Received July 15, 2019, accepted July 30, 2019, date of publication August 8, 2019, date of current version August 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933947

Automatic Design of Artificial Neural Networks for Gamma-Ray Detection

FILIPE ASSUNÇÃO[1], JOÃO CORREIA[1], RÚBEN CONCEIÇÃO[2], MÁRIO JOÃO MARTINS PIMENTA[2], BERNARDO TOMÉ[2], NUNO LOURENÇO[1], AND PENOUSAL MACHADO[1]

[1]CISUC, Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
[2]LIP/IST, 1600-078 Lisbon, Portugal

Corresponding author: Filipe Assunção (fga@dei.uc.pt)

This work was supported in part by the Fundação para a Ciência e Tecnologia (FCT), Portugal, under the PhD Grant SFRH/BD/114865/2016.

ABSTRACT The goal of this work is to investigate the possibility of improving current gamma/hadron discrimination based on the shower patterns recorded on the ground. To this end, we propose the use of Convolutional Neural Networks (CNNs) for their ability to distinguish patterns based on automatically designed features. In order to promote the creation of CNNs that accurately uncover the hidden patterns in the data, and at the same time avoid the burden of hand-crafting the topology and learning hyper-parameters we resort to NeuroEvolution; in particular, we use Fast-DENSER++, a variant of Deep Evolutionary Network Structured Representation. The results show that the best CNN generated by Fast-DENSER++ improves by a factor of 2.37 when compared to the results reported by classical statistical approaches. Additionally, we experiment ensembling the best-generated CNNs; the ensemble leads to an improvement by a factor of 2.48. These results establish a new state-of-the-art in the gamma/hadron discrimination problem, based on the ground impact patterns, and thus prove that CNNs automatically discovered by Fast-DENSER++ can be used to enable investment savings due to the need for smaller grids of sensors.

INDEX TERMS Artificial neural networks, evolutionary computation, Gamma-ray detection.

I. INTRODUCTION

High-energy gamma-rays constitute one of the best probes to investigate extreme phenomena in the Universe, such gamma-rays arising from fast rotating neutron stars or super-massive black holes. The detection of this kind of astrophysical radiation, whose energies span from 10 GeV up to 100 TeV, can be done at lower energies by satellite bourne detectors. However, above a few hundreds GeV, the flux becomes too small, and only ground-based experiments can measure indirectly gamma-rays. These experiments take advantage of the electromagnetic cascade that is produced by the interaction of gamma-rays with Earth's atmosphere to infer the direction and energy of the primary gamma-ray. If the energy of the gamma-ray is sufficiently high and the detection of the secondary shower particles is done at high altitude, then it is possible to survey large portions of the sky and be sensitive to transient phenomena. The observation of high-energy gamma-rays with ground-arrays, although effective, comes with a cost: one has to deal with the huge background of cosmic rays that bombard the Earth continuously. To select gamma-rays out of the hadronic background one can explore the characteristics of the shower development. Contrary to pure electromagnetic showers, hadron induced showers produce high transverse momentum particles which lead to the transverse broadening of the shower and the creation of clusters. Experimentally, the above features can be explored by measuring the steepness and bumpiness of the lateral distribution of particles at the ground with respect to the shower core position or by measuring the relative amount of signal (number of particles) at large distances from the shower core. However, the patterns of the secondary particles at the ground remain to be explored, although some studies have shown that this might have some gamma/hadron discrimination power. In this manuscript, we intend to explore the difference in the patterns at the ground, between gamma and proton induced showers, recurring to Artificial Neural Networks (ANNs). We compare the performance of ANNs to the performance of

The associate editor coordinating the review of this article and approving it for publication was Sotirios Goudos.

classic statistical approaches that resort to human-extracted features. To overcome the difficulty associated to the design of ANNs we use NeuroEvolution to automate the choice for the topology and learning strategy (Section III); in particular, we use Fast Deep Evolutionary Network Structure Representation ++ (Fast-DENSER++), detailed in Section IV. The results (Section V) show that the performance of the fittest network generated by Fast-DENSER++ surpasses the performance of classic statistical approaches. The gains in performance represent an improvement by a factor of up to 2.48; this indicates that with the same grid of sensors we can perform twice better than other methods; on the other hand, it can lead to investment savings because a smaller grid of detectors can be used.

II. GAMMA AND PROTON SIMULATION

The above-proposed investigations were done using gamma and proton (hadron) simulations, generated with CORSIKA [1], and an experiment layout as described in [2]. The detectors have been simulated with the Geant4 toolkit [3] and the recorded signals have been used to reconstruct the main shower characteristics (energy, direction, primary) so that the sensitivity of this experiment to gamma-ray sources could be evaluated realistically. The detector unit is composed of small water Cherenkov detectors (which maximizes the trigger efficiency), and segmented resistive plate chambers (which have a good time resolution providing in this way a good shower geometry reconstruction). This detector concept was chosen to lower the energy threshold of previous experiments and bridge the energy gap between satellite-bourne and present ground-based experiments.

The main aim of this work is to prove that the analysis of the pattern at the ground can be used to improve current gamma/hadron discrimination techniques. As such, for the present study we have opted to use only the information of the water-Cherenkov detectors (WCDs). Moreover, only showers reconstructed with energies between 1 and 1.7 TeV were used. Secondary shower particles that hit the WCD will produce light that can be recorded by photomultipliers mounted sideways. As such, for each shower event, a WCD station provides the following information: its position (x and y coordinates of the centre of the WCD), and the recorded signal (approximately proportional to the number of particles in it). It is only this information that shall be used to distinguish gamma from hadron induced showers.

In [2], it was demonstrated that this detector concept could perform the usual gamma/hadron discrimination. Two discrimination variables based solely in the WCD information were built: Compactness and S40. The former explores the information in the shower lateral distribution function (LDF), in particular, the steepness and bumpiness. This is done comparing the shower event LDF to a reference gamma LDF, built from the average of many gamma showers. The variable S40 is used to identify particle clusters away from the shower core. This is achieved by computing, for stations above 40 meters away of the reconstructed shower core, the ratio between the signal of the hottest station and the total signal. Although there is some level of correlation between the two variables, they carry independent information. To further explore the combined discrimination power of Compactness and S40, linear discriminant analysis is used, henceforth referred simply as Fisher. It is worth to mention that although the above quantities are certainly exploring the shower pattern at the ground, these classical statistics analysis cannot fully extract all the information due to the stochastic nature of the shower, forcing the use of non-parametric cuts.

