A novel hybrid differential evolution strategy applied to classifier design for mortality prediction in adult critical care admissions

SHENFIELD, Alex <http://orcid.org/0000-0002-2931-8077>, RODRIGUES, Marcos <http://orcid.org/0000-0002-6083-1303>, MORENO-CUESTA, Jeronimo and NOORELDEEN, Hossam

Published version

SHENFIELD, Alex, RODRIGUES, Marcos, MORENO-CUESTA, Jeronimo and NOORELDEEN, Hossam (2017). A novel hybrid differential evolution strategy applied to classifier design for mortality prediction in adult critical care admissions. In: CIBCB 2017 : IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology, Manchester, 23-25 August 2017. (In Press)

A novel hybrid differential evolution strategy applied to classifier design for mortality prediction in adult critical care admissions

Alex Shenfield* and Marcos Rodrigues
Geometric Modelling and Pattern Recognition Group
Sheffield Hallam University
Sheffield
UK
Email: *a.shenfield@shu.ac.uk

Hossam Nooreldeen and Jeronimo Moreno-Cuesta
North Middlesex University Hospital
Enfield
London
UK

***Abstract*—The optimisation of classifier performance in pattern recognition and medical prognosis tasks is a complex and poorly understood problem. Classifier performance is greatly affected by the choice of artificial neural network architecture and starting weights and biases - yet there exists very little guidance in the literature as to how to choose these parameters. Recently evolutionary artificial neural networks have been proposed to mitigate some of these problems; however, whilst evolutionary methods are extremely effective in finding global optima, they are notoriously computationally expensive (often requiring tens of thousands of function evaluations to arrive at a solution).**

This paper proposes a novel hybrid adaptive approach to the optimisation of artificial neural network parameters where the global search capabilities of differential evolution and the efficiency of local search heuristics (such as resilient back-propagation for artificial neural network training) are combined. A state-of-the-art adaptive differential evolution algorithm, JADE, has been chosen as the basis for this hybrid algorithm due to its proven effectiveness in optimising high dimensional problems. The performance of this hybrid adaptive differential evolution algorithm is then demonstrated in the design of a classifier for mortality risk prediction in a critical care environment, where the optimised classifier is shown to outperform the current state-of-the-art in risk prediction.

I. Introduction

The performance of artificial neural networks (ANNs) in classification tasks is influenced to a great degree by the choice of network architecture and starting weights. However, there exists little guidance in the literature in how to choose these parameters. Although numerous gradient descent based local search methods (such as back-propagation of errors) exist for tuning the weights and biases within an ANN, the success of these methods depends on the choice of starting parameters as these local search techniques are prone to premature convergence to local optima [8].

In the last few years there has been increasing interest in the use of soft computing global optimisation techniques such as evolutionary algorithms (EAs) to provide a solution to the problem of parameter choice in artificial neural network design [33]. EAs are an effective choice for this kind of optimisation problem as they are capable of dealing with large, non-differentiable, and multi-modal search spaces. However, one drawback to the use of evolutionary algorithms is that they typically require a large number of function evaluations to arrive at a solution. This can be a significant issue with computationally expensive objective functions such as those involving the evaluation of classifier performance.

The purpose of this paper is to introduce a novel hybrid adaptive differential evolution based approach to optimising artificial neural network parameters. This novel algorithm will then be applied to the design of a classifier for a real-world mortality prediction problem. Results of this classifier will then be compared to the current state-of-the-art models for risk prediction in critical care environments in terms of both the discrimination of the classifier and the overall classification accuracy achieved.

The paper is organised as follows: section II will provide a brief introduction to artificial neural networks and evolutionary algorithms, and then section III will introduce the novel design of a hybrid adaptive differential evolution based approach for the meta-heuristic optimisation of artificial neural network parameters. Section IV will outline the mortality risk prediction problem considered in this paper and how the proposed evolutionary approach to artificial neural network optimisation performs against currently used risk prediction models in a critical care environment. Finally, section V will present some conclusions and outline some ideas for further work.

II. Background

A. *Evolutionary algorithms*

Evolutionary algorithms (EAs) are an optimisation technique utilising some of the mechanisms of natural selection [7]. They are an iterative, population based method of global optimisation capable of both exploring the solution space of the problem and exploiting previous generations of solutions. The exploitation of the previous generation of solutions is primarily performed by a selection operator that gives preference to those solutions which have performed well when creating the next generation of solutions to be evaluated. Mutation and recombination allow the optimisation algorithm to explore the solution space, and help to ensure the robustness of the algorithm by preventing the algorithm from getting stuck in local optima.

