Applied Intelligence (2023) 53:5732–5749
https://doi.org/10.1007/s10489-022-03539-8

Neural network input feature selection using structured l_2- norm penalization

Nathaniel Egwu[1] · Thomas Mrziglod[2] · Andreas Schuppert[1]

Accepted: 22 March 2022 / Published online: 2 July 2022
© The Author(s) 2022

Abstract
Artificial neural networks are referred to as universal *approximators* due to their inherent ability to reconstruct complex linear and nonlinear output maps conceived as input-output relationships from data sets. This can be done by reducing large networks via regularization in order to establish compact models containing fewer parameters aimed at describing vital dependencies in data sets. In situations where the data sets contain non-informative input features, devising a continuous, optimal input feature selection technique can lead to improved prediction or classification. We propose a continuous input selection technique through a dimensional reduction mechanism using a 'structured' l_2- norm regularization. The implementation is done by identifying the most informative feature subsets from a given data set via an adaptive training mechanism. The adaptation involves introducing a novel, modified gradient approach during training to deal with the *non-differentiability* associated with the gradient of the structured norm penalty. When the method is applied to process data sets, results indicate that the most informative inputs of artificial neural networks can be selected using a structured l_2- norm penalization.

Keywords Input optimization · Sparsity · Regularization · Structured penalty · Feature selection · Deep learning

1 Introduction

An integral part of learning with artificial neural networks (ANNs) involves selecting the most relevant features from data sets [1–4]. The aim is to reduce the complexity and computational constraints associated with the resulting models in order to improve their ability to *generalize* on new data sets. This can be done by adapting classical *learning* algorithms, where intricate relationships between data features and labels conceived as internal mappings are identified via training. Artificial neural network (ANN) training is done by statistical inference via regression using *back-propagation*, where modelling involves parameter (weights in the case of ANNs) optimization – corresponding to the minimum error of an objective function [5–8]. For instance, when a learning algorithm is categorized as supervised based on data labels, optimal feature selection from data sets can improve the quality of output map reconstruction using well-known mathematical results [9–11].

In principle, feature selection techniques can be categorized as wrapper, filter and embedded methods [12–14]. While wrapper methods involve coupling a feature subset to a classifier in order to identify relevant data features [15–17], filter methods involve identifying *relevance* among feature subgroups using a statistical information measure. Also, filter methods do not involve data labels but transform the original feature space along a lower dimensional manifold. Meanwhile, wrapper methods are effective in obtaining feature subsets but inefficient and unsuitable for large-scale or high-dimensional data sets [18]. This is due to the fact that in wrapper methods, the performance of the selected feature subset is often *validated* on the training outcome. Hence, wrapper methods are suitable for feature selection in supervised learning scenarios. In [19], a hybrid formulation consisting of both the wrapper and filter method based on applying a multi-objective (whale) optimization was proposed. The aim is to merge the merits of both methods into a single system in order to achieve better performance. However, employing wrapper methods can

✉ Nathaniel Egwu
nathaniel.egwu@rwth-aachen.de

[1] Joint Research Center for Computational Biomedicine II, RWTH Aachen University, Pauwelsstraße 19, 52074 Aachen, Germany

[2] Engineering & Technology, Applied Mathematics, Bayer AG, 51368 Leverkusen, Germany

increase the computational time in problems with large feature space. Examples of filter selection methods include: F-statistic, reliefF [20, 21], minimal-redundancy-maximal-relevance criterion (mRMR) [22], t-test and information gain [23]. Both wrapper and filter methods are prone to over-fitting and can lead to extensive use of computational resources. Moreover, relying on statistical relevance for feature selection can introduce unwanted constraints on the training algorithm making it difficult for the resulting models to be properly interpreted. To cater for the aforementioned deficiencies, several embedded methods such as regularization [24, 25], dropout [26] methods have been proposed. In principle, embedded methods involve incorporating a mechanism into a supervised learning algorithm in order to identify the most relevant inputs and other features during training [26, 27]. However, these methods are limited in their ability to optimally select relevant (most informative) inputs and other features in problems with high dimensional data containing fewer data set points.

From several embedded feature selection methods, we focus on situations where a mechanism can be introduced during training – to analyze the *influence* of parameter groups associated with data inputs – in order to reduce the dimension of the input space. Examples of classical dimension reduction (DR) techniques include: the dropout method [26, 28], pruning [29, 30], feature extraction techniques such as the principal component analysis (PCA) [31, 32], or low dimensional approximation methods for feature selection in image classification problems [33] (where ANNs were used in DR of dynamical systems and applied to conductance-based neuron models [34]. In contrast to the aforementioned DR techniques, we focus on embedded feature selection methods which are based on regularization, where a penalty term is added to an objective function prior to ANN training.

In supervised learning literature, a well-developed theory on the application of regularization techniques in feature selection problems exists. For instance, in [25, 35–38] (and references therein), several regularization methods were proposed and successfully applied to many feature selection problems. Examples of regularization functions include the l_1 penalty (Lasso) – defined as the total sum of the absolute values of model parameters (or weights in ANNs) [39–42]. When ANNs are penalized with the Lasso, sparsity (zero parameters) is introduced in models. Other penalties include the ridge regression (l_2 norm squared) and the elastic net - obtained by combining the l_1 (Lasso) and the squared l_2 norm. Modified forms of the Lasso and refined l_2 penalties were proposed in [33] to eliminate unimportant filters and channels in convolutional neural networks. Meanwhile, [25] proposed the use of exclusive group Lasso for intra-group learning based on the extensive grouping of network parameters and validated on several benchmark problems. In [37], a tree-guided group Lasso was proposed to relate multiple genes in gene expression data by leveraging hierarchical clusters. The Lasso is known for inducing zeros in model parameters due to its geometric property, while the squared l_2 norm tends to *drive* parameters to small values. Several combinations of the aforementioned penalty functions have successfully been applied to many problems [24, 35, 43, 44]. Meanwhile, [24] proposed a penalty for the simultaneous selection of input and hidden layer neurons of deep neural networks in classification problems.

Furthermore, the embedded feature selection method proposed in [45] relies on an input-to-output skip-layer (residual) connection that allows a feature to have non-zero weights in a hidden unit only if its skip-layer connection is active. While [1] proposed a grouped lasso regularization which is an integrated scheme for feature selection along with controlling the extent of redundancy in the selected features using neural networks. In the latter case, the method also handles problems with redundant features. While in [46], an l_1 regularization for improving the performance of imbalanced classification, and an iterative scheme to solve the optimization problem associated with feature selection was proposed. Meanwhile, [47] proposed a three step method via a hypergraph based multi-task feature selection for multi-modal classification, which seeks to address limitations associated with including redundancy among features. While in [48], a two-layer feature selection approach that combines a wrapper and an embedded method in constructing an appropriate subset of predictors was proposed. Essentially, the wrapper method was adopted to search for optimal subset of predictors, while the elastic net was used to eliminate remaining redundant predictors in order to improve prediction accuracy. Meanwhile, in [49], a top$-k$ feature selection framework was proposed to utilize an $l_{0,2}$ regularization norm as a sparsity matrix constraint. The method was applied to unsupervised, semi-supervised and supervised learning scenarios. Besides, these embedded regularization techniques (including the Lasso and its flavours and elastic net) do not tend to recognize grouping in high dimensional data sets with implicit nonlinear relationships among input features.

However, the aforementioned embedded feature selection techniques are suited to problems with large data set points which are not obtainable in biomedical, pharmaceutical and crop industries. In such circumstances, the lack of *scalability* in the data sets corresponding to quality measurements involving batch runs during production do not often varry per year. Moreover, since training deep ANNs are often characterized by large data set points which are not available in these industries. Hence, applying the aforementioned feature selection techniques will not be suitable in these scenarios. Besides, relying on two-step modelling techniques where the feature selection step is separated from the classification or prediction step, can add unwanted computational problems during training. Other key constraints associated with classical approaches include under-fitting, over-fitting and reliance on validating performance outcomes by sub-diving the Training data set to determine *best* informative inputs and other relevant features. To cater for these deficiencies, we propose a continuous feature selection technique where a novel penalty function is incorporated during training to *control* the influence of non-informative input features. The proposed mechanism combines both the informative input selection, optimal parameters and neurons together with the classification or prediction steps during training. The aim is to establish *robust* models for *better* generalization and performance outcomes on new data sets. In this paper, model robustness refers to the fact that the *extrapolative* ability of the resulting model (represented by the most informative inputs and structural information or the least number of optimally selected parameters and neurons) is resistant to changes in the data set points. In principle, the goal is to reduce an *over-parameterized* ANN parameter and input feature space to get a feature subspace, which is then used to establish a simplified, partial or *sub*-network model. The mechanism is inspired by the notion of *parsimonious* models in statistical learning literature [41, 50] – where sparse models (containing few parameters) are often preferred to complex models (containing more parameters). This assumption is based on the fact that the most vital information contained in data sets can be described by models containing fewer inputs and other relevant features (parameters in the case of ANNs). The aim of the embedded structured penalty is to adapt the classical training problem to cater for constraints associated with the non-informative inputs and other relevant features contained in the original data set.

In principle, incorporating the proposed structured l_2 penalty function – defined as the cumulative sum of l_2 norms – to an objective (loss) function transforms the initial ill-posed objective function (containing non-informative inputs) to a well-posed problem. The theoretical background associated with the structured l_2 penalty is to implicitly introduce a *grouping* structure among the parameter set corresponding to each ANN input neuron. The formalism is aimed at determining the importance of each input feature using the influence of each l_2 norm on each parameter group emanating from each input neuron. This is due to the geometric property of the l_2 norm function, which is used to simultaneously *shrink* the l_2 norm of each group of parameters corresponding to each input. Hence, the most informative inputs can be simultaneously identified and retained, while non-informative inputs are eliminated during training. However, the newly introduced structured l_2 norm penalty renders the initial objective function *non-differentiable*. To deal with this, we adapt the classical training algorithm by introducing a modified gradient condition in order to guarantee convergence of the adapted training algorithm. Thus, the proposed formalism preserves the most significant information contained in the data set by selecting the most informative input and the least number of optimal parameters during training. The method also eliminates the need for sub-dividing the data set for validation during training, prevents over-fitting and provides high accuracy when applied in classifying or predicting data outputs. The main contributions can be highlighted as follows:

- Propose a novel embedded feature selection mechanism by incorporating a structured penalty during ANN training aimed at selecting the most informative inputs from data sets. The method will be validated on small to medium sized data set points in contrast to existing methods which are characterized by large data set points.
- The mechanism induces a grouping structure among inputs and relies on the implicit relationship between inputs and labels for optimal input and feature selection in contrast to using statistical importance, ranking or relevance methods which can lead to loss of vital information in data sets. It also optimizes the number of neurons in each ANN layer during training, enabling us to learn less complexity in data sets and thus provide a good understanding of the problem.
- The method differs from the aforementioned embedded regularization techniques because of its ability to simultaneously identify and eliminate distinct parameter groups corresponding to non-informative inputs. Also, it leads to models whose solution space is represented by the least number of optimal parameters thereby simplifying model evaluation in applications.
- The method also relies on fine-tuning parameters by reducing the model bias and over-parameterized ANNs via sparsity to determine optimal input and feature subsets with the best model performance. The aim is to remove unwanted *noise* from the data during training and in applications using robust models which offer flexibility in generalization and extrapolation (shown in the Result section).
- The technique is applicable in industrial settings where process quality measurements are not *scalable* due to the invariability of the dimension of the data set points and the high cost associated with generating such data. Hence, optimal inputs and relevant feature selection is needed to prevent the loss of vital information in such applications.

