# Comparative Analysis of Feature Selection and Optimisation Algorithms For Classification Tasks

Ankit Gupta
*Faculty of Exact Sciences and Engineering*
*University of Madeira*
Madeira, Portugal
2118619@student.uma.pt

Diogo Freitas
*Faculty of Exact Sciences and Engineering*
*University of Madeira*
Madeira, Portugal
2019214@student.uma.pt

***Abstract*—This paper analyses how the Levenberg–Marquardt backpropagation algorithm (LMA) and the Particle Swarm Optimisation (PSO) can be used as a training algorithm of Artificial Neural Networks (ANN), when subject to different feature selection mechanisms, in this case, the Minimum Redundancy and Maximum Relevance (mRMR) and the Chi-square test. In this view, four data sets were tested with different ANN architectures, aiming to select the best combination of optimisation algorithm-feature selection that leverage the accuracy of the models. Thus, the comparison between the different combinations was made in terms of maximum, minimum and average accuracy value provided by the confusion matrices. Our results demonstrated that, on average, the best combination is the use of PSO with the features selected by the Chi-square process.**

***Index Terms*—levenberg–marquardt, LMA, particle swarm optimisation, PSO, feature selection, chi-square, mRMR**

## I. Introduction

Artificial Neural Networks (ANN) are biologically inspired tools that were designed to identify patterns in the data and to find complex relationships between the independent and dependent variables.

Each ANN has a set of artificial neurons that are organised in layers and interconnected, as a resemblance with the human brain, by synaptic connections. Every synaptic connection has, in turn, a weight, that will be adjusted according to a learning algorithm, taking into consideration the examples provided and the feedback from the error.

Today, ANNs are being used, e.g., on self-driving cars, spam or non-spam prediction, detection of fraud, anomaly detection. ANNs can perform, thus, a variety of tasks, such as classification, regression applications, clustering, and predictive modelling. This study is focused on exploring the ANNs' capabilities to classification problems using four different data sets.

The feedforward ANN topology will be the only one considered in this work, in which the neuron units are only connected to the succeeding layer, i.e., the information flows in only one direction.

Besides that, two optimisation algorithms will be considered and compared. On the one hand, the Levenberg–Marquardt algorithm (LMA) is an exact optimisation algorithm that is used to solve the least-squares fitting problem, based on the derivative information. On the other hand, the Particle Swarm Optimisation (PSO) algorithm is a stochastic optimisation algorithm that uses particles and the information, acquired by these during the search space, to find the position that most minimises the error.

Before starting to train the networks with the two learning algorithms, two processes of feature selection were used, namely, the Minimum Redundancy and Maximum Relevance (mRMR) and the Chi-square tests. It is important to note here that feature selection is an important task in machine learning and data mining problems since it simplifies the models and improves the generalisation capabilities of the model.

In this view and based on the accuracy given by the confusion matrix, is possible to choose for the data sets considered what is the best optimisation algorithm and what is the best feature selection process that leverage the accuracy of the models. The comparison will take into account the maximum, minimum and average accuracy value provided by the confusion matrices, obtained in four different data sets.

### A. Parkinson's

The Parkinson's data set is composed of 195 biomedical voice measurements from 31 male and 23 female subjects with Parkinson's disease (PD), aged between 46 to 85 years. In this data set, each row corresponds to one of the 195 voice recording and each column to each of the 21 signals of the phonations.

Since people with PD often find difficulties to produce vocal sounds and have problems with the normal articulation of speech, the main objective with this data set is to, based on the acoustic measures, detect healthy people from those with PD.

### B. Car evaluation

The car data set contains 1728 samples of multivariate attributes for the evaluation of cars. The attributes are buying, maint, doors, persons, lug_boot, safety. All attributes are categorical having several categories ranging from 3 to 4. The categorical values for input attributes and the class labels are shown in Table I. There are four labels for this data set and are unacc, acc, good, vgood. It is important to note here that

it was used one approach of ordinal encoding for converting text to numerical data to be feed into the classifier.