EAs evaluate candidate solutions based on pay-off information from the objective function, rather than derivative information or auxiliary knowledge. This ensures that they are applicable to many different problem domains, including those where conventional optimisation techniques (such as hill-climbing) may fail. As evolutionary algorithms maintain a population of candidate solutions, each generation contains more information about the shape of the fitness landscape than would be available to conventional, non-population based methods such as hill-climbing [20]. This helps to ensure that EAs are robust in the presence of noise. However, one of the drawbacks of the population based nature of evolutionary algorithms is that the optimisation process frequently requires many function evaluations.

In the last two decades the differential evolution algorithm [26] has emerged as one of the most popular and powerful real-valued optimisation algorithms [4], consistently achieving highly in benchmark competitions such as those run as part of the Congress on Evolutionary Computation (CEC). One of the key strengths of differential evolution is the difference vector based mutation strategies that aim to guide the formation of child solutions towards promising areas of the search space [35], [26]. Several of these mutation strategies have been proposed by various authors [35], [6], [18] with elitist mutation mechanisms shown to improve convergence speed (but at the risk of converging prematurely to non-global optima due to the reduced population diversity).

B. *Artificial neural networks*

Artificial neural networks are a type of machine learning algorithm consisting of interconnected processing units inspired by the operation of biological neurons in the brain and central nervous system [17], [29]. As with biological neurons, the processing units in an ANN “learn” by adjusting the connections between each other in response to differences between expected and observed behaviour. This self-adaptive nature means that ANNs are capable of detecting complex relationships between variables without the need for prior knowledge [36].

In contrast to traditional classification techniques such as linear regression and discriminant analysis, ANNs provide a “model-free” approach capable of adapting to highly complex and non-linear underlying statistical models of the system [39]. This makes them particularly useful in applications such as decision support for medical diagnosis [12] where difficulties in model building associated with conventional classifiers such as linear regression, k-nearest neighbour, and decision tree algorithms can hamper the design of robust classifiers [5].

One of the key drawbacks in the use of ANN models is the difficulty both in choosing an appropriate network structure for the classifier and in refining the weights and biases for the interconnected artificial neurons within the network. Both of these factors have been shown to exert a significant influence on the overall performance of a classifier [9]. Often the weights and biases of the interconnected neurons within an ANN are tuned using gradient descent based back-propagation of error methods or those based on the Levenberg-Marquardt algorithm - however, these can be susceptible to convergence to local optima. Some authors have proposed strategies for dynamically growing [1] or pruning [27] the topology of an artificial neural network as part of the training process, but these add significant complexity to the training process and are predominantly gradient descent based so suffer from the same problems with premature convergence to local optima as the tuning of weights and biases.

Over the last two decades there has been increasing interest in evolutionary artificial neural networks (EANNs) that are capable of overcoming some of these problems by performing either parametric or structural learning [38]. Parametric EANNs focus on using evolutionary methods to optimise the weights and biases within a network [31], whilst EANNs for structural learning focus on evolving the ANN topology (i.e. the number and size of the hidden layers within the network) [13]. More recently there has been significant work in the development of EANN algorithms capable of performing both parametric and structural learning of an ANN [33], with promising results in their application to function approximation problems.

C. *Hybrid optimisation and memetic computing*

In many real-world optimisation problems, the evaluation of candidate solutions is computationally expensive. This hampers the application of population based global search methods, such as evolutionary algorithms and differential evolution, which frequently require large numbers of function evaluations to arrive at a solution (a typical evolutionary algorithm might run for 10,000 - 30,000 function evaluations). For non-trivial real-world problems this process can be prohibitively computationally expensive.

Several approaches to mitigating this issue have been proposed, from exploiting the inherent parallelism in population based global optimisation algorithms using distributed high performance computing resources [32] to hybrid and memetic algorithms [22]. These hybrid and memetic algorithms focus on exploiting all available knowledge about a problem under consideration [21] by blending population based search methods and local search algorithms. This blend of both global and local search has been shown to significantly accelerate the discovery of good solutions in the search landscape [24].