In order to validate the performance of the proposed mechanism, training will be performed on different data sets of varrying compositions. Next, essential parameters will be introduced to *fine-tune* models prior to application. In order to quantify the model prediction and classification accuracy, several performance statistics will be obtained by analyzing the resulting model residues and complexity computed from optimal model parameters. These quantities (model residues and complexity) will be obtained from an ensemble of models and used for model selection. The remaining outline of this paper is organized as follows: In Section 2, we introduce the notations and present the problem statement, while the implementation and optimization methods are discussed in Section 3. The input selection algorithm is presented in Section 4, while the experimental setup and the results obtained by applying the formalism on artificial and real data sets are presented in Section 5. Meanwhile, the conclusion and discussion are presented in Section 6.

2 Problem statement

Consider the process data set $\mathcal{D} = (\mathbf{x}_i, \mathbf{y}_i) \in \mathbf{R}^{n+1}$, where $\mathbf{x}_i$ are inputs and $\mathbf{y}_i$ are outputs. We assume that the outputs $\mathbf{y}_i$ consist of a noise vector ε and the *true* output $\mathbf{y}_i$; where both $\mathbf{y}_i$ and $\mathbf{y}_{i,true}$ are related by:

$$\mathbf{y}_i = \mathbf{y}_{i,true} + \varepsilon, \quad \mathbf{y}_{i,true} \in \mathbf{R}. \tag{1}$$

For each component in $\mathcal{D}$ containing N data points, we have that $\mathbf{x}_i \in \mathbf{R}^{N\times n}, \mathbf{y}_i \in \mathbf{R}^N$; while both components in $\mathcal{D}$ will be represented by $\mathbf{x}_i$ and $\mathbf{y}_i$ respectively. Note that $\varepsilon \sim \mathcal{N}(\mu, \sigma^2)-$ implies that the noise vector ε, is randomly drawn from the normal distribution with mean μ and standard deviation σ. Let the matrix $\Theta \subset \mathbf{R}^{n\times m}$ represent the parameters (weights) connecting all layers of a feed-forward ANN (Fig. 1) structure containing n inputs, m hidden and 1 output layer neurons. If the modelling objective is to approximate $\mathbf{y}_i$, the optimal parameter matrix corresponding to the resulting model will be represented by $\Theta^* \subset \mathbf{R}^{n\times m}$. In non-linear regression problems, modelling involves optimization of an objective function $F(\Theta)$ via the well-known residual sum of squares formula:

$$F(\Theta) = \sum_{i=1}^{N} \left(\mathbf{y}_i - f(\mathbf{x}_i, \Theta)\right)^2. \tag{2}$$

Meanwhile, for binary classification problems [51–53], the loss function is defined as:

$$G(\Theta) = -\frac{1}{M}\sum_{l=1}^{M}\left(\mathbf{y}_l \log(f(\mathbf{x}_l, \Theta)) + (1-\mathbf{y}_l)\log(1-f(\mathbf{x}_l, \Theta))\right), \tag{3}$$

where $f(\mathbf{x}_i, \Theta)$ can either represent a single hidden layer ANN structure:

$$f(\mathbf{x}_i, \Theta) = \sum_{j=1}^{m} \theta_j \sigma \left[\sum_{l=1}^{n} x_{ij}\theta_{jl}\right], \quad m_1, m_2, n_1 \in \mathbf{N}; \tag{4}$$

or a 2 hidden layer deep ANN structure:

$$g(\mathbf{x}_i, \tilde{\Theta}) = \sum_{k=1}^{m_1} \theta_k \sigma \left[\sum_{j=1}^{m_2} \theta_{jk} \sigma \left[\sum_{l=1}^{n_1} x_{ij}\theta_{jl}\right]\right], \quad m_1, m_2, n_1 \in \mathbf{N}; \tag{5}$$

and σ is the activation function. In compact form, (2) can be written as

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}}\, F(\Theta). \tag{6}$$

In order to compute Θ^*, we minimize the loss function $F(\Theta)$, by training a *suitable* ANN structure using a gradient-based algorithm [54]. The aim of the input selection problem is to find $\mathbf{x}_i^* \subset \mathbf{x}_i \in D$ for some $\varepsilon > 0$ where $\Theta^* \subset \mathbf{R}$ and:

$$\mathbf{x}_i = \{\mathbf{x}_i^F, \mathbf{x}_i^{NF}\}, \tag{7}$$

where $\mathbf{x}_i^F, \mathbf{x}_i^{NF}$ represent the informative (F) and non-informative inputs respectively. In order to accomplish this, we adopt the following grouping mechanism for a 1-hidden layered ANN structure: let $\Theta_1 \subset \Theta$ represent the set of parameters connecting the hidden to output layer neurons; while $\Theta_2 \subset \Theta$ represent the set of parameters connecting input to the first hidden layer neurons (with biases at each layer incorporated into the corresponding group). The grouping mechanism permits the use of $\Theta = \{\Theta_1, \Theta_2\}$ to represent the total parameters in each structure in Fig. 1, while $\tilde{\Theta} = \{\tilde{\Theta}_1, \tilde{\Theta}_2\}$ can be used to represent the total parameters in a 2- hidden layered deep ANN. For input selection, we propose that the structured l_2 norm penalty:

$$P_2(\lambda_2, \Theta_2) = \lambda_2 \left[\sum_{k=1}^{n} || \Theta_2^k ||_2\right], \tag{8}$$

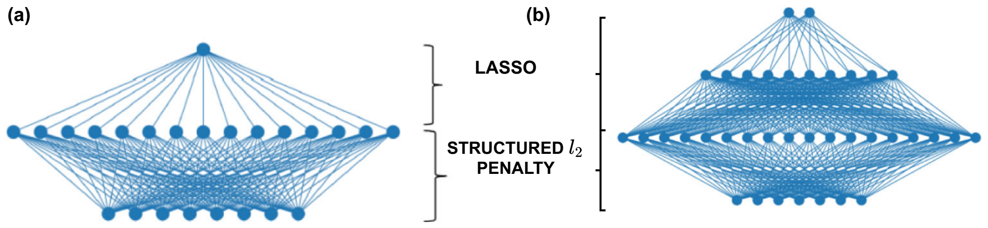

Fig. 1 A schematic diagram showing how the structured penalty and Lasso can be incorporated to different layers of a (a) shallow and (b) deep ANN structures

should be incorporated in (1). Note that the structured penalty function in (8) is defined as the cumulative sum of l_2 norms: $|| \Theta_2^l ||_2$, where each l_2 norm *penalizes* each set of parameters $\Theta_2^l \subset \Theta_2, \quad l = 1, \ldots, n$; – emanating from each input neuron. The structured penalty in (8) can be explained as follows:

$$P_2(\lambda_2, \Theta_2) = \lambda_2 \left[l_2(\Theta_2^1) + l_2(\Theta_2^2) + l_2(\Theta_2^3) \ldots + l_2(\Theta_2^n) \right] \tag{9}$$

where $l_2(\Theta_2^1)$ means that the l_2- norm penalizes the group of parameters corresponding to Θ_2^1. For a process containing n inputs, Θ_2 becomes:

$$\Theta_2 = \{\Theta_2^1, \Theta_2^2, \Theta_2^3, \ldots, \Theta_2^n\} \tag{10}$$

To optimize the ANN parameters in all layers, (8) is combined with the Lasso penalty [40]. Incorporating (8) and the Lasso to (6), we obtain the penalized objective function (POF) for a single hidden layer ANN:

$$\Theta^* = \arg \min_{\Theta_1, \Theta_2 \subset \Theta} \left[F(\Theta) + P_1(\lambda_1, \Theta_1) + P_2(\lambda_2, \Theta_2) \right], \tag{11}$$

where $\lambda_1, \lambda_2 \in \mathbf{R}$ are called the tuning, regularization or trade-off parameters. For deep-layered ANNs, Θ in (11) can be replaced by $\tilde{\Theta}$; while for classification problems, $F(\Theta)$ can be replaced by $G(\Theta)$. The choice of (8) for feature selection is influenced by the geometric property of the l_2- norm – which tends to *shrink* the total norm value of the associated parameter vector by the same amount thereby, controlling the influence of the *gradient* of the objective function and parameter values during optimization. The implementation is done by adapting the gradient-based algorithm [54], by incorporating the continuous input feature selection. Deductions will also made from the structural information associated with the resulting models and used for model selection.

3 Implementation and optimization methods

3.1 Implementation method

In Fig. 1, we present how both the structured l_2 penalty and Lasso can be incorporated to different layers of a (a) shallow and (b) deep ANN structure without overlaps (meaning both penalty functions are incorporated at different ANN layers). Let $\{\Theta_1 \cup \Theta_2\}$ represent the total parameters in each ANN structure. If Θ_2 represent the total parameters connecting the input to hidden layer neurons, another subgroup of $\Theta_2^l \subset \Theta_2 \quad l = 1, \ldots, n$ can be identified as shown in 10. The number of values in each $\Theta_2^l \quad l = 1, 2, \ldots, n$ is determined by the number of neurons in the first hidden layer. Thus, l_2 norms are incorporated in each input neuron containing Θ_2^l number of weight and the cumulative sum is added to (2) to get (11).

In order to train the POF, we devise a criterion to compare the *contribution* and influence of each Θ_2^l to the loss function and the structured l_2 norm penalty. When the contribution of each group of Θ_2^l to the structured penalty is less than its contribution to the loss function, the entire sub-group of Θ_2^l is set to zero and the corresponding neurons dropped from the ANN structure during the next iterative steps. This results in a reduction of the dimension of input feature space and the most informative inputs and features being retained during training. When this happens, sparse ANN model structures ensue. To understand the influence of the structured penalty (when combined with the Lasso), reference can be made to earlier results which are based on the interaction of different penalty functions in training ANNs [41, 42, 55]. Similarly, optimizing (8) results in two types of sparsity: the first case arises due to the Lasso, while the second case arises due to the elimination of non-informative inputs and features. Subsequently, a mechanism will be proposed to handle the non-differentiability associated with the gradient of both penalty terms.