III. NEUROEVOLUTION

NeuroEvolution (NE) [4] refers to the set of methods that apply Evolutionary Computation (EC) to automatically optimise ANNs. There are several NE approaches, which are often grouped according to the target of evolution. For example, Si *et al.* [5], David and Greental [6], and Morse and Stanley [7] optimise the *synaptic weights*, Shabash and Wiese *et al.* [8] search for the weights and activation functions, and Radi and Poli [9] evolve neural network *learning rules*. Differently, Soltanian *et al.* [10], Suganuma *et al.* [11], and Fernando *et al.* [12] search only the topology.

The separate optimisation of either the learning strategy or the topology has proven successful. On the one hand, NE has shown to be competitive with standard (non-evolutionary) learning algorithms [7], [13], and does not require the activation functions to be differentiable. On the other hand, when optimising the structure of the network, the evolutionary results match (and even surpass) the ones attained by grid or random search, given less computational time [14]. Nonetheless, Turner and Miller [15] state that "the choice of topology has a dramatic impact on the effectiveness of NE when only evolving weights; an issue not faced when manipulating both weights and topology", and therefore it is beneficial to evolve the topology and weights simultaneously. Examples of methods that simultaneously search for the best weights and topology are ANNA Eleonora [16], NeuroEvolution of Augmenting Topologies (NEAT) [17], or Cartesian Genetic Programming Artificial Neural Networks (CGPANN) [18].

The previous methods work well on the optimisation of the weights and topology of small scale networks, i.e., ANNs with few neurons; however, optimising hundreds, thousands or even millions of weights, and the topology of the network simultaneously is hard. That is the reason why the vast majority of the approaches that focus on the optimisation of deep networks [19]–[21] optimise the topology (e.g., number, type, and sequencing of layers), and the learning hyper-parameters rather than the weights, i.e., the methods focus on the optimisation of which learning algorithm to train the network (e.g., Backpropagation, or Adam), and its hyper-parameters (e.g., learning rate, or momentum).

One of the main drawbacks of NE concerns the time required for evaluating the population of candidate solutions. NE is based on EC, and thus a population of candidate solutions is evaluated throughout a (usually large) number

<fully-connected> ::=	layer:fc <activation>	(1)
	[num-units,int,1,128,2048 <bias>	(2)
<dropout> ::=	layer:droppu t [rate,float,1,0,0.7]	(3)
<activation> ::=	act:linear | act:relu | act:sigmoid	(4)
<bias> ::=	bias:True | bias:False	(5)
<softmax> ::=	layer:fc act:softmax num-units:10 bias:True	(6)
<learning> ::=	<bp> [batch_size,int,1,50,500]	(7)
	| <rmsprop> [batch_size,int,1,50,500]	(8)
	| <adam> [batch_size,int,1,50,500]	(9)
<bp> ::=	learning:gradient-descent [lr,float,1,0.0001,0.1]	(10)
	[momentum,float,1,0.68,0.99]	(11)
	[decay,float,1,0.000001,0.001] <nesterov>	(12)
<nesterov> ::=	nesterov:True | nesterov:False	(13)
<adam> ::=	learning:adam [lr,float,1,0.0001,0.1]	(14)
	[beta1,float,1,0.5,1] [beta2,float,1,0.5,1]	(15)
	[decay,float,1,0.000001,0.001]	(16)
<rmsprop> ::=	learning:rmsprop [lr,float,1,0.0001,0.1]	(17)
	[rho,float,1,0.5,1] [decay,float,1,0.000001,0.001]	(18)

FIGURE 1. Example of a grammar for encoding fully-connected networks.

of generations. To evaluate each candidate solution when the weights are not directly evolved we need to train the network, and when using large datasets the train process is time-consuming. Therefore, the networks are often trained for a fixed (low) number of epochs (e.g., 8-10 epochs). To overcome the burden of evolution we can use clusters of Graphic Processing Units (GPUs) (e.g., Amazon AWS, or Google Cloud) [22], evaluate the candidate solutions in a limited amount of data instances [7], or train for a fixed amount of epochs/time and let evolution resume the training in a subsequent generation by loading the previous weights [23].

In the current work, we use a variant of DENSER [20] to search for Convolutional Neural Networks (CNNs) to distinguish between gamma radiations and protons. DENSER, and the reasons for selecting it are detailed in Section IV.

IV. DEEP EVOLUTIONARY NETWORK STRUCTURED REPRESENTATION

DENSER [20], is a general-purpose grammar-based NE approach. It has successfully been applied to object detection tasks, and all the user inputs are defined in a human-readable format, and thus the framework is easy to adapt to different domains and network structures.

In DENSER, the individuals are encoded using a two-level representation: (i) the outer-level represents the macro-structure of the network, i.e., the sequence of evolutionary units;[1] and (ii) the inner-level keeps the parameters associated to the outer-level evolutionary unit. Whilst the outer-level is parameterised by the user-definition of an outer-level structure, the inner-level is parameterised by a Context-Free-Grammar (CFG). For example, for encoding a fully-connected network, with fully-connected and dropout layers the following outer-level structure can be defined: [((fully-connected, dropout), 1, 10), (softmax, 1, 1), (learning, 1, 1)]:[2] that is, the network structure is composed by between 1 and 10 fully-connected and/or dropout evolutionary units, 1 softmax evolutionary unit, and 1 learning evolutionary unit. The outer-level structure production-rules require a one-to-one mapping to the grammar that is used for the inner level. Figure 1 encodes an example of such grammar; there is a production rule for fully-connected, dropout, softmax, and learning. The grammar encodes the parameterisation required for each of the parameters of the evolutionary units; the parameters can be of one of the following types: integer, float or closed choice, and the parameter block has the following format [variable-name, variable-type, num_values, min_value, max_value].

The evolutionary engine of the inner-level of DENSER is based on Dynamic Structured Grammatical Evolution (DSGE): a variant of Grammatical Evolution (GE) [24] that solves its redundancy, and locality issues; there is a one-to-one mapping between the expansion possibilities and the production rules, and the genotype grows as needed, meaning that there are no non-coding parts in the genotype. For more details on DSGE the reader should refer to [25] and [26].

An example of a genotype and phenotype of an individual using the outer-level-structure [((fully-connected, dropout), 1, 10), (softmax, 1, 1), (learning, 1, 1)], and the grammar of Figure 1 is shown in Figure 2. The individual of the example is an ANN with 4 layers (2 fully-connected, 1 dropout, and 1 softmax) and the learning strategy. The inner-level representation follows the standard of DSGE, where the DSGE integer represents the expansion possibility; e.g., from Figure 1 we know that the activation non-terminal symbol has 3 expansion possibilities (linear, relu, or sigmoid), and therefore "DSGE: 1" on the example implies that we select the relu expansion (as evidenced on the phenotype). The inner-level genotype and the phenotype focus without the loss of generality on two evolutionary units.