III. A HYBRID DIFFERENTIAL EVOLUTION BASED APPROACH TO THE OPTIMISATION OF ARTIFICIAL NEURAL NETWORKS

A. *JADE: an adaptive differential evolution algorithm with external archive*

Differential evolution (DE) is a form of evolutionary algorithm that has been shown to be effective for many real-world optimisation problems [34], [35], [26]; however, like all evolutionary algorithms, its performance is heavily dependent on the selection of control parameters [6]. Whilst some general guidelines exist for choosing these parameters [6], [19], the optimal parameter settings are highly dependent on the problem under consideration. To overcome this limitation, a state-of-the-art adaptive differential evolution algorithm, JADE [40], has been proposed. The JADE algorithm adapts the crossover probability, CR, and the mutation factor, F, according to equations 1 and 2 respectively.

$$CR_i = randn_i(\mu_{cr}, 0.1) \tag{1}$$

$$F_i = randc_i(\mu_f, 0.1) \tag{2}$$

where μ_{cr} and μf are updated at the end of every generation using equations 3 and 4. Note that in equation 2, $randc_i$ is a random number generated using a Cauchy distribution.

$$\mu_{cr} = (1 - c) * \mu_{cr} + c * mean_A(S_{cr}) \tag{3}$$

$$\mu_f = (1 - c) * \mu_f + c * mean_L(S_f) \tag{4}$$

In these equations, c is a positive constant between 0 and 1 which determines the adaptation rate of μ_{cr} and μ_f, $mean_A$ is the arithmetic mean, $mean_L$ is the Lehmer mean, and S_{cr} and S_f are the set of all successful crossover and mutation factors in the previous generation.

The values of CR and F are then used in conjunction with a greedy mutation strategy, called DE/current-to-*p*best/1 (shown in equation 5) and the standard DE crossover operator (shown in equation 6). To avoid problems with premature convergence that often occur with greedy mutation strategies in differential evolution, the diversity of the population is increased by maintaining an external archive of recently explored inferior solutions and this is used in the variation process.

$$v_{i,g} = x_{i,g} + F_i(x^p_{best,g} - x_{i,g}) + F_i(x_{r1,g} - \widetilde{x}_{r2,g}) \tag{5}$$

where $\widetilde{x}_{r2,g}$ is a randomly selected individual from the combination of the archived solutions and the current generation of candidate solutions.

$$u_{i,j,g} = \begin{cases} v_{i,j,g}, & \text{if } rand(0,1) \leq CR_i \text{ or } j = j_{rand} \\ x_{i,j,g}, & \text{otherwise} \end{cases} \tag{6}$$

where $rand(0, 1)$ is a uniformly random number between 0 and 1 (independently generated for each gene in the child solution), and j_{rand} is a randomly generated integer generated once for each child solution that ensures at least one mutated gene makes it in to the child solution.

The JADE algorithm with external archive has been shown to perform well across a range of benchmark problems and shown particular promise in high dimensional optimisation problems [40]. On the selection of thirteen 100-dimension benchmark problems used in [40] the JADE algorithm with external archive significantly outperformed other algorithms (including JADE without external archive) in eleven of the test problems and performed comparably in the other two.

This state-of-the-art adaptive differential evolution algorithm has been augmented with back-propagation to form the hybrid approach outlined in section III-B. Several authors [25], [23] have shown that incorporating local search methods significantly improves the rate of convergence of evolutionary optimisation methods - a factor that is increasingly important when evaluating complex and computationally expensive objective functions such as classifier performance.

B. A hybrid differential evolution optimisation approach

The hybrid adaptive differential evolution approach outlined in this paper incorporates the resilient back-propagation learning heuristic [28] within the JADE optimisation algorithm to perform local search on the ANN parameters found by the global optimisation algorithm. This hybrid approach works by generating a set of candidate solutions (i.e. potential ANN parameters) using the global optimiser and then using these to build an artificial neural network. This ANN is then trained on a subset of the data using resilient back-propagation, and the trained weights and biases used to replace the original ones in the candidate chromosome. This chromosome is then copied back into the global population and used in mutation and crossover in the global JADE algorithm. This local search and evaluation procedure is shown in pseudo-code in algorithm 1.

Algorithm 1 Pseudo code of the local search heuristic

```
procedure EVALUATION AND LOCAL SEARCH(pop)
    for $i = 1, NP$ do
        ($arch_i$, $wb_i$) = decode_chromosome($pop_i$)
        $net_i$ = neural_network($arch_i$, $wb_i$)
        train_rprop($net_i$)
        $fitness_i$ = evaluate($net_i$)
        $wb\_optimal_i$ = get_wb($net_i$)
        $pop_i$ = encode_chromosome($wb\_optimal_i$)
    end for
end procedure
```

The metric used in the fitness evaluation of each candidate solution can vary depending on how the performance of the classifier is to be measured. The classifier design in section IV of this paper is evaluated in terms of the area under the receiver operator characteristic curve - a measure that is commonly used in medical classification as it captures the discrimination performance of a classifier (i.e. how effectively a classifier can determine one class from another). However, the overall classification accuracy or misclassification rate could just as easily be used.