3.2 The overall training steps

The Flowchart displayed in Fig. 2 represents an overview of the main implementation steps of the feature selection mechanism. The initial step consists of the data preprocessing, normalization and initialization of the regularization (λ_1 and λ_2) and training parameters. Next, a conditional counter variable *(COND_COUNTER)* > 0 and a *check variable* $K > 0$ are introducing prior to training. The value of K is used to determine how many times the condition in (12) can be computed (since (12) is calculated whenever *COND_COUNTER*$= K$) thereby preventing a significant increase in the overall training time. In our experiment, we chose $K = 5, 10, 15, \ldots, N_k$, where $N_k \leq$ *epochs* (see the detail description in Fig. 3). Note that when (12) is satisfied for any group of input parameter set, the entire parameters are fixed (thereby preventing non-differentiability of the structured penalty). As shown in Fig. 2, setting the parameter group to zero depends on comparing the value obtained from (12) with the threshold parameter - *THRESHOLD_PARAM*. Training stops when $K \geq$ *epochs*, since K is bounded by the total number of training *epochs*. A detailed description of the implementation steps (including a Pseudo code and Flowchart) of the proposed mechanism is presented in Section 4.

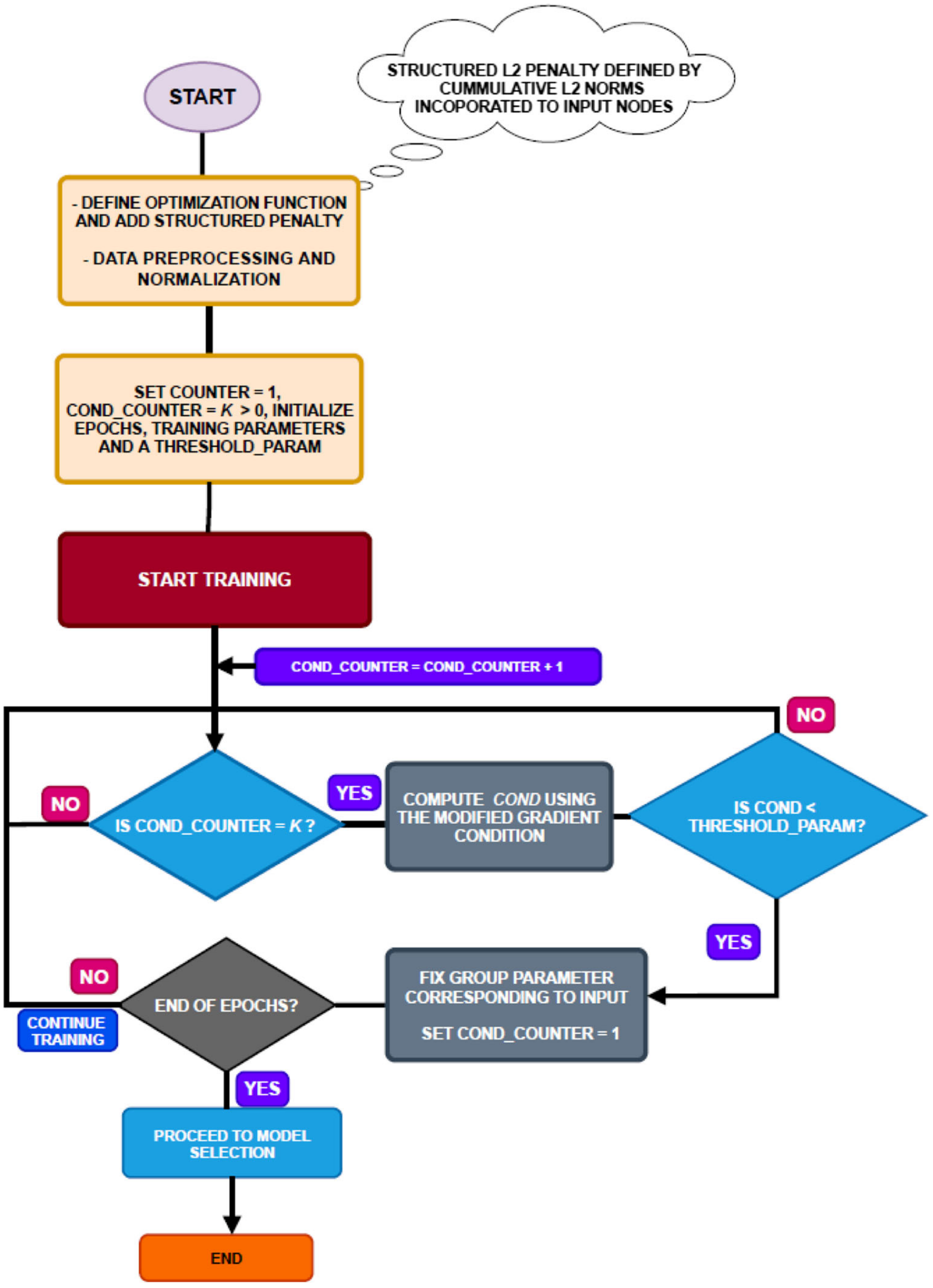
STRUCTURED L2 PENALTY DEFINED BY CUMMULATIVE L2 NORMS INCOPORATED TO INPUT NODES
START
- DEFINE OPTIMIZATION FUNCTION AND ADD STRUCTURED PENALTY
- DATA PREPROCESSING AND NORMALIZATION
SET COUNTER = 1, COND_COUNTER = K > 0, INITIALIZE EPOCHS, TRAINING PARAMETERS AND A THRESHOLD_PARAM
START TRAINING
COND_COUNTER = COND_COUNTER + 1
NO
NO
IS COND_COUNTER = K ?
YES
COMPUTE COND USING THE MODIFIED GRADIENT CONDITION
IS COND < THRESHOLD_PARAM?
YES
NO
CONTINUE TRAINING
END OF EPOCHS?
FIX GROUP PARAMETER CORRESPONDING TO INPUT
SET COND_COUNTER = 1
YES
PROCEED TO MODEL SELECTION
END

Fig. 2 A schematic view of the overall implementation steps. It presents the data pre-processing, normalization and initialization of the training parameters. The embedded feature selection mechanism is performed at $THRESHOLD\_PARAM$ number of times during training

3.3 Optimization method

In order to determine Θ^* in (11), an initial ANN structure is trained by minimizing $F(\Theta)$ using a gradient-based algorithm [54]. During training, two types of non-differentiable scenarios arise: the first case is due to the non-differentiability of the gradient of the structured l_2 penalty, while the second case arises due to gradient of the Lasso. To cater for the non-differentiability associated with the Lasso, we resort to the modified gradient approach in [40] and propose a new formalism to deal with the non-differentiability associated with the gradient of the structured l_2 norm penalty. In principle, this involves selecting an arbitrary number of iterative steps where a criterion is introduced to *test* the closeness of the objective function to zero, when a *search* direction is accepted during

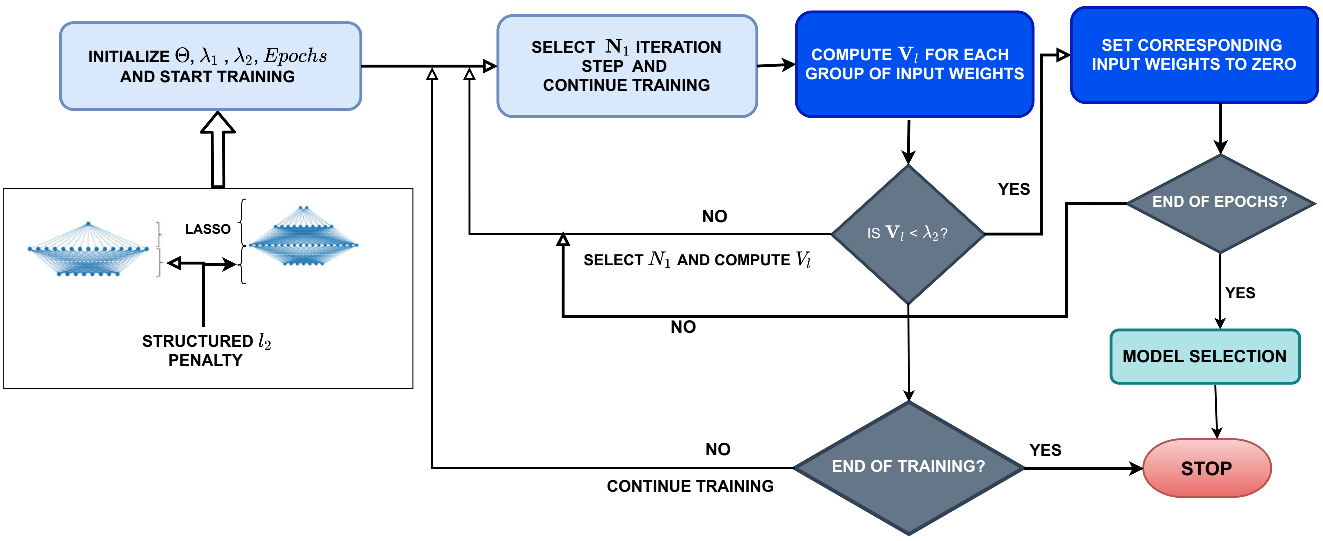
INITIALIZE Θ, λ_1, λ_2, $Epochs$ AND START TRAINING
SELECT N_1 ITERATION STEP AND CONTINUE TRAINING
COMPUTE V_l FOR EACH GROUP OF INPUT WEIGHTS
SET CORRESPONDING INPUT WEIGHTS TO ZERO
LASSO
STRUCTURED l_2 PENALTY
NO
IS $V_l < \lambda_2$?
YES
SELECT N_1 AND COMPUTE V_l
END OF EPOCHS?
YES
NO
MODEL SELECTION
NO
END OF TRAINING?
YES
STOP
CONTINUE TRAINING

Fig. 3 An overview of the input feature selection (optimization) steps

training. To understand this formalism, let the gradient of the loss function with respect to the group parameters Θ_2^k be given by:

$$\frac{\partial F(x_i,\Theta)}{\partial \Theta_2^k} = 2\left[\sum_{i=1}^{N}(\mathbf{y}_i - f(\mathbf{x}_i,\Theta))\,\frac{\partial f(\mathbf{x}_i,\Theta)}{\partial \Theta_2^k}\right]; \qquad (12)$$

while the gradient of structured l_2 norm with respect to Θ_2^k be given by:

$$\frac{\partial P_2(\lambda_2,\Theta_2)}{\partial \Theta_2^k} = \lambda_2\left[\sum_{k=1}^{n}\frac{\partial \,||\,\Theta_2^k\,||_2}{\partial \Theta_2^k}\right] = \lambda_2\left\{\frac{\Theta_2^k}{||\,\Theta_2^k\,||_2}, k=1,\ldots,n\right\} \qquad (13)$$

Applying the l_2 norm to the gradient in (13), we get λ_2 since:

$$\left\|\lambda_2\left[\frac{\Theta_2^k}{||\,\Theta_2^k\,||_2}\right]\right\|_2 = \lambda_2; \left\|\left[\frac{\Theta_2^k}{||\,\Theta_2^k\,||_2}\right]\right\|_2 = 1. \qquad (14)$$

If each group of $\Theta_2^k = 0$ for some $k - th$ index, we compare their contributions to the loss function and structured l_2 norm penalty term using the modified gradient condition (MGC):

$$\left\|2\sum_{i=1}^{N}(\mathbf{y}_i - f(\mathbf{x}_i,\Theta))\,\frac{\partial f(\mathbf{x}_i,\Theta)}{\partial \Theta_2^k}\right\|_{l_2} < \lambda_2, \qquad (15)$$

The expression in (15) can be interpreted as follows: If the contribution of the l_2 norm of each group of Θ_2^k to the loss function is less than its contribution to the structured l_2 norm penalty, we set the corresponding Θ_2^k to zero. This is motivated by the gradient of the structured l_2 norm penalty evaluated at each parameter group in Θ_2^k *locally* in the neighborhood of zero. The geometric intuition associated with this formalism is based on the *jump* created by the l_2 norm gradient of the corresponding group of input parameters at zero. Thus, the l_2 norm value of each parameter sub-group is computed and compared with a threshold value. Due to the *shrinking* property of the l_2 norm, individual contribution of each input can be estimated. In doing so, non-informative inputs can be *zeroed* out, since the l_2 norm values corresponding to their parameter sub-groups will be close to zero thereby introducing sparsity among input features.