TABLE I: Car Data Set Attributes Information.

| Attribute Name | No. Categories | Values |
|---|---|---|
| buying | 4 | vhigh, high, med, low |
| maint | 4 | vhigh, high, med, low |
| doors | 4 | 2, 3, 4, 5more |
| persons | 3 | 2, 4, more |
| lug_boot | 3 | small, med, big |
| safety | 3 | low, med, high |

### C. Wine Quality

The data set was prepared for red and white wine of the Portuguese "Vinho Verde" wine, with the same number of attributes. It consists of a total of $4\,898$ samples ($1\,600$ for red and $3\,298$ for white wine) and twelve features. The objective with this data set is to, based on the other attributes in the data set, predict the wine quality classification.

This paper is organised as follows: in Section II, the methods and algorithms that will be used in this work for feature selection and for adjusting the ANNs' weights are presented. In turn, Section III presents the results of the comparison for each data set between the LMA and PSO and the two feature selection process. Finally, the last section presents some concluding remarks.

## II. Methods and Algorithms

### A. Feature Selection

Feature selection is an essential task in machine learning algorithms and statistics. The objective of the feature selection is to select a subset of features, from all features collected, according to their importance to the output variable(s).

By just selecting a subset of features, one is simplifying the model, requiring less computational resources to compute it (e.g., execution time and memory), and also avoiding that the model learns specific information about the data set, such as noise (i.e., avoid overfitting).

The present study uses data sets with categorical and continuous attributes. Hence, feature selection methods supporting both types of features were needed to be chosen. Two feature selection methods namely the Minimum Redundancy and Maximum Relevance (mRMR) and the feature selection using Chi-square tests were chosen. The performance of feature selection methods is analysed using the accuracy post-classification on a small subset of samples from the data set, that was not used for training the classifier (i.e., the test data set).

*1) Minimum Redundancy and Maximum Relevance:* The Maximum Relevance Minimum Redundancy (mRMR) [2] employs mutual information for calculating the maximum relevance between features and the corresponding labels, and minimum redundancy between all feature vectors of the data set. That is, features are selected according to the correlation to the classification variable, and also according to the multi-collinearity of each independent variable.

The built-in MATLAB's function `fscmrmr` was used for implementing the mRMR method, which calculates a score based on maximum relevance and minimum redundancy.

The threshold for selecting the optimal feature set was set to be the mean of scores for each feature given by the method. Hence, those features above the threshold were excluded from the feature set.

*2) Feature selection using Chi-square tests:* This method is a univariate feature selection method which uses Chi-square tests for checking the correlation between the feature and the label vector. It calculates the p-value of the test statistic which signifies the inclusion of a particular variable in the optimal feature set. The lower the p-value for the feature-label pair, the higher is its importance.

The built-in function `fscchi2` was used for implementing the Chi-square-based feature selection method, which calculates a score based on the p-value. The final score is calculated by taking the negative log of the p-value for each feature-label pair. A higher score indicates the importance of a particular feature. The threshold used for the inclusion of features in the optimal feature set is the mean of scores for each feature.

### B. Levenberg–Marquardt Backpropagation

The Levenberg–Marquardt algorithm (LMA) is an optimisation algorithm, which is used to solve the least square fitting problem. It was suggested by Kenneth Levenberg and improved later by Donald Marquardt [5], [6].

Mathematically, given a set of pairs of dependent $(x_i)$ and independent variable $(y_i)$ as $(x_i, y_i)$, $i = 1, 2, 3, \cdots, n.$, the objective is to find the optimal set of parameters ($\mathbf{P}$), such that the sum of squared error deviation $\mathbf{S}(\mathbf{P})$ can be minimised as follows:

$$\hat{\mathbf{P}} \in \text{argmin}_{\hat{\mathbf{P}}}\ \mathbf{S}(\mathbf{P}) \equiv \text{argmin}_{\hat{\mathbf{P}}} \sum_{i=1}^{m} \Big[ y_i - f(x_i, \mathbf{P}) \Big]^2, \quad (1)$$

being $m$ the number of samples in the data set.