Figure 1 illustrates how this local search heuristic is incorporated into the JADE algorithm. Initially a random population is created and evaluated (using algorithm 1). The crossover rate, CR, and mutation factor, F, are then generated according to equation 3 and 4 and the JADE greedy mutation operator and DE crossover operator are applied to create the child population. This child population is then evaluated and optimised (using algorithm 1) before the best performing individuals are selected to form the next generation and the inferior solutions added to the archive. The information about which crossover and mutation rates produced successful solutions is then used to adapt the distribution parameters used in generating CR and F for the next generation.

IV. Mortality prediction in adult critical care admissions

A. Problem description

Over the last three decades, there has been significant work carried out in developing risk prediction models for patients admitted to critical care units [15], [14], [16]. The most recent of these is the Intensive Care National Audit and Research Centre (ICNARC) model developed in 2007 [11] which uses data from 231,900 admissions to 163 critical care units across the UK to develop and validate a UK based model outperforming previous approaches such as the APACHE-II score [11]. This section aims to show that by using the proposed hybrid adaptive differential evolution approach to artificial neural network classifier design it is possible to develop a

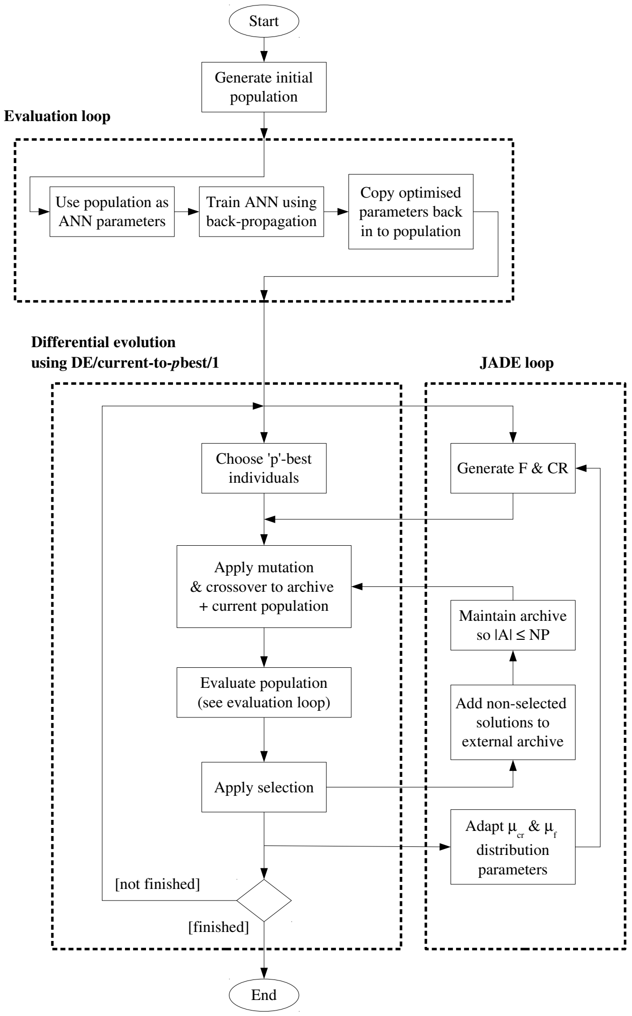

Start
Generate initial population
Evaluation loop
Use population as ANN parameters
Train ANN using back-propagation
Copy optimised parameters back in to population
Differential evolution using DE/current-to-pbest/1
JADE loop
Choose 'p'-best individuals
Generate F & CR
Apply mutation & crossover to archive + current population
Maintain archive so |A| ≤ NP
Evaluate population (see evaluation loop)
Add non-selected solutions to external archive
Apply selection
Adapt μ_{cr} & μ_f distribution parameters
[not finished]
[finished]
End

Fig. 1. Illustration of the hybrid differential evolution approach to ANN optimisation

classifier that can significantly outperform the current-state-of-the-art in risk prediction for critical care environments.

B. Data collection and preparation

This research was conducted using the ICNARC data collected at North Middlesex University Hospital over the period from January 1st 2012 until April 30th 2014. The original dataset consisted of 1,074 patient records, of which 432 records were excluded because of missing data.

The data used in this study consisted of 11 physiological variables obtained during the first 24 hours of admission to the ICU (shown in Table I). This included 10 of the 12 physiological variables used in the ICNARC model, but with the FiO2 and PaO2 levels separated out and applied as separate inputs to the classifier (following a similar approach to that used by Wong and Young [37]).