To promote evolution, DENSER introduces genetic operators tailored explicitly for the manipulation of ANNs. The mutations enable the addition, duplication,[3] or removal of evolutionary units (at the outer-level), and the perturbation of any of the parameters and expansion possibilities (at the inner-level). The crossover swaps evolutionary units.

To assess the fitness of the individuals they are evaluated using either a fixed learning strategy (in case only the topology is the target of evolution), or the learning policy that constitutes an evolutionary unit (as in the grammar of Figure 1). The candidate solutions in DENSER are trained for a limited number of epochs (fixed to 10).

The following sub-sections detail two cumulative variants of DENSER: Fast-DENSER (Section IV-A), and Fast-DENSER++ (Section IV-B). These variants solve issues of

[1] In DENSER the evolutionary units correspond to all aspects of the network that are to be optimised, e.g., layers and learning strategy, but can also include data pre-processing and data-augmentation blocks.

[2] The outer-level structure defines the network sequencing using the following format: [(production-rules, min_evo_units, max_evo_units),...].

[3] Whilst the addition creates a new evolutionary unit, at random, the duplication performs a copy by reference, i.e., if during evolution any of the copies parameters' is changed all copies are affected.

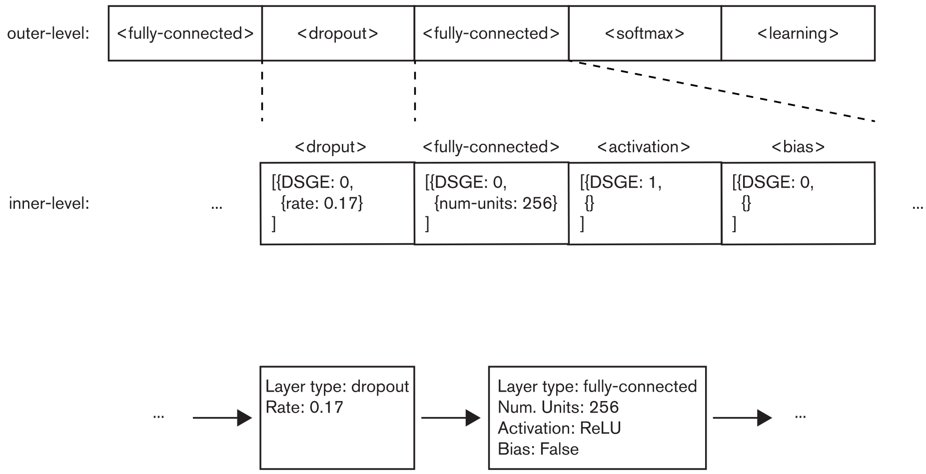


FIGURE 2. Example of a DENSER's genotype (top), and corresponding phenotype (bottom). The example is based on the outer-level structure [((fully-connected, dropout), 1, 10), (softmax, 1, 1), (learning, 1, 1)], and on the grammar of Figure 1.

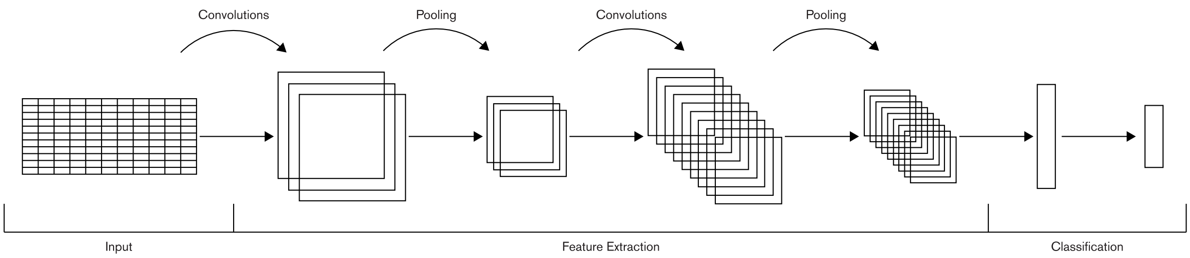


FIGURE 3. Topology of a convolutional neural network.

the standard DENSER version, that are pointed out in the following sections.

A. FAST-DENSER

The standard DENSER implementation follows the common guidelines of a Genetic Algorithm (GA), i.e., a population of individuals (often of size 100 or more) is evolved during a large number of generations. The evolutionary search requires a high number of evaluations, and thus slows down evolution; the end-goal of Fast-DENSER [27] is to speed up evolution. To accomplish that Fast-DENSER replaces the GA evolutionary procedure by a $(1 + \lambda)$-Evolutionary Strategy (ES); therefore, in each generation only $1 + \lambda$ individuals are evaluated. In addition, Fast-DENSER uses a different evaluation stop criteria; instead of training each individual for a fixed number of 10 epochs, Fast-DENSER tests the evaluation of the individuals up to a maximum GPU time, i.e., all individuals are granted access to the same computational resources. The train for a maximum granted GPU time makes the assessment of the learning strategy more adequate as more or fewer epochs can be performed depending on the requirements and topology of the network.

The authors compare DENSER (with a population size of 100 individuals) to Fast-DENSER (with $\lambda = 4$) on the evolution of CNNs; while in DENSER in each generation 100 individuals are evaluated, in Fast-DENSER only 5 individuals are evaluated. The results show that, on average, Fast-DENSER takes 20x less time than DENSER to generate the best solutions, and most importantly that the performance of the generated solution does not worsen.

B. FAST-DENSER++

Despite the speedup of Fast-DENSER over DENSER, the method is not able to generate networks that are ready for deployment right after the evolutionary process, i.e., during evolution the models are evaluated for a fixed number of epochs, or up to a maximum granted GPU time; nonetheless, further training may benefit the performance of the network.

Fast-DENSER++ [28] builds on top of Fast-DENSER by introducing a new mutation operator that modifies the maximum train time that is granted to each individual. The rationale is to increase the train time as the networks grow, i.e., during the initial generations the networks tend to be simple and therefore require less evaluation time; as time proceeds, the networks become more complex and may benefit from longer trains.

In the current paper, we conduct the experiments with Fast-DENSER++. We chose Fast-DENSER++ over other NE methods because of its plug-and-play configuration, where the user just needs to adapt the grammar to be able to

TABLE 1. Description of the dataset partitions.

Partition	#Gamma Instances	#Proton Instances
Train	22541	20261
Validation	1691	1519
Test	3945	3546
Generalisation	13879	12474

search for solutions for a specific problem, and for the ability of the method to generate high performing fully-trained ANNs, in a fraction of the time, and using considerably less computational resources than other approaches. The code for Fast-DENSER++ is publicly available under the Apache 2.0 license, and can be found in the github repository https://github.com/fillassuncao/f-denser, or as a docker image at https://hub.docker.com/r/fillassuncao/f-denser.