During implementation, several values of $\lambda_1, \lambda_2 \in \mathbf{R}$ were selected and their influence on the model observed during training. Note that the MGC was adapted to cater for the influence of λ_2 – which is used to *control* the amount of penalty – leading to the convergence of the training algorithm. By setting parameters which satisfy the MGC to zero, less informative input neurons can be disconnected from the ANN. For instance, choosing large values of λ_1 and λ_2 will introduced more zeros in Θ^*. Intuitively, the geometric interpretation can be understood by considering the relationship between the structured penalty and the loss function. Hence, setting non-informative or redundant input features to zero results in a reduction of the input feature dimension space during training. By setting the $k - th$ group of Θ_2^k to zero based on the MGC, convergence of the algorithm during training is guaranteed. Thus, if (15) is satisfied, λ_2 can be interpreted as a threshold value. The values displayed in Table 1 represent the results obtained when the method is applied to train a given number of benchmarks data sets.

4 The input selection algorithm

Let $\mathbf{V}_l$, $\quad l = 1,\ldots,n$, be used to denote the values computed from the left hand side of (15), and M, $\quad NIter \in$

$\mathbf{N}$; $M \leq NIter$ represent the total number of iteration points. During training, choose $N_j, j = 1, \ldots, N_M$ steps and compute $\mathbf{V}_l$ for each $\Theta_2^l \subset \Theta_2$, where $N_1 < N_2 < \ldots N_M \leq NIter$. For each $\mathbf{V}_l$, a check is performed to determine whether the modified gradient condition is satisfied and set $\Theta_2^l = \mathbf{0}$, where Θ_2^l corresponds to the $l - th$ parameter subgroup of Θ_2. When this happens, the entire parameter subgroup of $\Theta_2^l \subset \Theta_2$ will be *fixed* leading to a reduction in the input dimension. In doing so, we assume that Θ_2^l satisfies the MGC. Hence, when $\Theta_2^l = \mathbf{0}$, their corresponding neurons are also fixed during the next iteration steps, resulting in the non-informative inputs being removed from the ANN structure. An overview of the implementation steps involved in the algorithm is displayed in Fig. 3. The main steps can be summarized in the following *pseudo-code*:

```
– Pre-process data, initialize $\Theta_1, \Theta_2, n, m, \lambda_1, \lambda_2$ and set a counter $cnt = 0$.
– Select $N_1$ number of times and compute $\mathbf{V}_l$ resulting in $N_1$ vectors of $\mathbf{V}_l$, where $l \in \mathbf{N}$ corresponds to the total number of $n$ in each data set.
– Do
        Select $NIter$ iteration steps and and index subset $I \leq NIter$.
        For $N_1 < N_2 < \cdots < N_I$,   $N_I \leq NIter$,
        compute $\mathbf{V}_l$ from (15); $N_t \in \mathbf{R}, t = 1, 2, 3, \ldots, I$.
        Set group $\Theta_2^k = \mathbf{0}$ if (15) is satisfied, where $\Theta_2^k \subset \Theta_2$.
        Until end of training.
– For model selection:
        Varry each pair of $\lambda_1, \lambda_2$ and save the corresponding error (loss) values for each model. Compute the model complexity using non-zero parameter values in $\Theta^*$.
        Construct the Pareto boundary [56] and proceed with model selection.
```

5 Experimental setup and results

5.1 Experimental setup

In order to validated the method, 2 *prediction* and 4 *classification* data sets were trained. The prediction data consist of a Bayer Solar cell [40] and an Artificial data set, while the classification data consist of the Wisconsin Breast Cancer (WCDS) and Heart disease data sets (hosted on the UCI repository [57]). For the Prediction data sets, the modelling objective is to predict the output labels, while for the WCDS, the objective is to determine whether or not the diagnosis of participants (represented as features) can be classified into either malignant or benign cancer. While the aim of training the Heart disease data set is to determine whether or not participants have heart disease by classifying features into 2 classes. In particular, for a given data set, we let $n \leq M$; where M and n represent the data points and number of inputs respectively. Also, note that the method was validated for data sets where $n < 1000$ as presented in the Results section (see (5.4)) for which $M = \{126, 303, 100, 847\}$.

During each training phase, each value of λ_1 and λ_2 were varied 20 times, while the structural complexity and residual values corresponding to each model was computed. For each pair of (λ_1, λ_2), training was performed 10 times by randomly assigning different real values to Θ and the *global* minimum selected from the vector of 10 residual values. Initializing Θ during each training phase will provide a proper analysis regarding the relationship between each model error and complexity represented by the surface plots in Figures (Fig.) 4 (a)-(b). Also, experimenting with different initial Θ values can reduce the chance of outliers being introduced among Θ^*, thereby reducing inconsistency in the results. Further variation of Θ values can render the surface in Fig. 4(b) smoother thereby, facilitating the computation of the structural complexity corresponding to each model. To avoid repetition, we plot the residuals for 1 prediction data and use this to demonstrate the model selection in order to avoid multiple plots with same outcome for other data sets. The V_l values from (15) were computed at training intervals of 5, while features with 0 $V_l = 0$ were fixed during the next training steps.

5.2 Data sets

The composition of each data set containing n inputs and N set points is as follows:

– The prediction data sets containing $N = 100$ points are composed of:
  - ⋆ An artificial data composed of $n = 7 + 1$ (with 7 informative and 2 non-informative inputs),
  - ⋆ A Solar data set composed of $n = 6 + 1$ (with 6 informative and 1 non-informative inputs).

– The composition of the classification data sets (containing unknown number of informative and non-informative features prior to training) are defined as follows:
  - ⋆ A WCDS data set with $N = 899$ and $n = 32$; containing 1 output label which was encoded into 2 classes prior to training.
  - ⋆ A Heart disease data consisting of 2 files namely: the Heart disease 1 and the Heart

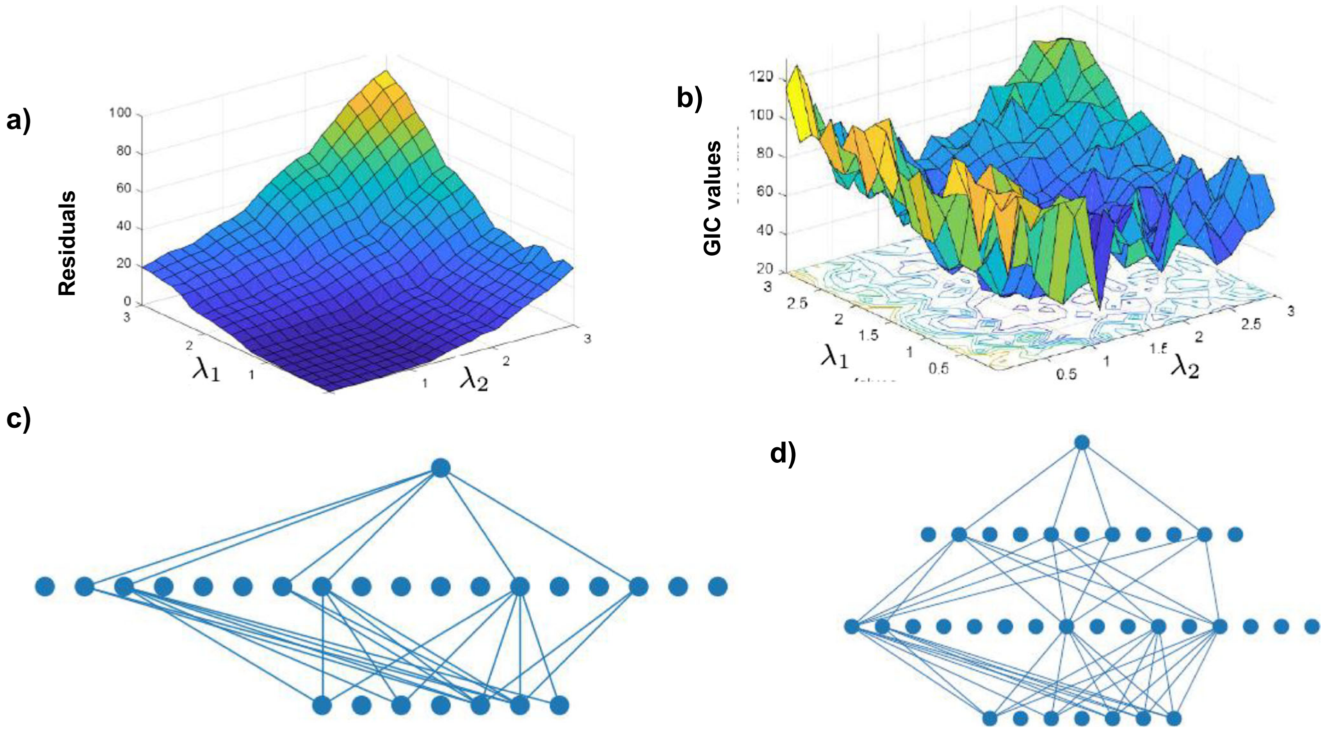

Fig. 4 Relationship between the (a) Residuals and (b) GIC values against λ_1, λ_2. Examples of ANN models with optimal topology and most informative inputs for the (c) Solar cell and (d) artificial data sets are also depicted. The non-informative inputs are disconnected from the ANN models in both cases

disease 2 data sets. Each contains $N = 303$ points with $n = 13$ and $n = 74$ inputs respectively.