As well as these physiological variables, some additional patient information was also collected and used in the classification process. This consisted of:

TABLE I. Physiological variables used as classifier inputs

1	Maximum heart rate
2	Lowest systolic blood pressure
3	Highest temperature
4	Lowest non ventilated respiratory rate
5	Lowest PaO2
6	FiO2 (at the time of the lowest PaO2 level)
7	Highest serum urea
8	Highest serum sodium
9	Urine output
10	Lowest white blood cell count
11	Lowest Glasgow Coma Score

- patient's age at admission to the ICU
- whether the patient had CPR within the 24 hours prior to ICU admission
- whether the patient was intubated during the first 24 hours
- the source of admission (e.g. the A&E department within the same hospital)

The input data was preprocessed in two stages. Firstly the categorical variables (whether the patient received CPR prior to admission, whether the patient was intubated during the first 24 hours of admission, and the source of admission) were converted into numerical representations suitable for use in the ANN. Secondly, the other continuous variables (i.e. the 11 physiological variables and the patient's age on admission to the ICU) were standardised by subtracting the mean of the each variable and dividing by the standard deviation. Many authors [2], [30] have shown that standardising the input variables to a neural network significantly improves the performance of the training process by improving the numerical condition of the optimisation problem.

The key outcome considered in this study was the mortality status on discharge from hospital as this is the same outcome used in the ICNARC risk prediction model and the APACHE-II score [14], [11].

The performance of the optimised ANN classification model was analysed and compared to the ICNARC model using both Receiver-Operator Characteristic (ROC) curves and classification tables. ROC curves were used to analyse the trade-off between sensitivity and specificity of both models across different classification thresholds. The area under the ROC curve was then used to characterise the overall discrimination of the considered classification models (with a higher value for the area under the ROC curve indicating the classifier is better at distinguishing between the two different classes). In many medical diagnosis problems the discrimination of a classifier (measured by the AUC of the ROC curve) is considered of more interest than the overall classification accuracy [10] as it provides a better measure of classifier robustness and is less affected by unbalanced datasets. In medical diagnosis problems datasets are often unbalanced since, when dealing with minority outcomes such as mortality prediction or instances of disease, there are frequently many more healthy / normal patients than abnormal ones.

Classification tables were then used to analyse the trade-off between the true positive rate (the proportion of patients in the sample who are correctly predicted to die) and the false positive rate (the proportion of patients who are predicted to die but survive) at 10% decision criterion intervals (from 0-100%). The overall classification accuracy (again at 10%

decision intervals for 0-100%) was also used to analyse the performance of both models.

C. Encoding the problem

In order to use the proposed hybrid adaptive differential evolution algorithm to optimise the topology and weights of the ANN classifier for mortality prediction in a critical care environment, the ANN structure and parameters must be encoded into a real-valued chromosome, which can then be optimised using differential evolution and then decoded for evaluation. Figure 2 illustrates the chromosome structure used to store the encoding of an ANN with 2 output neurons, a maximum of 2 hidden layers, and 15 input neurons.

For the ANN used in this network, each candidate solution contains 784 variables, with the first 2 defining the number and size of the hidden layers, the following 300 variables defining the weights for the input layer, 400 for the first hidden layer, and 40 for the third and final hidden layer.

Regardless of the topology of the candidate solution ANN (which in this case is defined by the first two genes of the encoded chromosome) the maximum number of weights and biases will be stored with each chromosome. However, not all these weights and biases will necessarily be used to create the candidate ANN (as the candidate ANN might have fewer than the maximum hidden layers) and may lay dormant until the first two genes indicate that they are needed. This introduces the interesting feature of atavism[1].

At each function evaluation, a chromosome is decoded from its encoded state (as described in Figure 2) and used to instantiate an ANN. This ANN is then trained using the local search operator (resilient back-propagation) and used to classify the training data. Following the local search procedure, the trained weights are copied back into the population (in place of the originals) for use in variation. The result of the classification is then used as the fitness function when deciding which individuals to select for the next generation.

The hybrid adaptive differential evolution algorithm was run using the parameters shown in Table II. Unlike conventional differential evolution, there is a minimal number of parameters to set in the JADE algorithm due to its adaptive nature. Those parameters that are needed (c, which sets the adaptation rate of μ_{cr} and μ_f, and p, which controls how greedy the mutation strategy is) are problem insensitive. [40] has shown that parameters in the ranges of $c \in [0.05, 0.2]$ and $p \in [0.05, 0.2]$ are particularly effective.

