cost in terms of mean squared error has the highest selection probability. The other selection methods include roulette wheel, rank selection and tournament [21]. Roulette wheel is a probability based method, whereas the tournament selection is a winner-takes-all based technique [22].

4.1.4. Operation of a GA

The operation of a GA is summarized as follows:

1. Create an initial population from a randomly generated weights and biases vector.
2. Repeat until the best individuals are selected:

- Evaluate the fitness using MSE,
 - Select the parents with best fitness level,
 - Apply the selected crossover and mutation operators.
3. Terminate upon convergence.

Optimization is a process of finding the best possible solution from a given search space. In this work, an evolutionary strategy is utilized to fine-tune the parameters so as to minimize the cost function. Considering the GA's exquisite performance on various platforms, and their ability to handle larger space problems even for stochastic objective functions, another domain is exploited, which is to optimize the solutions of artificial neural networks. The procedure has two possible directions: (1) optimize the weights and biases; and (2) optimize the architecture of ANN. In this work, the former method is used to improve the prediction accuracy.

4.2. Entropy Based GA for Optimization

For a set of discrete random variables $C_i^k = \{G_1^k, G_2^k, \dots, G_m^k\}$, the Shannon entropy is defined as the following:

$$\delta_E = - \sum_{k=0}^{m-1} (G_i^0 / \delta G) \log_2 (G_i^0 / \delta G), \quad (6)$$

$$\delta G = \sum_{k=0}^{m-1} G_0^i, \quad (7)$$

where G_i^j is a vector of weights and biases called a chromosome.

The proposed cascaded design is the conjunction of three separate modules comprising an entropy calculator, GA optimizer, and ANN's approximator. Here, the entropy calculator controls the uncertainty with the constraint of minimum randomness. The resulting vector, which is then forwarded to the GA module for optimization, comprises weights and biases with either minimum entropy or maximum repetition. Finally, the third module uses the optimized weights and biases to train the ANN for final approximation. Figure 4 summarizes the proposed methodology, and therefore the major contribution of this work, in a flow chart, and the proposed curve approximation with the optimized coefficients approach is given in Algorithm 1.

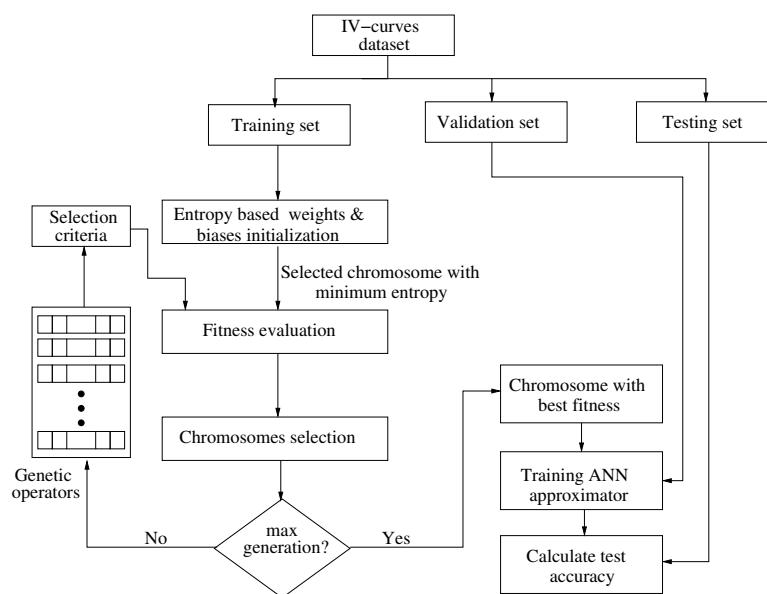


Figure 4. Flow chart depicting the proposed methodology.