- ⋆ The contents of the Heart disease 1 data file were drawn from the Heart disease 2 data file and is often used in research studies (see [57]). Using the information provided in [57], 4 input features in the Heart disease 1 data was encoded into 2 features.
- ⋆ In each case, the output labels were encoded into 2 classes prior to training.

– Classification data set for speech assessment (PD Data) containing $n = 300$, $N = 126$ [58] with output 2 classes.

5.3 Data preparation

Prior to training, the classification data sets were *pre-processed* using the *Sklearn* package [59] in *Python 3.7* [60]. Using the information provided by the data set Authors on the UCI Repository [57], 2 columns were dropped from the original data set. For the heart disease data set, *imputation* was used to handle cases of missing values. Then, each data set was normalized and divided in the ratio of 70 : 30 for Training and Testing. The implementation was done by adapting the Levenberg-Marquardt (LM) algorithm using the TensorFlow package [61].

5.4 Results

The relationship between the model residuals and tuning parameters for the Solar cell data is displayed in Fig. 4(a), while the relationship between the structural complexity (computed from Θ^*) and the residuals corresponding to each model is displayed in Fig. 4(b). Observe from Fig. 4(a) that models with low residual values contain high *complexity* (contain more non-zero parameters in Θ^*), while models with higher residual values and low *complexity* contain fewer number of non-zero parameter values in Θ^*. Thus, peaks correspond to over-fitted models while under-fitted models are represented by troughs in Fig. 4(a). This implies that the peaks and troughs in Fig. 4(a) correspond to small and large pairs of λ_1 and λ_2 values respectively. By aligning candidate compact models between the lower and higher parts of Fig. 4(a), models lying on the *plateau* in Fig. 4(a) can be selected and analyzed to determine their performance on the output. Next we compute general information criterion (GIC) values using the non-zero parameters in each Θ^* corresponding to pairs of (λ_1, λ_2) (Fig. 4(a)). The computed GIC values can provide a useful insight on how to devise a formulation for model selection.

Examples of models containing the selected most informative inputs are presented in Fig. 4(c) and (d). For each data set, a total of $20 \times 20 \times 10 = 4000$ runs corresponding to each pair of (λ_1, λ_2) were performed by assigning 50 different values to Θ. Suitable values for λ_1 and λ_2 were obtained via a 2-dimensional grid search by comparing model residuals and structural complexity [62]. For each pair of $(\lambda_1\ \lambda_2)$, a weighting term was computed using the Generalized Information Criterion (GIC) (with Akaike (AIC) and Bayesian Information Criteria (BIC) as special cases) [63]. Each (λ_j, λ_k), $j, k, = 1, \ldots, 20$ was selected in the interval $[0, 5] \times [0, 5]$ in steps of 0.001, while the GIC was computed from:

$$GIC_\lambda = loss \quad + \quad \kappa_n df_\lambda \tag{16}$$

where $\kappa_n \in \mathbf{R}$. The complexity weighted term df_λ, is computed from the non–zero parameters in each Θ^*. For each group of candidate models, the same number of inputs were selected irrespective of the number of non-zero parameters in Θ^*. In order to devise a mechanism for model selection, a global minimum value was obtained from several local *minima* of GIC values. This was done by identifying models which lie along the "*dark region*" (corresponding to the global minima of the GIC values) as shown in Fig. 4(b).

For model selection, we resort to a modified form of Pareto ranking [56, 64] – by ranking candidate models using their residual values and structural complexity values. The candidate models are selected by comparing their accuracy and performance on new data sets. As seen in Fig. 4(b), models whose GIC point values lie on the *plateau* were selected from an ensemble of candidate models which lie within an artificially defined Pareto boundary [65]. The relationship between the predicted and true output labels for the Solar and Artificial Test Sets are displayed in Fig. 5.

The plots in Fig. 6 correspond to the relationship between the error and training epochs for each data set. The results indicate convergence when the classical back-propagation algorithm is adapted for input selection.

In Fig. 7, we compare the relationship between the PCA values and the feature contributions of original and selected features from the WCDS data [59]. From Fig. 7 (a), we observe that the most significant relationship can be described using 5 features (as shown at the peak positions of the graph), while 6 features were selected via our approach. Similarly, in Fig. 8, the relationship between the PCA values for the original and selected features for the Heart disease data 1 are displayed. From Fig. 7(a) and (b), the output can be described by 28 input features obtained using our approach. Also, the corresponding accuracy and other essential statistics will be presented in Table 3.

Observe from the results that the proposed feature selection mechanism provides more information for the selected input features in contrast to the PCA which projects relevant features along principal components. Thus, in situations where information about the selected inputs are significant in interpreting the modelling outcome, applying PCA is limited since it does not provide selected feature information.

5.5 Performance analysis

In Fig. 9, we compare essential performance statistics for models trained using the optimally selected input features and compute the corresponding Accuracy on the Testing Set. We also present the corresponding True Positives (TP), False Positive (FP), True Negative (TN) and False Negative (FN) for the WCDS data set. The comparison is made between our approach (OA) and several well-known classical methods (the Random Classifier (RF), Lasso (L1), Dropout (DP), Xgboost (Xbc) and the Elastic Net (EN) [52, 53, 66]). We observe that 6 input features were selected via using our approach while 8 – 9 features were selected using RF and XbC. On the other hand, deciding on how to set the threshold value when using both RF and XbC is critical. Thus, choosing non suitable threshold values can result in less optimal inputs being selected and lead to loss of vital information in the resulting model. Furthermore, the results also indicate that over-fitting can be prevented via our approach as seen from the accuracy of Test Set, in addition to input feature selection and topology optimization during training. Also, we note that automatic fine-tuning is performed implicitly using our approach. Meanwhile, no input features were selected using the EN, Lasso and Dropout methods. Also, highly accurate models were obtained when the mechanism was validated on the Testing Set when compared with these methods.

Similarly, Fig. 10 we present the performance statistics results for the Heart Disease Data set. From the results, the accuracy on the Testing set is dependent on the number of optimal features. Observe that the performance of our approach on the Testing data set is higher when compared to other classification algorithms. This can attributed to the composition of the original data due to cases of missing values. Also, the correct and miss classified output labels obtained by retraining the Heart disease and WCDS sets using the most informative feature subsets using different classification algorithms (DP, RF [52], L1 and (EN)) are displayed in Figs. 9 and 10 respectively. The statistical quantities (TP, FP, TN and FN) in Figs. 10 and 9 correspond to the Heart disease 1 and WCDS Test sets respectively. In both Figs. 10 and 9, TP represent the number of participants that were correctly classified as belonging to class 1 and TN for class 0. The correctly classified participants are represented by TP and TN, while the miss classified participants are represented ny FN and FP.

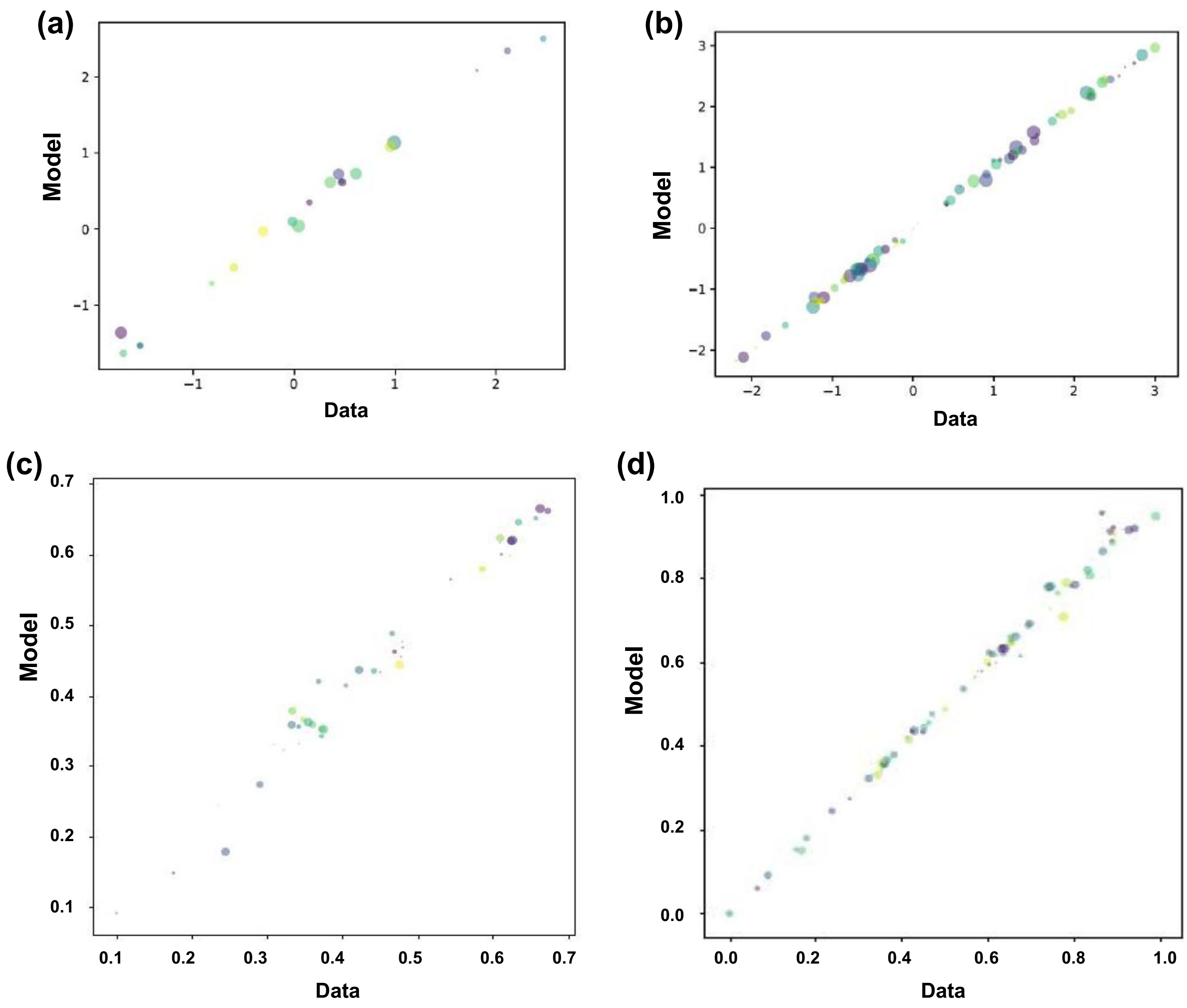

Fig. 5 Scatter plots corresponding to the Testing and Training data of the artificial data set are depicted in (a) and (b) respectively, while (c) and (d) represent the Testing and Training data of the Solar cell data set

The TP and TN in Figs. 9 and 10 results also indicate that very few miss-classified were identified as seen from the few FP and FN, and high TP and TN values. The accuracy obtained by retraining the selected features using different classification algorithms (L1, XbC, DP, EN, RF [53]) are also shown. The results also indicate that a higher Accuracy for the Testing set was obtained via our approach when compared with different algorithms in Fig. 10. Even though the Lasso required a minimum training time, it did not result in input feature selection.