TABLE II. OPTIMISER PARAMETERS

Maximum function evaluations	50,000
Population size	100
c	0.1
p	0.05
Initial μ_{cr}	0.5
Initial μ_f	0.5

D. Results and discussion

A comparison between the ROC curves for both the ICNARC model and the artificial neural network classification model is presented in Figure 3 with point wise confidence intervals shown for each 5% false positive rate threshold. The resulting area under the curve (AUC) metrics for the ROC curves of both the ICNARC model and the optimised artificial neural network model (found using 10 fold cross validation of the classifier) were 0.8080 and 0.9320 respectively, showing that the optimised artificial neural network model presented in this paper shows significantly better discrimination on the North Middlesex University Hospital ICNARC data set. Using Analysis of Variance (ANOVA) techniques to test the significance of the Area under the ROC curve results from 10-fold cross-validation (as recommended by Bradley [3]) gives us a p-value of 0.0001, indicating that the optimised ANN classification approach significantly outperforms the existing ICNARC model in term of the AUC metric.

Table III shows true positive rate and false positive rate for both classification models at 10% decision criteria between 0 and 100%. It can be seen that the optimised ANN model significantly outperforms the ICNARC model in terms of false positive rate at all decision criteria. The ANN model also performs better with respect to the true positive rate at decision criteria from 40% upwards. Whilst the ICNARC model has a better true positive rate at decision criteria below 40%, this comes at the cost of significantly higher false positive rates.

TABLE III. TRUE POSITIVE AND FALSE POSITIVE RATES FOR BOTH THE ICNARC MODEL AND THE OPTIMISED ANN MODEL AT 10% DECISION CRITERIA

	True positive rate (%)		False positive rate (%)	
Decision criterion (%)	ICNARC	ANN	ICNARC	ANN
10	100.0	89.2	94.0	19.2
20	96.2	82.9	68.5	10.7
30	85.6	78.4	43.0	6.9
40	62.5	72.1	20.6	4.3
50	44.2	65.8	8.0	3.0
60	29.8	58.6	4.4	1.9
70	17.3	52.3	2.4	1.3
80	9.6	46.9	0.6	0.7
90	4.8	38.7	0.2	0.1

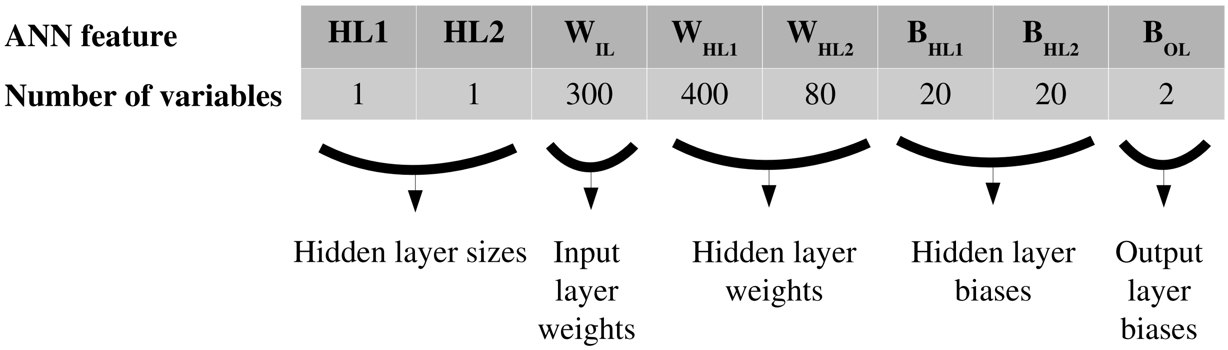
Table IV shows the total classification accuracy of both models, again at 10% decision criteria between 0 and 100%. This shows that the optimised ANN model outperforms the ICNARC model in terms of classification accuracy at all decision criteria and, by using decision criteria between 30% and 80%, the ANN model is capable of achieving an overall classification accuracy of over 90%.

TABLE IV. OVERALL CLASSIFICATION ACCURACY FOR BOTH THE ICNARC MODEL AND THE OPTIMISED ANN MODEL AT 10% DECISION CRITERIA

	Total correct classification rate (%)	
Decision criterion (%)	ICNARC	ANN
10	23.6	82.2
20	43.6	88.2
30	62.3	90.5
40	76.2	91.6
50	83.1	91.6
60	83.2	91.3
70	82.5	90.7
80	82.5	90.2
90	82.0	89.3

Figure 4 shows a performance comparison between the

[1] Atavism is a tendency for evolutionary traits to lie dormant but remain intact.