The LMA is an iterative procedure which starts with random initialisation of $\mathbf{P}$ (vector or scalar), and at each iteration, $\mathbf{P}$ is updated, i.e., $\mathbf{P}$ is updated to $\mathbf{P} + \delta$.

For $\delta$ value determination, $f(x_i, \mathbf{P}+\delta)$ has to be linearised, so the equation, according to the Taylor series expansion, becomes:

$$f(x_i, \mathbf{P} + \delta) \approx f(x_i, \mathbf{P}) + \mathbf{J}_i \delta, \quad (2)$$

where $\mathbf{J}$ is the gradient of $f$ with respect to $\mathbf{P}$ and it is given by (Jacobian matrix):

$$J_i = \frac{\partial f(x_i, \mathbf{P})}{\partial \mathbf{P}}. \quad (3)$$

The minimum of $S(\mathbf{P})$ is found at $\mathbf{J}\delta = 0$. The first-order approximation for $f(x_i, \mathbf{P} + \delta)$ is:

$$S(\mathbf{P} + \delta) = \sum_{i=1}^{m} \Big[ y_i - f(x_i, \mathbf{P}) - \mathbf{J}_i \delta \Big]^2. \quad (4)$$

The vectorised form of the above equation is as follows:

$$S(\mathbf{P} + \delta) = \| \mathbf{y} - \mathbf{f}(\mathbf{P}) - \mathbf{J}\delta \|^2 . \quad (5)$$

Differentiating the above term with respect to the $\delta$ and equating it to 0 gives:

$$(\mathbf{J}^T\mathbf{J})\delta = \mathbf{J}^T\Big[\mathbf{y} - \mathbf{f}(\mathbf{P})\Big], \tag{6}$$

where $J$ is the singular Jacobian matrix of size $m \times n$, and $\mathbf{y}$ and $\mathbf{f}(\mathbf{P})$ are the vector components for the $i^{\text{th}}$ sample $y_i$ and $f(x_i, \mathbf{P})$, respectively.

The above equation is given by Marquardt. In turn, Levenberg added a parameter, $\lambda\mathbf{I}$, making it the damped version:

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\delta = \mathbf{J}^T\Big[\mathbf{y} - \mathbf{f}(\mathbf{P})\Big], \tag{7}$$

where $\mathbf{I}$ is an identity matrix and thus:

$$\delta = \Big[\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}\Big]^{-1}\mathbf{J}^T\Big[\mathbf{y} - \mathbf{f}(\mathbf{P})\Big]. \tag{8}$$

The $\lambda$ is adjusted at each iteration of the algorithm to ensure the minimisation of $S(\mathbf{P})$.

The LMA is also called the mixture of Gauss–Newton method and Gradient Descent. If $S(\mathbf{P})$ is minimised at a faster rate, a small change in $\lambda$ is required. This makes the algorithm to behave like Gauss–Newton method. On the other hand, if the change in $\lambda$ is not sufficient enough for minimisation, then $\lambda$ has to be increased, leading to the algorithm to taking the steps in the direction of the negative gradient. This makes the algorithm to behave like Gradient Descent.

The initialisation of $\lambda$ is random. At the next step, it is decreased by $\frac{\lambda}{v}$, where $v > 1$. If this does not minimise the $S(\mathbf{P})$, then the value of $\lambda$ can be increased by a factor of $v^k(\lambda v^k)$, being $v$ and $k$ chosen by the user/researcher.

### C. Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is a stochastic optimisation technique for linear and nonlinear functions (also known as fitness function), suggested by Eberhart and Kennedy [3], [4] in 1995.

PSO starts with random initialisation of particles between the boundaries of the $d$-dimensional search space of the fitness function. Each particle has its fitness value, which is decided by the fitness function, a position, and velocity, which enables it to move in the problem space.