In Table 1, we compare the performance statistics between our approach and other classical training methods [28, 41, 53, 67]. From the results, we observe a similarity in the Testing set accuracy. But our approach has the added benefit that under-fitting, over-fitting, most informative inputs and optimal topology can be simultaneously selected by choosing suitable λ_1 and λ_2 values during training.

In contrast to our approach, we also observe that the most informative inputs cannot be selected using EN, DP and L1, while statistical ranking and feature importance are required for both Xbc abd RF. We also note that the results obtained from the classical approach in Table 1 constitute a form of systematic approach which is applied during or after training. However, both RF and Xbc are done by using statistical ranking and feature importance whereby, informative inputs are selected if their computed feature importance values are less than a manually selected threshold value. This can result in the loss of vital information especially in non-linear high dimensional problems. Also, informative inputs and relevant features

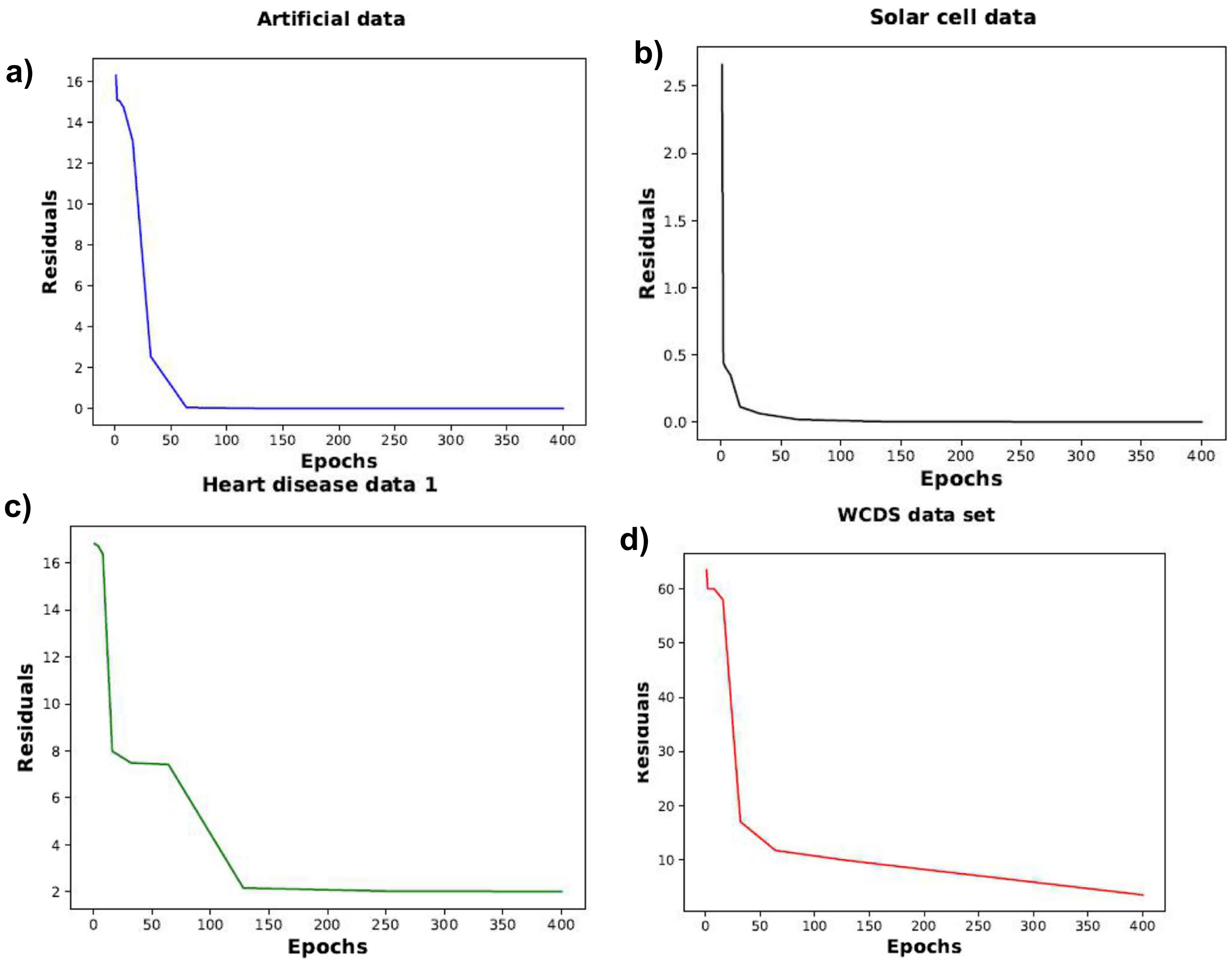

Fig. 6 Relationship between the residues and Epochs corresponding to the (a) Artificial (b) Solar cell, (c) Heart disease 1 and (d) WCDS data sets

selected using these classical methods are done in two-steps in contrast to our one-step approach. The implementation was done by training the data set using the [59].

In Tables 2 and 3, we analyze different statistical quantities corresponding to the WCDS data set with Feature Selection (FS) and without feature selection (WFS). While in Table 2, we compare vital performance metrics (Precision, recall, f-score) [59], and present the number of correctly and miss classified outputs in both instances (WFS and FS).

In Table 3, we compare the statistical quantities from Table 2 in both instances (WFS and FS) for the Heart disease 1 data set. The results indicate that a high accuracy was obtained when training was performed using the most informative inputs.

The resulting True Negative (TN), False Positive (FP), False Negative (FN) and True Positive (TP) values are displayed in Fig. 9. In Table 4, we compare the results between the original and selected inputs when l_2 and Lasso are incorporated in different ANN layers. The results indicate that a significant reduction in the number of inputs, parameters and neurons were obtained in each layer. Models containing the optimally selected inputs were trained using different classification algorithms and the results are shown in Fig. 10 (a) - (b).

In Table 5, we compare the number of non-zero values in Θ and Θ^* for each data set. The results also indicate that a significant reduction in the number of input features was obtained using our approach. Also included in the results are suitable values for λ_1, λ_2 and the errors corresponding to the selected models for each data set.

5.6 Computational and time complexity

The analytical computational complexity associated with the proposed input selection technique can be obtained by using foundational concepts derived from classical

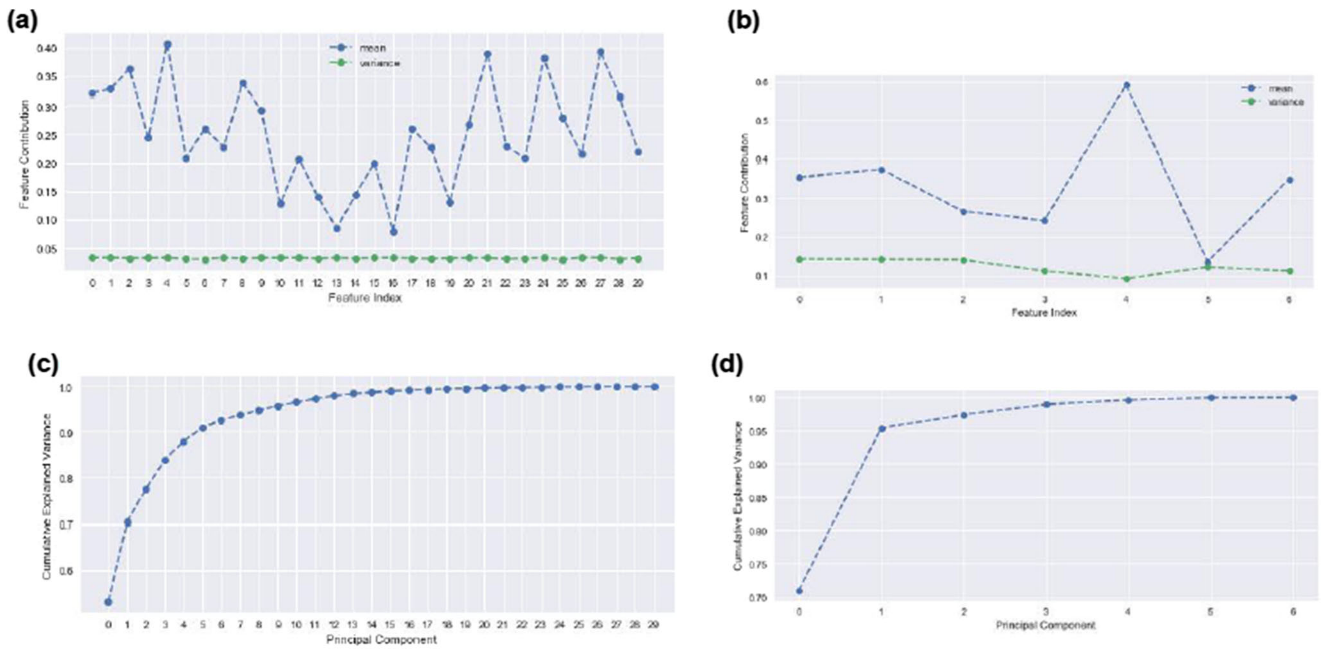

Fig. 7 A comparison between the PCA contribution for the (a) original and (b) selected features, including the cumulative explained variance and principal component for the (c) original and (d) selected features for the WCDS data set

results. Based on the results of the analytical computational complexity of the Lasso penalty derived from [68], let n be the number of inputs and N be the data set points. The Lasso complexity is given by $\mathcal{O}(n^3+n^2N)$, and if $n < N$, we have that $n^3 < n^2N$, while then the computational complexity of the Lasso penalty equals n^2N, and is dependent on the given problem.