Algorithm 1: Entropy based GA for weights and biases optimization

Result: Curve Approximation

- 1 **Begin:**
- 2 Initialize step size (δ_s), learning rate (δ_r), k , ϵ and weights and biases vector δ_{WB} .
- 3 Calculate Entropy δ_E^k of vector δ_{WB}^k using Equation (1), k times.
- 4 Calculate $\mathfrak{F}_{maxrep}$ and $\mathbb{V}_{min}$ from δ_E^k .
- 5 **if** $\exists (\mathfrak{F}(\text{mode}(\delta_E^k)))$ **then**
- 6 $\delta'_{WB} = [V_{maxrep}]$
- 7 **end**
- 8 **else**
- 9 $\delta'_{WB} = [V_{min}]$
- 10 **end**
- 11 **Return** (Weights and biases vector with entropy δ'_{WB})
- 12 $t_{GA} \leftarrow 0$
- 13 $pop_0 \leftarrow$ Initialize from δ'_{WB} with (pop_{size})
- 14 Evaluate pop_0
- 15 **while** $t_{GA} < Gen_{max}$ **do**
- 16 Parents (χ_{par}) $\leftarrow$ Select χ_{par} from pop_t
- 17 Offspring (χ_{ofs}) $\leftarrow$ crossover (χ_{ofs} , C_{prob})
- 18 mutation (χ_{ofs} , M_{prob})
- 19 Evaluate χ_{ofs}
- 20 $pop_{t+1} \leftarrow$ actual population through replacement
- 21 $pop_t \ \& \ \chi'_{ofs}$
- 22 $t_{GA} \leftarrow t_{GA} + 1$
- 23 **end**
- 24 **Return** (Optimized chromosome (a vector of weights and biases), χ'_{ofs})
- 25 $t_{ANN} \leftarrow 0$
- 26 **while** $t_{ANN} < (Max_{epochs} \ || \ \epsilon)$ **do**
- 27 Initiate the training process using selected algorithm to compute $\tilde{\phi}_{pred}$
- 28 Update weights and biases (ΔW & ΔB)
- 29 Calculate difference between predicted output and actual output
- 30 **if** ($\tilde{\phi}_{pred} \leq \epsilon$) **then**
- 31 Exit
- 32 **end**
- 33 $t_{ANN} \leftarrow t_{ANN} + 1$
- 34 **end**
- 35 **Return** (Trained network with updated weights and biases)

5. Simulation Results

5.1. Design Parameters

The number of hidden layers is fixed at three, where each has 27, 17, and 8 neurons in order; this is generally represented by *network's configuration* {27 17 8}. The mutation rate is defined to be 0.2 and entropy count variable κ is selected to be 200 for the optimal solution. Maximum generations of GA and maximum epochs of ANN are fixed to be 1000, while GA tolerance value ϵ is selected to be 1×10^{-8} . Learning rate δ_r and initial step size δ_s are selected to be 0.9 and 0.8, respectively.

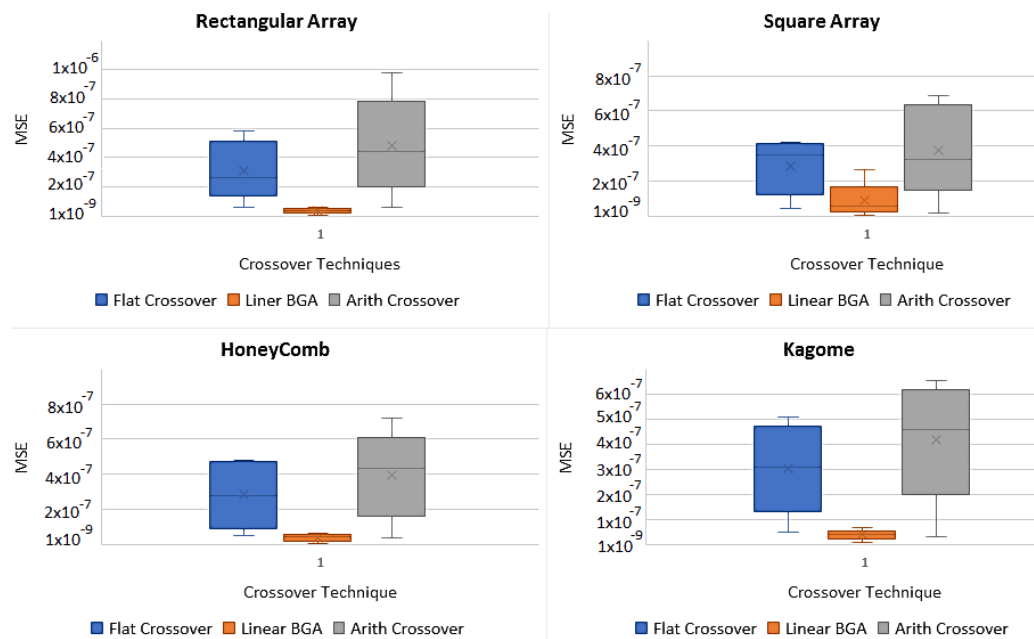


Figure 5. MSE comparison of various cross-over techniques on the selected geometries.