ANN feature	HL1	HL2	W_{IL}	W_{HL1}	W_{HL2}	B_{HL1}	B_{HL2}	B_{OL}
Number of variables	1	1	300	400	80	20	20	2

Fig. 2. Encoded chromosome for the ANN consisting of 2 hidden layers (HL), an input layer (IL), 2 neurons on the output layer (OL), and associated biases, totalling 784 variables

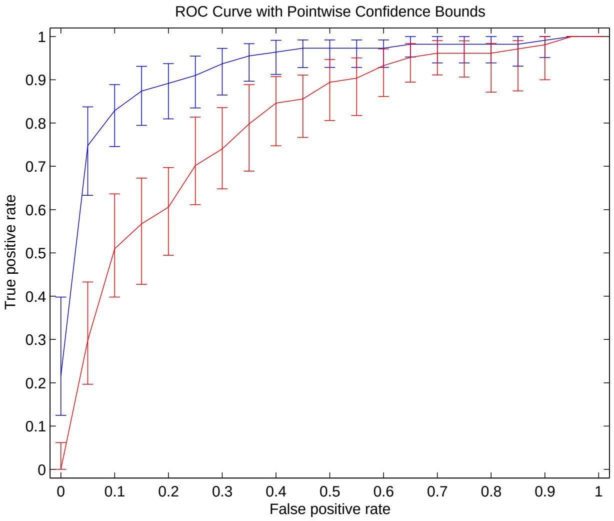


Fig. 3. Receiver-operating characteristic curves for the optimised ANN classifier (in blue) and the ICNARC model (in red)

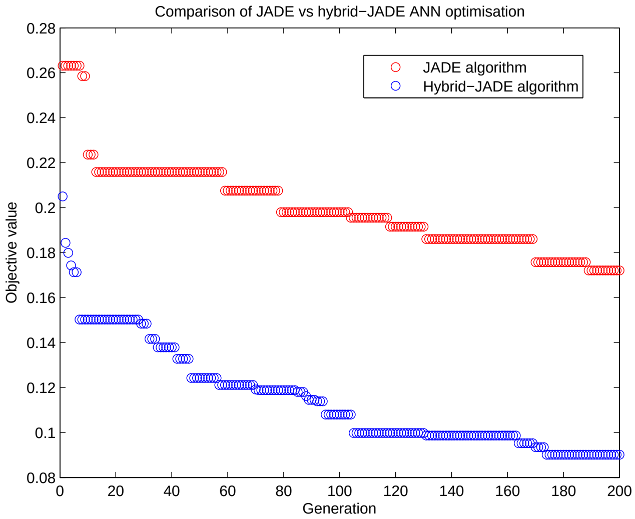


Fig. 4. Performance comparison of the JADE algorithm and hybrid JADE algorithm applied to the design of an ANN classifier for mortality prediction

JADE algorithm with no local search and the hybrid JADE algorithm proposed in this paper. Both algorithms were applied to the classifier design problem discussed in this section and it can be seen that the hybrid JADE algorithm not only exhibits much faster initial convergence, but also achieves significantly better classifier performance after 200 generations.

V. Conclusions and Further Work

In this paper, a novel hybrid adaptive differential evolution algorithm has been introduced that combines the resilient back-propagation local search heuristic and an adaptive differential evolution algorithm, JADE. This algorithm is capable of efficiently optimising artificial neural network parameters by utilising global search methods to avoid getting stuck in local optima, whilst still leveraging the efficiencies of gradient descent based back-propagation.

The hybrid JADE algorithm proposed in this paper has been applied to the design of a classifier for the real-world problem of predicting mortality rates in critical care admissions to North Middlesex University Hospital. Results presented in this paper have shown that the predictive capacity of the classifier designed using the hybrid JADE algorithm significantly outperforms the current state-of-the-art in mortality risk prediction for critical care environments. In particular, the classifier designed using the hybrid JADE algorithm exhibits robust classification accuracy across a wide range of decision criteria - with overall classification accuracy of greater than 90% at decision criteria between 30 and 80%.

Further work is planned on extending this work to larger datasets gathered across multiple hospital sites. The results currently show that designing and training a classifier on the North Middlesex University Hospital critical care admissions dataset using the hybrid JADE algorithm can outperform the benchmark ICNARC model currently used by the NHS for auditing critical care units in the UK. However, the ICNARC model has been developed using a national database of critical care admissions, so it is possible that the geographical diversity of the data used in developing the ICNARC model makes it more generally applicable for critical care admissions across the UK.

More further work is also planned on evaluating the proposed hybrid JADE algorithm on a wider variety of machine learning benchmark datasets (such as the Cleveland Heart Disease dataset, Pima Indians Diabetes dataset, and the Wisconsin Breast Cancer dataset).

References

[1] M. Attik, L. Bougrain, and F. Alexandre. Neural network topology optimization. In *ICANN 2005: the 15th International Conference on Artificial Neural Networks*, pages 53–58. Springer, 2005.

[2] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[3] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[4] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation*, 15(1):4–31, 2011.