V. EVOLUTION OF CONVOLUTIONAL NEURAL NETWORKS

The gamma-ray detector, as described in Section II, is composed by 3m × 1.5m individual stations that occupy a full circle array with a radius of approximately 80m. Therefore, each event is a matrix with the recorded signal by each of the cells. The goal is to, based on the signal matrix, distinguish between gamma radiations and protons. CNNs [29] are adequate for analysing spatially-correlated data, and thus appropriate for this supervised pattern classification task.

CNNs are a type of Deep Learning (DL) model that learns from the raw data (the matrix of energies); it learns the feature space of the data, and then performs classification based on the acquired data representation. The typical structure of CNNs divides the hidden-layers into two major blocks: (i) a set of layers responsible for representation learning and feature extraction, which is formed by convolutional and pooling layers; and (ii) a set of layers for classification, where fully-connected layers are used (see Figure 3). Convolutional layers are composed by a set of learnable filters that are convolved with the layer's input signal; each filter connects locally (to what is known as the receptive field) to the input and is activated by different patterns, thus encoding a different feature. Pooling layers down-sample the input by aggregating neurons and consequently reduce the number of trainable parameters. Fully-connected layers densely connect to all neurons of the previous layer.

The design of CNNs requires the definition of: (i) the topology, i.e., the number of layers, type, sequencing, and parameterisation; and (ii) the learning strategy, i.e., the learning algorithm, and its parameterisation. Instead of hand-designing a CNN to solve our gamma-ray detection problem we use Fast-DENSER++ to automate the search.

The dataset description, the parameterisation of Fast-DENSER++, and the fitness function are respectively detailed in Sections V-A, V-B, and V-C. The experimental results are presented in Section V-D, and are discussed in Section V-E.

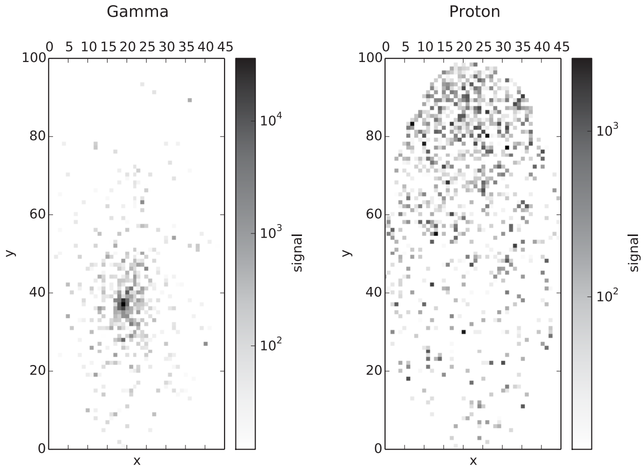


FIGURE 4. Example of the impact patterns of gamma and proton radiations.

A. DATASET

The dataset is composed of 79856 instances (shower events) of two disjoint classes: gamma or proton. Each instance is a 45 × 100 matrix, where each position represents the energy at a specific 3m × 1.5m cell of the circular grid of radius 80m. The positions of the matrix where there are no cells (because the grid is circular and the matrix is rectangular) are set to 0. An example of the impact patterns of gamma and proton radiations is depicted in Figure 4; the main difference on the ground impact patterns between gamma and proton radiations is that the dispersion of the signal of the gamma radiations tends to be more compact than the dispersion of the signal of the protons.

We partition the dataset into 4 independent sets. The first 3 are used during evolution:

- Train – used for training the individual with the evolved learning strategy;
- Validation – necessary for measuring the loss during the train, to perform early stopping;
- Test – applied to compute the fitness of the network after the training (guides evolution).

The last partition is used after the end of the evolutionary search and measures the generalisation ability of the models. If this partition was not created it would be impossible to perform an unbiased evaluation of the generated networks because evolution is conducted towards the test partition, and consequently it is expected that the networks perform well on it; that does not mean that they perform well beyond the data used during evolution. The number of instances of each partition is detailed in Table 1.

B. EXPERIMENTAL SETUP

To apply Fast-DENSER++ to the evolution of CNNs, first of all, we need to define the outer-level structure and the inner-level grammar. We use the outer-level structure: [(features, 1, 30), (classification, 1, 10), (softmax, 1, 1), (learning, 1, 1)], and the grammar of Figure 5. The search space

<features> ::= <convolution> | <convolution> (1)
| <pooling> | <pooling> (2)
| <dropout> | <batch-norm> (3)
<convolution> ::= layer:conv [num-filters,int,1,32,256] (4)
[filter-shape,int,1,2,5] [stride,int,1,1,3] (5)
<padding> <activation> <bias> (6)
<batch-norm> ::=layer:batch-norm (7)
<pooling> ::= <pool-type> [kernel-size,int,1,2,5] (8)
[stride,int,1,1,3] <padding> (9)
<pool-type> ::= layer:pool-avg | layer:pool-max (10)
<padding> ::= padding:same | padding:valid (11)
<classification> ::= <fully-connected> | <dropout> (12)
<fully-connected> ::= layer:fc <activation> (13)
[num-units,int,1,128,2048 <bias> (14)
<dropout> ::=layer:dropput [rate,float,1,0,0.7] (15)
<activation> ::= act:linear | act:relu | act:sigmoid (16)
<bias> ::= bias:True | bias:False (17)
<softmax> ::= layer:fc act:softmax num-units:2 bias:True (18)
<learning> ::= <bp> <stop> [batch_size,int,1,50,300] (19)
| <rmsprop> <stop> [batch_size,int,1,50,300] (20)
| <adam> <stop> [batch_size,int,1,50,300] (21)
<bp> ::= learning:gradient-descent [lr,float,1,0.0001,0.1] (22)
[momentum,float,1,0.68,0.99] (23)
[decay,float,1,0.000001,0.001] <nesterov> (24)
<nesterov> ::= nesterov:True | nesterov:False (25)
<adam> ::= learning:adam [lr,float,1,0.0001,0.1] (26)
[beta1,float,1,0.5,1] [beta2,float,1,0.5,1] (27)
[decay,float,1,0.000001,0.001] (28)
<rmsprop> ::= learning:rmsprop [lr,float,1,0.0001,0.1] (29)
[rho,float,1,0.5,1] [decay,float,1,0.000001,0.001] (30)
<stop> ::= [early_stop,int,1,5,20] (31)

FIGURE 5. Grammar used by Fast-DENSER++ for the evolution of CNNs to classify between gamma and proton radiations.

encompasses CNNs with between 3 and 41 layers, and all parameters including the learning strategy are encoded in the grammar.