The PSO algorithm is an iterative process which tends to find the optimal solution by the movement of particles. In each iteration, every particle is updated with its best fitness value $p_{\text{best}}$ and global best fitness value achieved by any particle in the swarm $g_{\text{best}}$. These values are then used for updating the particle velocity, such as [7]:

$$v_{\text{new}} = \omega \cdot v + l_1 \cdot r_1 \cdot \Big(p_{\text{best}} - x\Big) + l_2 \cdot r_2 \cdot \Big(g_{\text{best}} - x\Big), \tag{9}$$

where $v_{\text{new}}$ is the new velocity, $\omega$ is known as the inertia term and controls the influence of the previous velocity in the next particle's velocity $v$. $l_1$ and $l_2$ are the learning parameters, and $r_1$ and $r_2$ are random numbers with values ranging between 0 and 1. Finally, $x$ is the current position of the particle.

In turn, the position of the particle is updated as:

$$x_{\text{new}} = x + v_{\text{new}}. \tag{10}$$

The algorithm stops on reaching the maximum number of iterations or when an appropriate solution is found, according to a predefined accuracy, being the output of the algorithm the position of the best particle in the swarm.

In order to use the PSO algorithm to find the optimal weights of an ANN, each particle can be seen as a different ANN. The weights of the ANN represent the position of the particle in the search space, and the fitness value is given by the MSE of the ANN in the training data set.

Particles move in the search space according to Equations (9) and (10), and the output of the algorithm is the position (i.e., the weights) of the best particle in the swarm in terms of the MSE in the validation data set.

## III. Results

This study is intended to perform a comparative analysis of two feature selection algorithms: mRMR and Chi-square-based feature selection and their impact on the ANNs with different optimisation algorithms. The two optimisation algorithms used for the study are the LMA and PSO algorithm. Further, the optimisation algorithms were also compared using accuracy on the test data set. In other words, we are comparing the pairs of feature selection and optimisation algorithms for solving classification problem using different data sets.

The ANNs for classification tasks were tested ten times for each number of neurons in the hidden layer considered: 4, 7, 10, 12, 15 and 20. Besides that, the initial weights were set in the range $[-2.4/I, 2.4/I]$, where $I$ is the number of inputs. In its turn, the inputs were normalised using the MATLAB's built-in function `mapstd`.

Each data set was split into a training (70%), validation (15%) and test data set (15%), being divided using blocks of sequential indices.

On the other hand, the number of epochs was defined to be 5 000, and the error limit was set to be zero. For this set of tests, the early stopping of the algorithm was not used.

The LMA algorithm was parameterised so that the initial $\lambda$ was set to be equal to 1, with increments of 10 and decrements of 1. On the other hand, 24 particles were used with PSO. Besides that, $\omega = 0.9$, $l_1 = 0.5$ and $l_2 = 0.3$.

Tests were executed from the above-mentioned data sets, under different experimental conditions. In these experiments, it was included a comparison between the performance of the LMA and PSO, when submitted to the features selected by the mRMR and Chi-square tests.

### A. Feature Selection

In this section, the features selected for training the ANNs will be presented. Features were obtained using, as previously mentioned, the mRMR and the Chi-squared test.

For each figure presented in the above sections, it will be included a horizontal line indicating the threshold for selecting the features. This threshold was obtained by computing the mean of scores for the set of features.

*1) Parkinson's Data Set:* According to Fig. 1 and Fig. 2, both mRMR and Chi-Square-based method selected the same number of features (7). However, the selected features were different.

For the mRMR, the selected features were: MDVP:Fhi(Hz), MDVP:Jitter(Abs), Jitter:DDP, MDVP:Shimmer(dB), Shimmer:APQ5, HNR and PPE. On the other hand, for the Chi-square tests: MDVP:Fo(Hz), MDVP:Jitter(Abs), MDVP:Shimmer, RPDE, DFA, spread1 and PPE. Thus, only the MDVP:Jitter(Abs) and PPE were the features that were selected by both methods.

It is important to note that although the score of the feature Jitter:DDP achieved the highest score using mRMR, the Chi-square-based method has not even selected it. This might be due to the high relevance of the feature to the response variable; however, due to lack of adequate correlation, Chi-square method did not choose it.