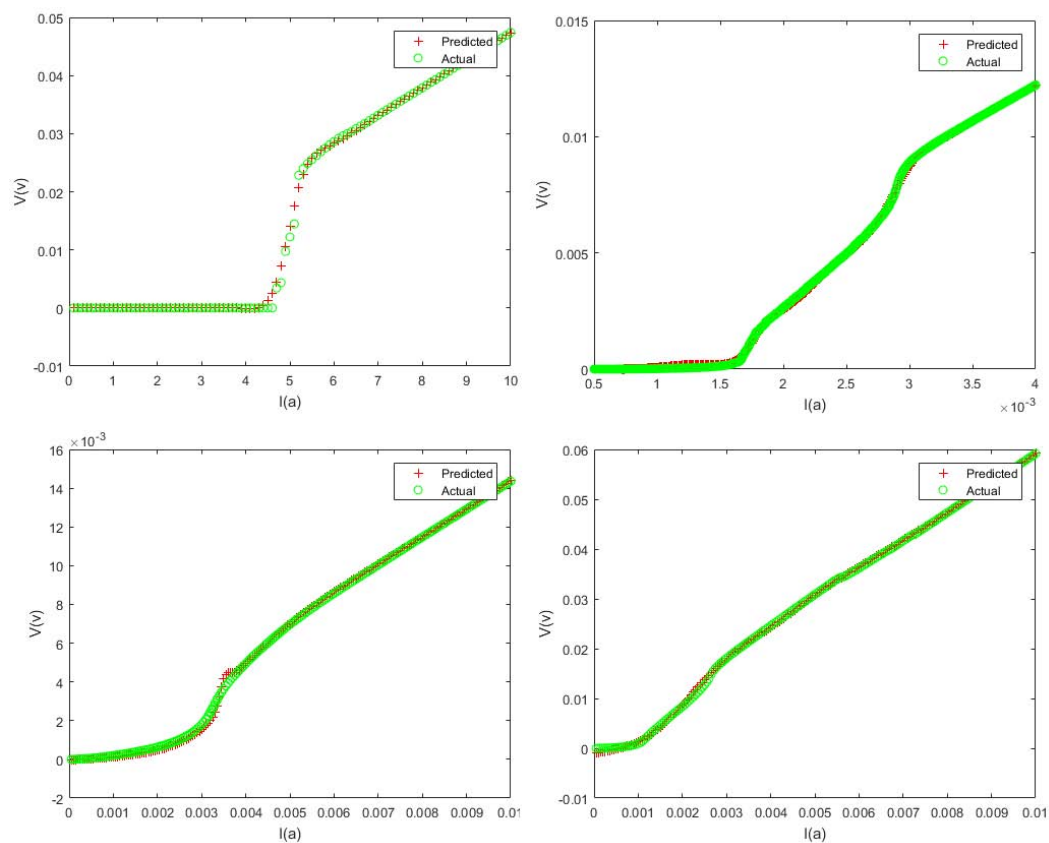


Figure 6. Predicted vs. measured IV characteristics.

Table 2. Comparison of various ANN's training algorithms as used in benchmark works.