Fast-DENSER++ parameters are summarised in Table 2. The table is divided into two independent sections: (i) evolutionary parameters – specify the evolutionary engine properties (number of generations, mutation rates, etc.); and (ii) train parameters – enumerate the learning parameters that are fixed for all networks. The default training time is of 10 minutes and can increase in multiples by mutation.

No data augmentation strategy is used, and the dataset is pre-processed by feature-wise centring and standard deviation normalization.

C. FITNESS FUNCTION

To assess the fitness of each individual, we evaluate the model in the test partition, and compute the true positive rate (TPR) and false positive rate (FPR) to build the Receiver Operating Characteristic (ROC) curve; we consider the positive class as the instances classified as a proton. The fitness of each individual of the population (ind) is computed as:

$$\text{fitness(ind)} = max\left(\frac{\text{TPR(x)}}{\sqrt{\text{FPR(x)}}}\right),$$

TABLE 2. Experimental parameters.

Evolutionary Parameter	Value
Number of runs	30
Number of generations	100
λ	4
Add layer rate	25%
Duplicate layer rate	15%
Remove layer rate	25%
Inner-level rate	15%
Train time rate	20%
Train Parameter	**Value**
Default train time	10 minutes
Loss	Categorical Cross-entropy

where TPR(x) and FPR(x) represent the TPR and FPR of the model at the point x of the FPR threshold, respectively. Since we are maximising, the models assigned with higher fitness values are those with a higher response of TPR for each FPR point, with emphasis to points with low FPR threshold.

The choice of the fitness function is related to the fact that the observation of astrophysical gamma-ray sources relies on the identification of gamma-rays which are immersed in a huge cosmic ray (hadronic) background. As the background is continuous and isotropic, while gamma-rays are localized in space, an excess of events coming from the gamma-ray sky region should be visible if one acquires during enough time. To state that there is an excess of events, the number of gamma-ray events has to be higher than the fluctuations of the background. As events are considered independent, the fluctuations follow the Poisson distribution, i.e., the square root of the number of events measured. By taking the number of background events much higher than the number of signal events, one can neglect the signal contribution in the square root which finally leads to the chosen fitness equation.

D. EXPERIMENTAL RESULTS

The analysis of the experimental results focuses on the performance of the evolved networks, measured on the evolutionary test set. The fitness function described in Section V-C is strictly related to the ROC curve, and thus in Figure 6 we depict the ROC curves (measured over the generalisation set) of the fittest networks that achieve the worse, median, and highest fitness values. The fittest networks are selected according to their fitness value on the test set.

The curve of the individual with the median fitness value is close to the curve of the best individual, indicating that the results are consistent, i.e., a high performing network is not discovered by chance, but is instead an outcome of the evolutionary search of Fast-DENSER++. The minimum, average, median, and maximum fitness values are 4.07, 5.78, 5.89, and 8.72, respectively.

Despite the importance of the analysis of the overall results, the ultimate goal is to select a model that is capable of addressing the problem we have at hand, in this case, a CNN which is capable of classifying between gamma and proton radiations. We select the best performing network according

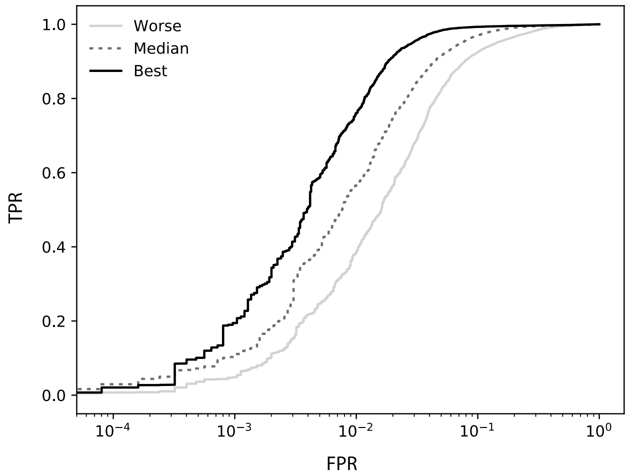


FIGURE 6. ROC curves of the worse, median, and best fittest individuals. A logarithmic scale is used.

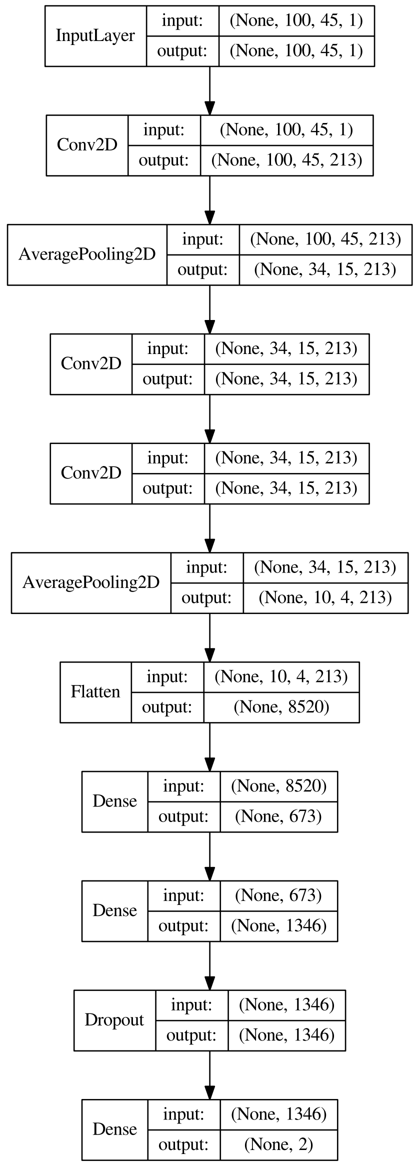


FIGURE 7. Topology of the fittest CNN discovered by Fast-DENSER++.

to the evolutionary test fitness. Recall that this choice is not biased because we will be later comparing the results based on a different, disjoint, set of instances.

The topology of the best performing network is shown in Figure 7. The CNN is composed by 8 hidden-layers: 3 convolutional, 2 average pooling, 2 fully-connected, and 1 dropout; as typical when the networks are designed by human practitioners the convolutional layers tend to be followed by pooling layers; on the other hand, the classification block of the network is formed by more hidden-layers than the usual, which demonstrates that evolution helps to generate novel and out of the box topologies that human-designers would hardly think of. The fittest CNN is trained using the Adam [30] learning algorithm with a learning rate of 0.0001, a beta 1 of 0.86486, a beta 2 of 0.68028; the learning rate decay is 0.00068, and the batch size is 117. The fittest CNN is compared with the performance obtained by other approaches in Section V-E.