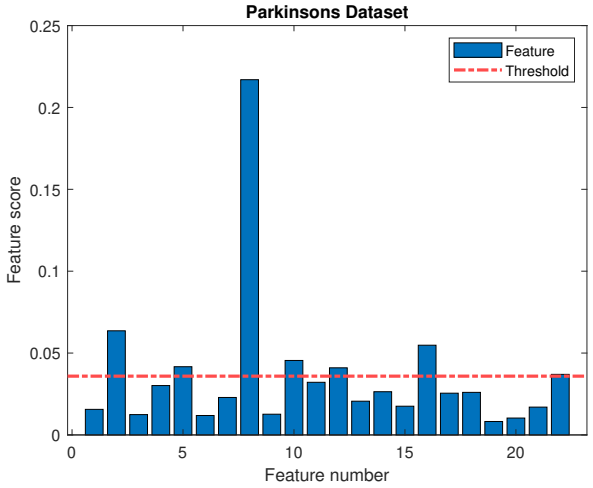


Fig. 1: Feature selection for the Parkinson's data set using mRMR.

*2) Car Evaluation:* The features selected by mRMR for the car evaluation data sets are maint and lug_boot whereas for Chi-square-based method the vital features for classification are doors and lug_boot, as can be seen by Fig. 3 and 4.

Like the Parkinson's data set, both feature selection methods have selected different features. Since both methods have selected one common and one different feature, it is difficult to select the best feature selection method from the two.

*3) Wine Quality:* Similar to the car and Parkinson's data sets, the mRMR and Chi-square-based method have selected for the red wine data set an equal number of features with two common features: volatile acidity and pH, as presented in Fig. 5 and 6.

Unlike all other data sets, where both selection methods choose a similar number of features, in the case of the wine data set, the mRMR selected 5 features (citric acid, residual sugar, chlorides, free sulfur dioxide and sulphates) and the Chi-square choose only 3 features (fixed acidity, residual sugar and total sulfur dioxide), as depicted in Fig. 7 and 8.

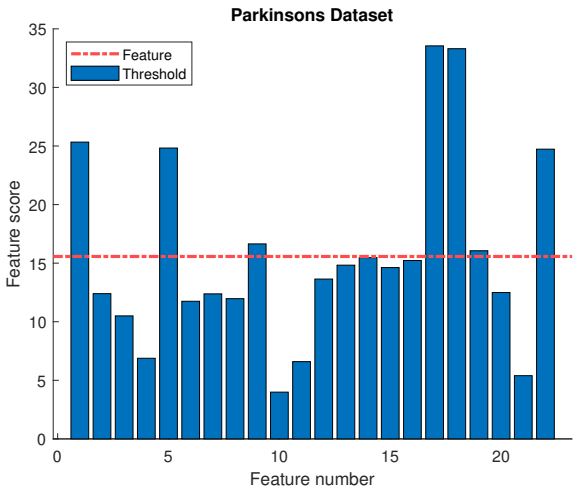


Fig. 2: Feature selection for the Parkinson's data set using Chi-square tests.

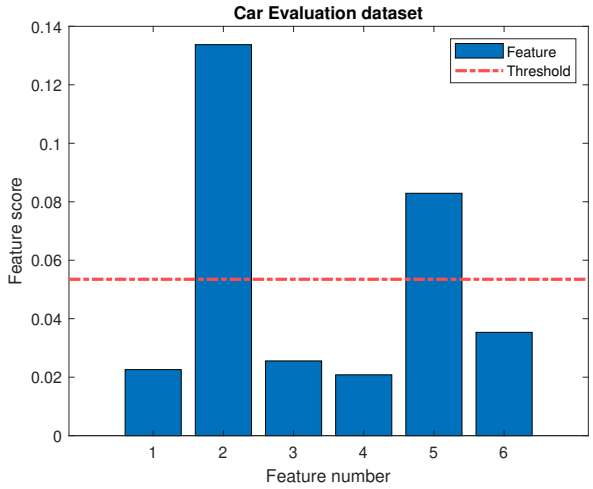


Fig. 3: Feature selection for the car evaluation data set using mRMR.