[5] S. Dreiseitl and L. Ohno-Machado. Logistic regression and artificial neural network classification models: a methodology review. *352-359*, 35(5-6), 2002.

[6] R. Gämperle, S. D. Müller, and P. Koumoutsakos. A parameter study for differential evolution. *Advances in intelligent systems, fuzzy systems, evolutionary computation*, 10:293–298, 2002.

[7] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA., 1989.

[8] M. Gori and A. Tesi. On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):76–86, 1992.

[9] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. D. Jesús. *Neural network design*. PWS Publishing, Boston, 1996.

[10] K. Hajian-Tilaki. Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. *Caspian Journal of Internal Medicine*, 4(2):627–635, 2013.

[11] D. A. Harrison, G. J. Parry, J. R. Carpenter, A. Short, and K. Rowan. A new risk prediction model for critical care: the intensive care national audit & research centre (icnarc) model. *Critical care medicine*, 35(4):1091–1098, 2007.

[12] T. Iidaka. Resting state functional magnetic resonance imaging and neural network classified autism and control. *Cortex*, 63(0):1047–1057, 2015.

[13] H. Ishigami, T. Fukuda, T. Shibata, and F. Arai. Structure optimization of fuzzy neural network by genetic algorithm. *Fuzzy Sets and Systems*, 71(3):257–264, 1995.

[14] W. A. Knaus, E. A. Draper, D. P. Wagner, and J. E. Zimmerman. Apache ii: a severity of disease classification system. *Critical care medicine*, 13(10):818–829, 1985.

[15] W. A. Knaus, J. E. Zimmerman, D. P. Wagner, E. A. Draper, and D. E. Lawrence. Apache-acute physiology and chronic health evaluation: a physiologically based classification system. *Critical care medicine*, 9(8):591–597, 1981.

[16] J.-R. Le Gall, S. Lemeshow, and F. Saulnier. A new simplified acute physiology score (saps ii) based on a european/north american multicenter study. *Jama*, 270(24):2957–2963, 1993.

[17] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[18] E. Mezura-Montes, J. Velázquez-Reyes, and C. C. Coello. Modified differential evolution for constrained optimization. In *IEEE Congress on Evolutionary Computation, 2006*, pages 25–32. IEEE, 2006.

[19] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello. A comparative study of differential evolution variants for global optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 485–492. ACM, 2006.

[20] Z. Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer, Berlin, 2000.

[21] P. Moscato and C. Cotta. A gentle introduction to memetic algorithms. In *Handbook of metaheuristics*, pages 105–144. Springer, 2003.

[22] F. Neri and C. Cotta. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2:1–14, 2012.

[23] F. Neri and V. Tirronen. Scale factor local search in differential evolution. *Memetic Computing*, 1(2):153–171, 2009.

[24] F. Neri and V. Tirronen. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*, 33(1-2):61–106, 2010.

[25] N. Noman and H. Iba. Accelerating differential evolution using an adaptive local search. *IEEE Transactions on Evolutionary Computation*, 12(1):107–125, 2008.

[26] K. Price, R. M. Storn, and J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer, 2005.

[27] R. Reed. Pruning algorithms-a survey. *IEEE transactions on Neural Networks*, 4(5):740–747, 1993.

[28] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *IEEE International Conference on Neural Networks, 1993*, pages 586–591. IEEE, 1993.

[29] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, 1958.

[30] S. Saarinen, R. Bramley, and G. Cybenko. Ill-conditioning in neural network training problems. *SIAM Journal on Scientific Computing*, 14(3):693–714, 1993.

[31] R. S. Sexton, R. E. Dorsey, and J. D. Johnson. Toward global optimization of neural networks: a comparison of the genetic algorithm and backpropagation. *Decision Support Systems*, 22(2):171–185, 1998.

[32] A. Shenfield, P. J. Fleming, V. Kadirkamanathan, and J. Allan. Optimisation of maintenance scheduling strategies on the grid. *Annals of Operations Research*, 180(1):213–231, 2010.

[33] K. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

[34] R. Storn. Differential evolution design of an iir-filter. In *Proceedings of IEEE International Conference on Evolutionary Computation, 1996*, pages 268–273. IEEE, 1996.

[35] 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.

[36] J. V. Tu. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, 49(11):1225–1231, 1996.

[37] L. Wong and J. Young. A comparison of icu mortality prediction using the apache ii scoring system and artificial neural networks. *Anaesthesia*, 54(11):1048–1054, 1999.

[38] X. Yao and Y. Liu. Making use of population information in evolutionary artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 28(3):417–425, 1998.

[39] G. P. Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 30(4):451–462, 2000.

[40] J. Zhang and A. C. Sanderson. Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, 13(5):945–958, 2009.