Given the applicational type of the task, although it is not required for the network to perform in critical-time, it is important that it predicts fast. The network reports an average prediction time of approximately 3.12 ms, i.e., approximately 321 frames-per-second; this time includes the pre-processing. In order to be comparable, the experiments are conducted in a dedicated machine, with 4 NVIDIA 1080 Ti GPUs (each with 12GB), 64 GB of RAM, and an Intel Core i7-6850K @ 3.60GHz CPU. The predictions are carried out at the CPU level; GPUs are used for training.

E. DISCUSSION

Figure 8 compares the ROC curves of the fittest CNN discovered by Fast-DENSER++, with the performance reported by the classic statistics (Compactness, S40, and Fisher). To better analyse the evolutionary results we perform statistical tests to confirm whether or not the data follows a random distribution, i.e., to investigate the consistency of the results; in particular, we use the chi-square test. With a significant level $\alpha = 0.05$ the test reveals that the data does not follow a random distribution, and thus it highlights that the high performances tend to be consistent.

In addition to the fittest network we also investigate the performance of the ensemble formed by the best networks (one from each run); the generated networks are diverse in topology, and consequently they may be better suited to some patterns of inputs over others, i.e., while some of the networks can fail to predict a specific instance others can predict it correctly. The ensemble is formed by 16 voters, which are the CNNs that report a performance above the average performance of the 30 evolutionary runs (this choice is based on the evolutionary performance, and thus not biased); the predicted class is computed based on the maximum of the average confidences. For all methods, we measure the performance on the same partition of the data, and thus the results are comparable. The data is the same as in [2], but distinct from the one used in the evolutionary experiments;

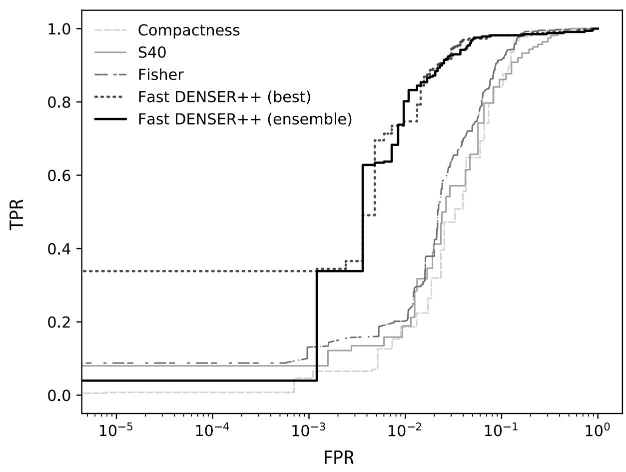


FIGURE 8. Comparison between the performance of CNNs discovered by Fast-DENSER++ (best and ensemble) and the performance other methods: Compactness, S40, and Fisher. A logarithmic scale is used. The models found with Fast-DENSER++ are represented by thicker lines.

nonetheless, it generated from the same source. It consists of 1158 instances: 328 gamma, and 830 protons. The dataset is unbalanced and follows the distribution expected in nature.

The analysis of the plot shows that the CNNs generated by Fast-DENSER++ surpass the results obtained by the classical statistics. Further, the average fitness of the evolved CNNs, the fitness of the fittest CNN,[4] and the fitness of the ensemble, Compactness, S40, and Fisher are of approximately 7.34, 10.01, 10.45, 3.13, 3.35, and 4.22, respectively. Comparing to the best result of the classic statistics, the average of all the generated CNNs, the fittest CNN, and the ensemble, improve the previous result by a factor of 1.71, 2.37, and 2.48, respectively. Recall that the results of evolution do not follow a random distribution (confirmed by statistics), and thus the comparison to the previous (deterministic) state-of-the-art results can be established based on the average performance of the evolutionary results.

VI. CONCLUSIONS AND FUTURE WORK

Gamma-ray detection helps to investigate extreme phenomena in the Universe, e.g., gamma-ray burst arising from fast rotating neutron stars or supermassive black holes. In this work, it is our objective to use deep learning to improve the gamma/hadron discrimination, based on the patterns they produce at ground impact. The impact patterns are stored as matrices of signal, where each position keeps the energy detected in a specific WCD. Therefore, we compare the performance of CNNs to the performance of the current state-of-the-art approach, based on classic statistics. The state-of-the-art approach is based on the design of features by expert practitioners; on the other hand, CNNs are a deep learning ANN that is known for its ability to learn to distinguish patterns based on the raw signal, i.e., it automatically learns the features and classifiers for the problem to be solved.

The problem associated to the deployment of CNNs is related to the design and parameterisation difficulties: the networks are composed by several layers, each with specific parameters; in addition to the definition of the layers and their sequence we require the choice for the most effective learning algorithm and its hyper-parameters. To overcome this challenge we use Fast-DENSER++ to automatically search for a capable CNN for our gamma-ray detection problem. This NE method was chosen because of its setup ease, and ability to generate ready to deploy networks using limited computational resources.

The results show that the CNN generated by Fast-DENSER++ is able to solve the gamma-ray detection problem, and it does so surpassing the performance reported by the previous classic methods, namely compactness, S40, and Fisher. Whilst the fittest CNN reports a fitness value of 10.01, the best performance of the classic methods is of 4.22, i.e., an improvement by a factor of 2.37. We also consider an ensemble formed by the best CNNs, which increases the performance from 10.01 to 10.45, i.e., an improvement by a factor of 2.48.

From all the above we can state that the main contribution of the current work is a CNN that establishes a new state-of-the-art result in the gamma/hadron discrimination problem, based on the ground impact patterns. The result is human-competitive: it outperforms the previous state-of-the-art without the need for experts nor to design the features nor to select the most appropriate topology and/or learning strategy for the model. In addition, the paper also contributes to the body of knowledge of Fast-DENSER++: the method is tested in a different domain, and using different metrics.

Future work will expand in two separate directions: (i) investigate the performance of Fast-DENSER++ in the search for CNNs for different primary energies; and (ii) study the impact of the detector configuration on the detection performance (e.g., number and shape/dimensions of the sensors). In terms of evolution, we will incorporate the number of layers and trainable parameters in the evolutionary objectives, with the rationale of generating more compact networks that may be easier to analyse and validate; this will be carried in a multi-objective fashion.