The mRMR will require, thus, a more complex ANN topology (with higher training times), since a higher number of inputs will be considered. This is an important test case, in order to compare if an increase in the complexity of the model corresponds to an increase in the accuracy.

A summary of the selected features of both methods is presented in Table II. This features will be used next as predictors for the ANNs trained with LMA and PSO, in order to assess which optimisation algorithm and feature selection process are more suitable for these data sets.

### *B. Optimisation and Feature selection*

The optimisation algorithms' performance is dependent upon the network architecture which is ultimately selected by the number of features feeding to the ANN. Thus, it is

TABLE II: Summary of the features selected by mRMR and Chi-square tests.

| Data Set | Features selected by the mRMR | Features selected by the Chi-square |
|---|---|---|
| Parkinson's | (1) MDVP:Fhi(Hz);<br>**(2) MDVP:Jitter(Abs)**;<br>(3) Jitter:DDP;<br>(4) MDVP:Shimmer(dB);<br>(5) Shimmer:APQ5;<br>(6) HNR;<br>**(7) PPE**. | (1) MDVP:Fo(Hz);<br>**(2) MDVP:Jitter(Abs)**;<br>(3) MDVP:Shimmer;<br>(4) RPDE;<br>(5) DFA;<br>(6) spread1;<br>**(7) PPE**. |
| Car evaluation | (1) maint;<br>**(2) lug_boot**. | (1) doors;<br>**(2) lug_boot**. |
| Red wine | **(1) volatile acidity**;<br>(2) free sulfur dioxide;<br>(3) density;<br>**(4) pH**. | **(1) volatile acidity**;<br>(2) residual sugar;<br>(3) chlorides;<br>**(4) pH**. |
| White wine | (1) citric acid;<br>**(2) residual sugar**;<br>(3) chlorides;<br>(4) free sulfur dioxide;<br>(5) sulphates. | (1) fixed acidity;<br>**(2) residual sugar**;<br>(3) total sulfur dioxide. |

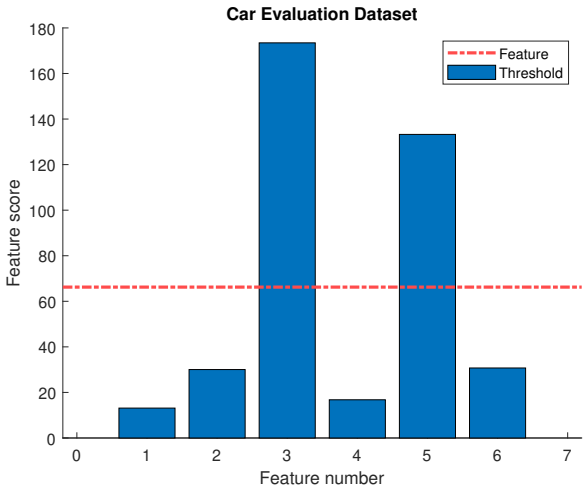


Fig. 4: Feature selection for the car evaluation data set using Chi-square tests.

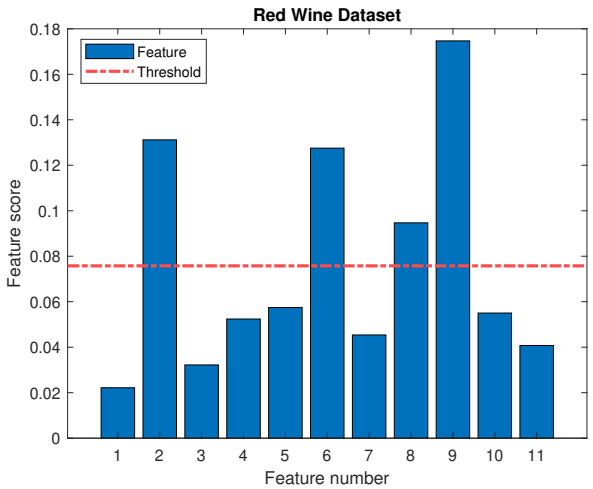


Fig. 5: Feature selection for the red wine quality data set using mRMR.