The computational complexity for structured l_2 penalty can be obtained by using the results corresponding to the Lasso penalty. For instance, two key concepts considered in our analysis include: composition of the data sets consisting of N points and n features. Note that the complexity for small to medium size data sets with smaller N will scale below $\mathcal{O}(n^3 + n^2N)$. To derive the complexity associated

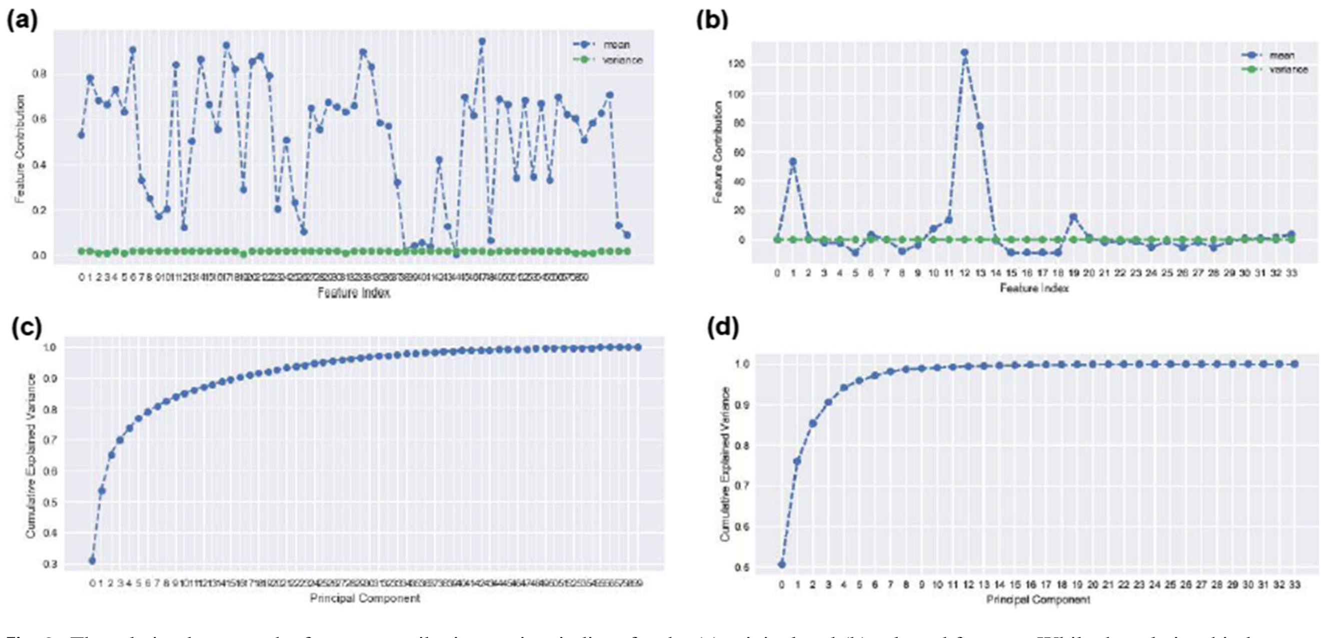

Fig. 8 The relation between the feature contribution against indices for the (a) original and (b) selected features. While the relationship between the cumulative explained variance against PCA values for the original and selected features of the Heart Disease 2 data set are depicted in (c) and (d) respectively

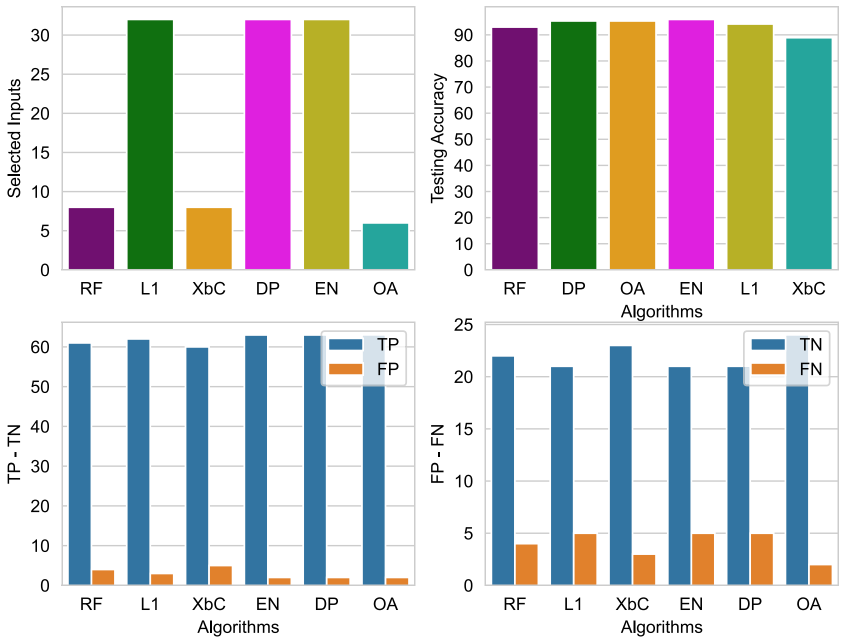

Fig. 9 Comparing the performance statistics (Testing Accuracy, optimally selected features, True Positives (TP), False Positive (FP), True Negative (TN) and False Negative (FN)) between our approach and conventional methods. Results were obtained from feature ranking using Random Classifier (RF), Lasso (L1), Dropout (DP), Xgboost (Xbc), Elastic Net (EN) and Our Approach (OA) for the WCDS data set containing $n = 32$ inputs

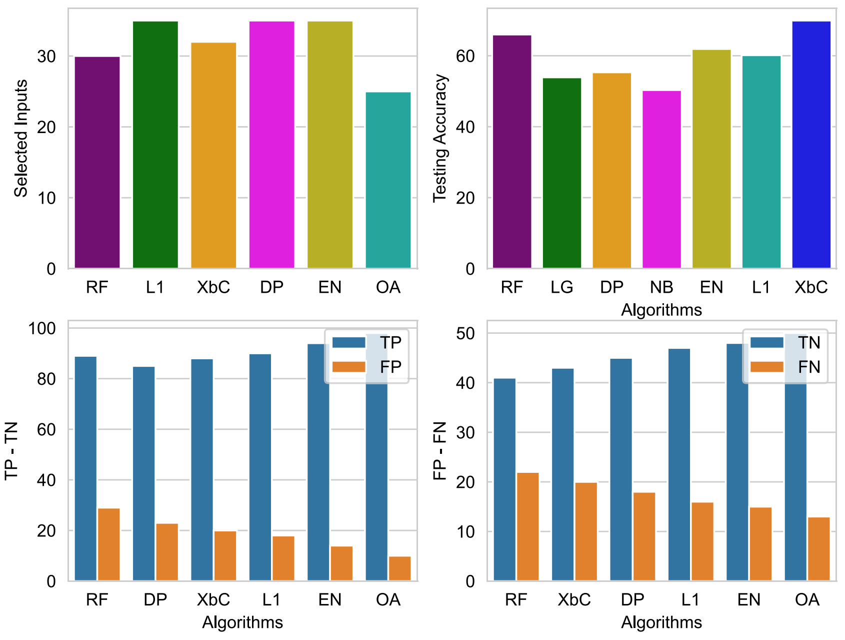

Fig. 10 Comparing the most essential performance statistics (Testing data accuracy, optimal features, True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) output labels) between our approach (OA) and classical methods for the Heart data set with initial $n = 35$ inputs. Results were obtained from via ranking using Random Classifier (RF), Lasso (L1), Dropout (DP), Xgboost (Xbc), Elastic Net (EN) and OA

Table 1 Comparing performance statistics (accuracy, optimal features, loss) for the Training and Testing data sets between our approach and conventional methods (Random Classifier (RF), Lasso (L1), Dropout (DP), Xgboost (Xbc), Elastic Net (EN) and Our Approach (OA)), corresponding to the WCDS data set, which initially contains $n = 32$ inputs

Methods	Selected	Error	Training Data	Testing Data
RF	7-8	0.07	0.94	0.93
DP	32	0.08	0.93	0.91
Xbc	8	0.05	0.94	0.93
L1	32	0.09	0.93	0.93
EN	32	0.07	0.94	0.92
OA	6	0.09	0.94	0.93

Table 2 Results showing the Accuracy and statistical quantities (Precision, F-Score and recall) for the WCDS data sets

	Precision		recall		f–score		support
	WFS	FS	WFS	FS	WFS	FS	
0	0.93	0.95	0.96	0.945	0.91	0.93	71
1	0.97	0.98	0.93	0.90	0.93	0.94	43
accuracy					0.95	0.96	114
macro avg	0.95	0.96	0.94	0.95	71	0.96	114
weighted avg	0.95	0.96	0.94	0.96	71	0.96	114

Also displayed are the number of correctly predicted outputs for the Testing Set

Table 3 Accuracy and statistical results for the Heart disease data sets

	Precision		recall		f–score		support
	WFS	FS	WFS	FS	WFS	FS	
0	0.88	0.86	0.87	0.86	0.86	0.865	149
1	0.87	0.87	0.88	0.88	0.87	0.868	121
accuracy					0.89	0.87	270
macro avg	0.88	0.87	0.87	0.85	0.89	0.87	270
weighted avg	0.88	0.86	0.88	0.86	0.90	0.88	270

Table 4 A comparison between the original and the optimally selected inputs for all data sets

Data set	Original Features	Selected Features
Full	8	6
Solar	7	4
Heart Disease 1	22	9
Heart Disease 2	74	35
WCDS	30	7
PD Data	300	35

The results indicate a significant number of inputs were selected in each case using our approach

Table 5 Comparing the resulting initial, optimal neurons and parameters for different data sets selected using the proposed mechanism

Data set	Θ	Θ^*	λ_2	λ_1	Error
Full	150	16	0.009–0.003	0.008–0.05	0.05
Solar	128	35	0.004–0.008	0.001–0.006	0.09
Heart Disease	150	28	0.07–0.2	0.04–0.2	1.2
WCDS	757	178	0.008–0.04	0.01–0.06	0.78
PD Data	1520	210	0.008–0.04	0.01–0.04	0.18

Table 6 Comparing training and feature selection times for different algorithms obtained through Feature ranking (FeatRank), Feature Importance (FeatRel), Lasso and our approach

Method	Time (seconds)
FeatRank	17.972
FeatRel	13.241
Lasso	5.34
Our Approach	15.72

Out method compares with existing methods and leads to optimal input feature selection

with the search space for both λ_1 and λ_2, eliminating large and small values for λ_1 and λ_2 will narrow the search interval resulting in a reduction in the computational complexity of the grid space. This implies that there will be no significant increase in the value of the computational complexity will be introduced by applying the proposed mechanism for feature selection.

In Table 6, we present the time obtained by training the WCDS Data set with $n = 32$ inputs and $N = 600$ points. Observe that apart from the Lasso, the time required in in our approach is comparable to Xgboost with feature ranking (FeatRank) and feature importance (FeatRel). The training was performed on an Intel Computer with an i7-8550U (8 cores) using 4.000GHz Processor containing a 16 Random Access Memory (RAM) size. This shows that the time derived from our approach is comparable to other methods and do not add any significant over head when compared with the training times obtained from other approaches.

6 Discussion and conclusion

In this paper, an embedded, continuous mechanism was proposed to optimally select the most informative features from data sets. The technique is based on incorporating a novel structured l_2 penalty to the input layer of an ANN structure. It relies on the ability of l_2 norms to simultaneously shrink a group of parameters corresponding to a given input feature towards zero. The introduction of the structured l_2 penalty transforms the objective function from an ill to a well-posed optimization problem which was solved by adapting the classical LM back-propagation algorithm. In principle, the implementation was done by devising a modified gradient-based technique whereby, the contribution of a parameter group (associated with a given input) to the structured penalty was compared to its contribution to the loss function. The method also enforces the importance of *parsimonious* models – a well-known phenomenon in statistical learning literature – where models containing fewer parameters are preferred in applications. The results show a significant reduction in the number of input features when applied to different data sets. When combined with the Lasso, the method resulted in the simultaneous input feature selection and topology optimization. Hence, the results also indicate that the most informative input features can be selected using the proposed mechanism via adaptive training.