Geometry	Training Algorithm					Epochs	MSE
	LM	CGF	CGB	CGP	BR		
Square	✓					30	5.73×10^{-7}
		✓				591	9.41×10^{-7}
			✓			252	2.33×10^{-6}
				✓		322	6.21×10^{-7}
					✓	55	2.08×10^{-7}
Rectangular	✓					39	6.37×10^{-7}
		✓				478	8.33×10^{-6}
			✓			767	4.67×10^{-6}
				✓		388	4.56×10^{-7}
					✓	69	9.22×10^{-8}
HoneyComb	✓					54	4.33×10^{-7}
		✓				634	3.56×10^{-7}
			✓			276	5.82×10^{-7}
				✓		376	2.76×10^{-7}
					✓	67	8.67×10^{-8}
Kagome	✓					43	6.77×10^{-7}
		✓				476	4.59×10^{-7}
			✓			291	3.54×10^{-7}
				✓		463	9.34×10^{-7}
					✓	79	8.56×10^{-8}

6. Conclusions

Being predominantly dependent upon random initial conditions, the generated network's coefficients mostly force the artificial neural networks to converge to a different local minima on every execution. This generally leads to an inconsistent prediction accuracy, even for identical experimental setup and tunable parameters. In this work, we have proposed an entropy based genetic algorithm, and used it to control the randomness in coefficient generation. This technique forces the network to converge (close) to the global minima on every execution, which constrains the prediction accuracy, measured in mean-squared error, to a certain acceptable level. We have applied our technique to approximate the current–voltage curves for four different lattices on a superconducting film, and compared our results with three recent works, which made use of artificial neural networks to achieve the same. Our results have shown that the proposed methodology yields better consistence and greater prediction accuracy.

Author Contributions: S.R.N. conceived the entire concept, and was responsible for most of the writeup. T.A. came up with the concept of entropy controlled genetic algorithm, and was assisted by T.I. in performing MATLAB simulations. S.A.H. and H.G.U. integrated artificial neural networks with the proposed genetic algorithm, and performed MATLAB simulations. W.K. carried out an in-depth analysis of the obtained results, and made a comparison with state-of-the-art prediction techniques. A.S. proofread the manuscript, and contributed significantly in submitting the revision. M.K. performed transport measurements on Physical Properties Measurement System.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following is a list of abbreviations and symbols used throughout the paper.

$\mathbb{R}^l$	Set of real numbers
Ω	Subset of features within $\mathbb{R}^l$
ϕ	Selected set of features
$\tilde{\phi}_{pred}^y$	Vector of predicted values
δ_R	Learning rate
δ_s	Step size
δ_{WB}^k	Vector of weights and biases
Ψ	MSE constant
δ_E^k	Vector of entropy values
δ'_{WB}	Vector of optimized weights and biases
J_{we}	Jacobian matrix
I	Identity matrix
ϵ	Error threshold
ϱ	Direction set variable
r_k	Input vector
e_q	Generated error
$\Gamma(\tilde{w})$	Cost function based on mse
Γ_{tar}	Target vector
Γ_{act}	Actual output vector
δ_o	Output layer transfer function
δ_H	Hidden layer transfer function
pop_t	Population size
X'_{ofs}	Offspring with maximum fitness
C^k	k th chromosome
G^k	k th gene
η_{rate}	Population selection rate
δ_{keep}	Selected number of individuals in population pop_t

References

1. Acharya, S.; Bangera, K.V.; Shivakumar, G.K. Electrical characterization of vacuum-deposited p-CdTe/n-ZnSe heterojunctions. *Appl. Nanosci.* **2015**, *5*, 1003–1007.
2. Cleuziou, J.P.; Wernsdorfer, W.; Andergassen, S.; Florens, S.; Bouchiat, V.; Ondarçuhu, T.; Monthieux, M. Gate-tuned high frequency response of carbon nanotube Josephson junctions. *Phys. Rev. Lett.* **2007**, *99*, 117001.
3. Dinsmore, R.C., III; Bae, M.H.; Bezryadin, A. Fractional order Shapiro steps in superconducting nanowires. *Appl. Phys. Lett.* **2008**, *93*, 192505.
4. Mandal, S.; Naud, C.; Williams, O.A.; Bustarret, É.; Omnès, F.; Rodière, P.; Meunier, T.; Saminadayar, L.; Bäuerle, C. Detailed study of superconductivity in nanostructured nanocrystalline boron doped diamond thin films. *Phys. Status Solidi* **2010**, *207*, 2017–2022.
5. Kamran, M.; He, S.-K.; Zhang, W.-J.; Cao, W.-H.; Li, B.-H.; Kang, L.; Chen, J.; Wu, P.-H.; Qiu, X.-G. Matching effect in superconducting NbN thin film with a square lattice of holes. *Chin. Phys. B* **2009**, *18*, 4486.
6. Heiselberg, P. Shapiro Steps in Josephson Junctions. Niels Bohr Institute, University of Copenhagen. 2013. Available online: https://cmt.nbi.ku.dk/student_projects/bsc/heiselberg.pdf (accessed on 13 July 2018).
7. Sidorenko, A.; Zdravkov, V.; Ryazanov, V.; Horn, S.; Klimm, S.; Tidecks, R.; Wixforth, A.; Koch, T.; Schimmel, T. Thermally assisted flux flow in MgB₂: Strong magnetic field dependence of the activation energy. *Philos. Mag.* **2005**, *85*, 1783–1790.
8. Fogel, N.Y.; Cherkasova, V.G.; Koretzkaya, O.A.; Sidorenko, A.S. Thermally assisted flux flow and melting transition for Mo/Si multilayers. *Phys. Rev. B* **1997**, *55*, 85.