REFERENCES

[1] D. Heck, G. Schatz, T. Thouw, J. Knapp, and J. N. Capdevielle, *CORSIKA: A Monte Carlo Code to Simulate Extensive Air Showers*. Hannover, Germany: TIB Hannover, 1998.
[2] P. Assis, U. B. de Almeida, A. Blanco, R. Conceição, B. D. Piazzoli, A. D. Angelis, M. Doro, P. Fonte, L. Lopes, G. Matthiae, M. Pimenta, R. Shellard, and B. Tomé, "Design and expected performance of a novel hybrid detector for very-high-energy gamma-ray astrophysics," *Astroparticle Phys.*, vol. 99, pp. 34–42, May 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927650518300586
[3] S. Agostinelli, "Geant4—A simulation toolkit," *Nucl. Instrum. Meth.*, vol. 506, pp. 250–303, Jul. 2003.
[4] D. Floreano and P. Dürr, and C. Mattiussi, "Neuroevolution: From architectures to learning," *Evol. Intell.*, vol. 1, no. 1, pp. 47–62, 2008.
[5] T. Si, A. De, and A. K. Bhattacharjee, "Grammatical swarm for artificial neural network training," in *Proc. Int. Conf. Circuits, Power Comput. Technol.*, Mar. 2014, pp. 1657–1661.
[6] O. E. David and I. Greental, "Genetic algorithms for evolving deep neural networks," in *Proc. GECCO*, Jul. 2014, pp. 1451–1452.

[4]The fittest network is selected based on a different set of the dataset, and thus this choice is not biased.

[7] G. Morse and K. O. Stanley, "Simple evolutionary optimization can rival stochastic gradient descent in neural networks," in *Proc. GECCO*, Jul. 2016, pp. 477–484.

[8] B. Shabash and K. C. Wiese, "EvoNN: A customizable evolutionary neural network with heterogenous activation functions," in *Proc. Genetic Evol. Comput. Conf. Companion*, Jul. 2018, pp. 1449–1456.

[9] A. Radi and R. Poli, "Discovery of backpropagation learning rules using genetic programming," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1998, pp. 371–375.

[10] K. Soltanian, F. A. Tab, F. A. Zar, and I. Tsoulos, "Artificial neural networks generation using grammatical evolution," in *Proc. 21st Iranian Conf. Electr. Eng. (ICEE)*, May 2013, pp. 1–5.

[11] M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2017, pp. 497–504.

[12] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "PathNet: Evolution channels gradient descent in super neural networks," 2017, *arXiv:1701.08734*. [Online]. Available: https://arxiv.org/abs/1701.08734

[13] D. J. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms," in *Proc. IJCAI*. San Mateo, CA, USA: Morgan Kaufmann, 1989, pp. 762–767.

[14] P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, and J. R. Pastor, "Particle swarm optimization for hyper-parameter selection in deep neural networks," in *Proc. GECCO*, Jul. 2017, pp. 481–488.

[15] A. J. Turner and J. F. Miller, "The importance of topology evolution in neuroevolution: A case study using cartesian genetic programming of artificial neural networks," in *Proc. Int. Conf. Innov. Techn. Appl. Artif. Intell.*, Nov. 2013, pp. 213–226.

[16] V. Maniezzo, "Genetic evolution of the topology and weight distribution of neural networks," *IEEE Trans. Neural Netw.*, vol. 5, no. 1, pp. 39–53, Jan. 1994.

[17] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, 2002.

[18] A. J. Turner and J. F. Miller, "Cartesian genetic programming encoded artificial neural networks: A comparison using three benchmarks," in *Proc. GECCO*, Jul. 2013, pp. 1005–1012.

[19] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, and B. Hodjat, "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. New York, NY, USA: Academic, 2019, pp. 293–312.

[20] F. Assunção, N. Lourenço, P. Machado, and B. Ribeiro, "DENSER: Deep evolutionary network structured representation," *Genetic Program. Evolvable Mach.*, vol. 20, no. 1, pp. 5–35, Mar. 2019. doi: 10.1007/s10710-018-9339-y.

[21] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Trans. Evol. Comput.*, to be published. doi: 10.1109/TEVC.2019.2916183.

[22] J. Liang, E. Meyerson, B. Hodjat, D. Fink, K. Mutch, and R. Miikkulainen, "Evolutionary neural automl for deep learning," 2019, *arXiv:1902.06827*. [Online]. Available: https://arxiv.org/abs/1902.06827

[23] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 694–713, May 1997.

[24] M. O'Neill and C. Ryan, "Grammatical evolution," *IEEE Trans. Evol. Comput.*, vol. 5, no. 4, pp. 349–358, Aug. 2001.

[25] F. Assunção, N. Lourenço, P. Machado, and B. Ribeiro, "Towards the evolution of multi-layered neural networks: A dynamic structured grammatical evolution approach," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2017, pp. 393–400.

[26] N. Lourenço, F. Assunção, F. B. Pereira, E. Costa, and P. Machado, "Structured grammatical evolution: A dynamic approach," in *Handbook Grammatical Evolution*. New York, NY, USA: Springer, 2018, pp. 137–161.

[27] F. Assunção, N. Lourenço, P. Machado, and B. Ribeiro, "Fast DENSER: Efficient deep neuroevolution," in *Proc. Eur. Conf. Genetic Program.*. New Yrok, NY, USA: Springer, Mar. 2019, pp. 197–212.

[28] F. Assunçao, N. Lourenço, P. Machado, and B. Ribeiro, "Fast-denser++: Evolving fully-trained deep artificial neural networks," 2019, *arXiv:1905.02969*. [Online]. Available: https://arxiv.org/abs/1905.02969

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

FILIPE ASSUNÇÃO received the B.Sc. degree in informatics engineering from the University of Coimbra (UC), in 2013, and the M.Sc. degree in informatics engineering, in 2015.

He was a Visiting Researcher with the Natural Computing Research and Applications Group, University College Dublin, Ireland, for six months. He is currently a Researcher with the Computational Design and Visualization Lab, Cognitive and Media Systems Group, Centre for Informatics and Systems, UC, where he is currently enrolled in the Doctoral Program in Information Science and Technology. His main interests include artificial intelligence, machine learning, and neuroevolution. He is the Creator of the Deep Evolutionary Structured Representation (DENSER). He has authored and co-authored several articles in conferences and journals, and is involved in national and international projects and collaborations.

JOÃO CORREIA received the M.Sc. and B.S. degrees in informatics engineering, and the Ph.D. degree in information science and technology from the University of Coimbra.

He is currently an Invited Assistant with the University of Coimbra, and also a Researcher of the Computational Design and Visualization Lab, Cognitive Media Systems Group, Centre for Informatics and Systems. He is also an Invited Assistant at the Lusófona University of Lisbon. His main research interests include evolutionary computation, machine learning, neuroevolution, computer vision, and computational creativity. He is involved in different International Program Communities of International conferences of the area of evolutionary computation, artificial intelligence and computational art and creativity. He was also the Chair of the International Conference of Evolutionary Art Music and Design conference. Furthermore, he has authored and coauthored several articles on the different international conferences and journals on artificial intelligence and evolutionary computation and is involved in national and international projects.