Further improvements can be made on the formalism by adapting the modified gradient technique to deal with data sets containing redundant inputs. Also, extending the influence of the modified gradient step *globally* during training can improve convergence. Meanwhile, analyzing the *interaction* between *non-differentiable* gradients of the structured and other sparsity inducing penalties can reduce unwanted outliers in the models. Furthermore, the tuning parameters search space can be expanded in order to facilitate the interpretation of model complexity.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was supported by the Joint Research Center for Computational Bio-medicine, RWTH Aachen University and partly by the European Marie-Curie Framework grant FP7/2007-2013.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Wang J, Zhang H, Wang J, Pu Y, Pal YR (2021) Feature selection using a neural network with group lasso regularization and controlled redundancy. IEEE Trans Neural Netw Learn Syst 32(3):1110–1123
2. Chandrashekar C, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28
3. Dhal P, Azad C (2021) A comprehensive survey on feature selection in the various fields of machine learning. Appl Intell
4. Verikas A, Bacauskiene M (2002) Feature selection with neural networks. Pattern Recogn Lett 23(11):1323–1335
5. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 55:436–444
6. Domingos P (2012) A few useful things about machine learning. Commun ACM 55(10):78–87
7. Schmidhuber J (2015) Deep learning in neural networks: An overview. Neural Netw 61:85–117

8. Rumelhart D, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536
9. Barron AR (1993) Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans Inf Theory 39(3):930–945
10. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward neural networks are universal approximators. Neural Netw 2(5):359–366
11. Kurkova V (1992) Kolmogorov's theorem and multilayer neural networks. Neural Netw 5(3):501–506
12. Blum AL, Langley P (1997) Selection of relevant features and examples in machine learning. Artif Intell 97:245–271
13. Nguyen BH, Xue B, Zhang M (2020) A survey on swarm intelligence approaches to feature selection in data mining. Swarm and Evolutionary Computation, 54(100663)
14. Chen CW, Tsai YH, Chang FR, Lin WC (2020) Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. Expert Syst
15. Blessie E, Eswaramurthy K (2012) Sigmis: A feature selection algorithm using correlation based method. J Algorithm Comput Technol 6:385–394
16. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422
17. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97:273–324
18. Xiaoping L, Yadi W, Ruben R (2020) A survey on sparse learning models for feature selection. IEEE Transactions on Cybernetics, pp 1–19
19. Got A, Moussaoui A, Zouache D (2021) Hybrid filter-wrapper feature selection using whale optimization algorithm: A multi-objective approach. Expert Syst Appl, 183
20. Kira K, Rendell LA (1992) A practical approach to feature selection. Machine Learning Proceedings 1992, pp 249–256
21. Kononenko I (1994) Estimating attributes: Analysis and extensions of relief. Machine Learning: ECML-94, pp 171–182
22. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27:77–93
23. Raileanu LE, Stoffel K (2004) Theoretical comparison between the gini index and information gain criteria. Ann Math Artif Intell 41:77–93
24. Scardapane S, Hussain A, Uncini A (2017) Group sparse regularization for deep neural networks. Neurocomputing 241(1):81–89
25. Kong D, Fujimaki R, Liu J, Nie F, Ding C (2014) Exclusive feature learning on arbitrary structures via $l_{1,2}$ - norm. Advan Neural Process Syst (NIPS) 24(258):241–258
26. Labach A, Salehinejad H, Valaee S (2019) Survey of dropout methods for deep neural networks. CoRR abs/1904.13310
27. May R, Dandy G, Maier H (2011) Review of input variable selection methods for artificial neural networks. Methodological Advances and Biomedical Applications
28. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
29. Stalin S, Sreenivas TV (2002) Vectorized backpropagation and automatic pruning for mlp network optimization. IEEE International Conference on Neural Networks
30. Gethsiyal MA, Kathirvalavakumer T (2011) A novel prunning algorithm for optimizing feedforward neural networks of classification problems. Neural Process Lett 34(3):241–258
31. Hui Z, Hastie T, Tibshirani R (2006) Sparse principal component analysis. J Comput Graph Stat 2:265–286
32. Saxena DK, Duro JA, Tiwari A, Deb K, Zhang Q (2013) Objective reduction in many-objective optimization: linear and nonlinear algorithms. IEEE Trans Evolut Comput 1:77–99
33. Ioannou Y, Robertson D, Shotton J, Cipolla R, Criminisi A (2015) Training cnns with low–rank filters for efficient image classification. coRR abs/1511.06744
34. Doya K, Selverston A (1994) Dimension reduction of biological neuron models by artificial neural networks. Neural Comput 6:696–717
35. Bach FR, Jenatton R, Mairal J, Obozinski G (2012) Structured sparsity through convex optimization. Stat Sci 27(4):450–468
36. Wen W, Wu C, Wang Y, Chen Y, Li H (2016) Learning structured sparsity in deep neural networks. Proceedings of the 30th International Conference on Neural Information Processing Systems 24(258):2082–2090
37. Kim S, Xing EP (2012) Tree-guided lasso for multi-response regression with structured sparsity, with an application to eqtl mapping. Ann Appl Stat 6(3):1095–1117
38. Cruz JA, Wishart DS (2007) Application of machine learning in cancer prediction and prognosis. Cancer Informat 2(258):59–77
39. Kourou K, Exarchos TP, Exarchos KP, Karamouzis MV, Fotiadis DI (2015) Machine learning in cancer prognosis and prediction. Comput Struct Biotechnol J 13(258):8–17
40. Goerlitz L, Mrziglod T, Loosen R (2010) Topology optimization of artificial neural networks using l_1- penalization. Proc Work Comput Intell 33(20):80–87
41. Tibshirani R (1996) Regression shrinkage and selection via lasso. J Stat Soc B 58(1):267–288
42. Tibshirani R (1997) Regression shrinkage and selection via lasso. Stat Med 16(1–12):385–395
43. Liu Y, Wu Y (2007) Variable selection via a combination of the l_0 and l_1 penalties. J Comput Graph Stat 16(4):782–798
44. Jenatton R, Audibert JY, Bach F (2011) Structure variable selection with sparsity-inducing norms. J Mach Learn Res 12:2777–2824
45. Lemhadri I, Ruan F, Abraham L, R T (2021) Lassonet: A neural network with feature sparsity. J Mach Learn Res 22(5):1–29
46. Du G, Zhang J, Luo Z, Ma F, Ma L, Li S (2020) Joint imbalanced classification and feature selection for hospital readmissions. Knowledge Based Systems, 200
47. shao W, Peng Y, Zu C, Wang M, Zhang D (2021) Hypergraph based multi-task feature selection for multimodal classification of alzheimer's disease. Comput Med Imaging Graph 80:101663
48. Amini F, Hu G (2021) A two-layer feature selection method using genetic algorithm and elastic net. Expert Systems With Applications, 166
49. Zhang X, Fan M, Wang D, Zhou P, Tao D (2021) Top-k feature selection framework using robust 0-1 integer programming. IEEE Transactions on Neural Networks and Learning Systems, 32(7)
50. Zou H, Hastie T, Tibshirani R (2000) On the degrees of freedom of the lasso. The Annals of Statistics 5(35):2173–2192
51. Dehua W, Yang Z, Yi Z (2017) Lightgbm: An effective mirna classification method in breast cancer patients. Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics, pp 7–11
52. Dreiseitl S, Ohno-Machado L (2002) Logistic regression and artificial neural network classification models: A methodology review. J Med Inform, pp 352–359
53. Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. CoRR, abs/1603.02754, pp 785–794
54. Marquardt DW (1963) An algorithm for least–squares estimation of nonlinear parameters. J Soc Ind Appl Math 11(2):431–441
55. Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning. Springer Series in Statistics Springer New York Inc
56. Kim IY, de Weck OL (2006) Adaptive weighted sum method for multiobjective optimization: A new method for pareto front generation. Struct Multidiscip Optim

57. Dheeru D, Casey G (2017) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml
58. Tsanas A, Little MA, Fox C, Ramig LO (2014) Objective automatic assessment of rehabilitative speech treatment in parkinson's disease. IEEE Trans Neural Syst Rehabil Eng 22:181–190
59. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: Machine learning in python. J Mach Learn Res 12(Oct):2825–2830
60. Van Rossum G, Drake Jr FL (1995) Python tutorial. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands
61. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, I. Jia I, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) Tensorflow: Large-scale machine learning on heterogeneous systems. TensorFlow
62. Begastra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13:211–305
63. Zhang Y, Li R, Tsai CL (2010) Regularization parameter selection via generalized information criterion. J Am Stat Assoc 105(489):312–323
64. Luc DT (2008) Pareto optimality. Pareto Optimality, Game Theory and Equilibria. Springer Optimization and Its Applications, 17
65. Legriel J, Guernic CL, Cotton S, Maler O (2010) Approximating the pareto front of multi-criteria optimization problems. Tools and Algorithms for the Construction and Analysis of Systems, pp 69–83
66. Liaw A, Wiener M (2002) Classification and regression by randomforest. R News 2(3):18–22
67. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B 67:301–320
68. Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. The Annals of Statistics, 32(2)

Nathaniel Egwu graduated from the University of Nigeria; the African Institute for Mathematical Sciences, Capetown South Africa; the Eindhoven University of Technology, The Netherlands; the University of Kaiserslautern, Germany; and currently pursuing his PhD research at the Bayer AG/JRC-RWTH Aachen University, Germany. His research interests are focused on the development and the application of novel machine, deep learning, computational and mathematical optimization techniques to model industrial processes with emphasis on the theoretical and practical aspects of such methods. Key aspects of his research interests also include developing strategies to deal with the *pitfalls* associated with conventional techniques in applications - in order to reduce cost, improve model generalization and other *key* parameters in control and production.

Thomas Mrziglod is head of the applied mathematics group at Engineering and Technology at Bayer company in Germany. He got a PhD in mathematics from University of Cologne in 1989 and had several research and development positions in chemical and pharmaceutical industry. Since 2016 he is head of the applied mathematics group. In addition, he gives lectures in applied mathematics at the University of Cologne.

His focus is on development and application of machine learning, deep learning, and hybrid modelling techniques in life science industry.

Andreas Schuppert is Professor for Computational Biomedicine and founding director of the Joint Research Center for Computational Biomedicine at RWTH Aachen University. He studied Physics and got a PhD in mathematics from University Stuttgart, followed by research and development positions in chemical-pharmaceutical industry. In 2007 he became Adjunct Professor at RWTH Aachen, where he became founding director of the Joint Research Center for Computational Biomedicine in 2013, a private-public partnership between RWTH Aachen University Hospital Aachen and Bayer AG. Since 2017 he is head of the Institute for Computational Biomedicine.

His focus is on research and development of hybrid modelling technologies with focus on applications in intensive care, oncology, pain research. During the Covid19 pandemics he developed the DIVI prognosis tool and focused on pattern recognition in pandemic dynamics.