9. Yu, C.C.; Chen, Y.T.; Wan, D.H.; Chen, H.L.; Ku, S.L.; Chou, Y.F. Using one-step, dual-side nanoimprint lithography to fabricate low-cost, highly flexible wave plates exhibiting broadband antireflection. *J. Electrochem. Soc.* **2011**, *158*, J195–J199.
10. Haider, S.A.; Naqvi, S.R.; Akram, T.; Kamran, M. Prediction of critical currents for a diluted square lattice using Artificial Neural Networks. *Appl. Sci.* **2017**, *7*, 238.
11. Haider, S.A.; Naqvi, S.R.; Akram, T.; Kamran, M.; Qadri, N.N. Modeling electrical properties for various geometries of antidots on a superconducting film. *Appl. Nanosci.* **2017**, *7*, 933–945.
12. Kamran, M.; Haider, S.A.; Akram, T.; Naqvi, S.R.; He, S.K. Prediction of IV curves for a superconducting thin film using artificial neural networks. *Superlattices Microstruct.* **2016**, *95*, 88–94.
13. Naqvi, S.R.; Akram, T.; Haider, S.A.; Kamran, M. Artificial neural networks based dynamic priority arbitration for asynchronous flow control. *Neural Comput. Appl.* **2016**, *29*, 627–637.
14. Güçlü, U.; van Gerven, M.A. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* **2015**, *35*, 10005–10014.
15. Haykin, S. *Neural Networks and Learning Machines*; Pearson: Upper Saddle River, NJ, USA, 2009; Volume 3.
16. Zermeno, V.; Sirois, F.; Takayasu, M.; Vojenciak, M.; Kario, A.; Grilli, F. A self-consistent model for estimating the critical current of superconducting devices. *Supercond. Sci. Technol.* **2015**, *28*, 085004.
17. Bonanno, F.; Capizzi, G.; Graditi, G.; Napoli, C.; Tina, G.M. A radial basis function neural network based approach for the electrical characteristics estimation of a photovoltaic module. *Appl. Energy* **2012**, *97*, 956–961.
18. Naqvi, S.R.; Akram, T.; Iqbal, S.; Haider, S.A.; Kamran, M.; Muhammad, N. A dynamically reconfigurable logic cell: From artificial neural networks to quantum-dot cellular automata. *Appl. Nanosci.* **2018**, *8*, 89–103.
19. Haupt, R.L.; Haupt, S.E. *Practical Genetic Algorithms*; Wiley: New York, NY, USA, 1998; Volume 2.
20. Herrera, F.; Lozano, M.; Sánchez, A.M. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *Int. J. Intell. Syst.* **2003**, *18*, 309–338.
21. Razali, N.M.; Geraghty, J. Genetic algorithm performance with different selection strategies in solving TSP. In Proceedings of the World Congress on Engineering, London, UK, 6–8 July 2011; International Association of Engineers: Hong Kong, China, 2011; Volume 2, pp. 1134–1139.
22. Mc Ginley, B.; Maher, J.; O’Riordan, C.; Morgan, F. Maintaining healthy population diversity using adaptive crossover, mutation, and selection. *IEEE Trans. Evol. Comput.* **2011**, *15*, 692–714.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).