RÚBEN CONCEIÇÃO is currently a Researcher with the Portuguese national organization for Particle and Astroparticle Physics (LIP), and also an Invited Professor with the Physics Department, Instituto Superior Técnico, University of Lisbon.

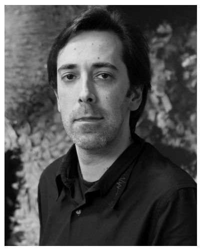
He is also a High-Energy Physicist and Astrophysicist. He is an Active Member of the Pierre Auger Collaboration, since 2006, dedicated to the study of ultra-high-energy cosmic rays. He is also currently one of the lead persons for the construction of a wide field observatory in the Southern hemisphere for the detection and study of high-energy gamma-rays. His original scientific contributions have been mostly related with the study of high-energy hadronic interactions, understanding, and modeling of extensive air showers and the conceptual development of innovative detection techniques and methods for cosmic ray physics. He has been teaching high-energy particle physics and astroparticle physics in master's courses in the University in Lisbon and in several international Ph.D. schools. He has been a Visiting Scientist with the Santiago de Compostela University, and have periodic scientific stays in the Pierre Auger Observatory at Malargüe, Argentina.

MÁRIO JOÃO MARTINS PIMENTA born in Lisbon, Portugal, in 1957. He received the Engineer degree, in 1980, the Ph.D. degree in physics, in 1986, and the Aggregation in physics, in 1992.

He is currently a Full Professor of physics with the Instituto Superior Técnico, Lisbon, and the President of the Laboratório de Instrumentação e Física Experimental de Partículas. He is a Representative for Portugal at the Pierre Auger Observatory, and the Coordinator and co-founder of the IDPASC Doctoral Network. His research interests are particle and astroparticle physics, in particular with ultra-high energy cosmic rays and gamma-rays, experimental high-energy physics, searches for new physics, Hadronic interactions, and detectors for high energy physics and space applications. Since 1989, he has been a Principal Investigator of many projects in the framework of Portuguese participation at CERN and at ESA and in cosmic rays experiments. He has supervised 13 Ph.D. theses and of many master's theses. He is a member of the Pierre Auger Observatory, a former member of the EUSO Space Project, and of DELPHI, NA38, WA74, and WA72 experiments at CERN. He has coauthored more than 500 scientific articles and has authored many invited scientific talks and seminars. He has coauthored a physics textbook for master's and Ph.D. students named the *Introduction to Particle and Astroparticle Physics* (Springer, first and second editions), and a physics textbook for undergraduates named the *Introdução à Física* (McGrawHill, first and second editions), and the *Escolar Editora*, (third edition). He is the organizer of more than 40 International Conferences and Workshops. He has also coauthored two stamp collections for Macau Post, China (Science and Technology—Standard Model of Particle Physics, 2002; Science and Technology—Cosmology XXI, 2004). He is a participant in many outreach activities.

BERNARDO TOMÉ received the Ph.D. degree from the University of Lisbon, in 1998. He is currently an Assistant Researcher with the Laboratório de Instrumentação e Física Experimental de Partículas (LIP), and an Invited Teacher with the Physics Department, Instituto Superior Técnico, University of Lisbon. His research interests cover the areas of experimental high-energy particle and astroparticle physics, in particular ultra-high-energy cosmic-rays and gamma-ray physics; detector development and performance studies for high-energy physics and space applications; and advanced Monte Carlo detector simulation techniques. He has been a member of several international collaborations, namely the DELPHI experiment at LEP/CERN, the EUSO space project, as well as collaborating in several projects in the framework of the LIP participation at ESA. He is currently a member of the Pierre Auger Observatory and of the GEANT4 collaboration at CERN. He is also involved in the development of a new wide Field-of-View Gamma Ray Observatory in the Southern Hemisphere. He has co-authored more than 400 scientific papers.

NUNO LOURENÇO received the Ph.D. degree in information science and technology from the University of Coimbra, in 2016, where he is currently an Assistant Professor (non-Tenure). He has been with the Centre for Informatics and Systems, University of Coimbra (CISUC), since 2009, as a member of the Evolutionary and Complex Systems (ECOS) Group. Formerly, he was appointed as a Senior Research Officer of the University of Essex, U.K. His main research interests are in the areas of bio-inspired algorithms, optimization, and machine learning. He is the creator of the Structured Grammatical Evolution, a genotypic representation for Grammatical Evolution, and participated in the proposal of a novel approach to automatically design Artificial Neural Networks (ANNs) using Evolutionary Computation called DENSER. He has served as the publication chair for two of the most important conferences of the Evolutionary Computation field, namely the PPSN XV and EuroGP 2019. He will be the Co-Chair of the EuroGP 2020. He is a member of the Programme Committee of some of the main conferences in the field of Evolutionary Computation; a member of the Steering Committee of EuroGP; and an Executive Board Member of SPECIES. He has authored or co-authored several articles in journals and top conferences from the Evolutionary Computation and Artificial Intelligence areas, and has been involved as a Researcher in many national and international projects.



PENOUSAL MACHADO is currently an Associate Professor with the Department of Informatics, University of Coimbra, Portugal. He is also the Deputy Director of the Centre for Informatics and Systems, University of Coimbra (CISUC), the Coordinator of the Cognitive and Media Systems Group, and the Scientific Director of the Computational Design and Visualization Lab, CISUC. His research interests include evolutionary computation, computational creativity, artificial intelligence, and information visualization. He is the author of more than 200 refereed journal and conference papers in these areas, and his peer-reviewed publications have been nominated and awarded multiple times as the best paper. Until May 2019, his publications gathered over 2262 citations, an h-index of 24, and an i10-index of 51. He was the advisor of seven Ph.D. thesis and 39 M.Sc. thesis, and is currently advising 9 Ph.D. and 3 M.Sc. thesis. He is also the Chair of several scientific events, including, amongst the most recent, ICCC 2020, PPSN XV, and EvoStar 2016; a member of the Programme Committee and Editorial Board of some of the main conferences and journals in these fields; a member of the Steering Committee of EuroGP, EvoMUSART, and Evostar; and an executive board member of SPECIES. He is a recipient of several scientific awards, including the prestigious EvoStar Award for outstanding Contribution to Evolutionary Computation in Europe, and the Award for Excellence and Merit in Artificial Intelligence granted by the Portuguese Association for Artificial Intelligence. He has been invited to perform keynote speeches in a wide set of domains, from evolutionary computation to visualization and art. His work was featured in the Leonardo journal, Wired magazine, and presented in venues such as the National Museum of Contemporary Art (Portugal) and the ''Talk to me'' exhibition of the Museum of Modern Art, NY, USA (MoMA).