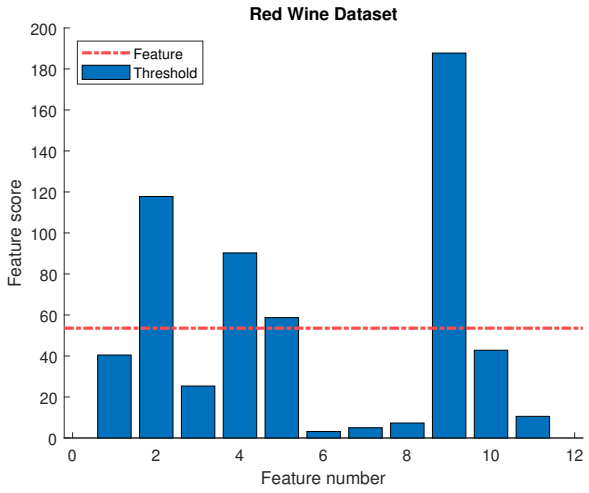


Fig. 6: Feature selection for the red wine quality data set using Chi-square tests.

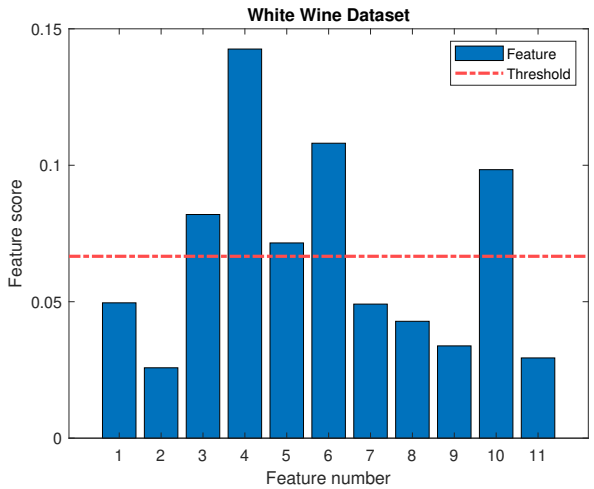


Fig. 7: Feature selection for the white wine quality data set using mRMR.

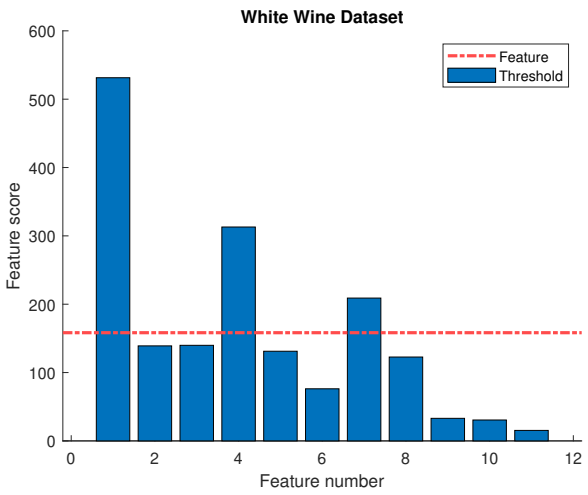


Fig. 8: Feature selection for the white wine quality data set using Chi-square tests.

essential to select the appropriate feature selection method and optimisation for better training.

Keeping this view, this section deals with the comparison of the performance of feature selection-optimisation algorithm pair, in order to select the best combination for better performance. For the purpose of this work, the accuracy was given by the confusion matrix of the test set. As the study includes two feature selection methods and two optimisation algorithms, there will be four combinations, as can be depicted in Table III.

It is noteworthy that the discrete response variable of each data set was encoded using a one-hot encoder, i.e., the output of each data set was transformed into a set of binary columns for each category. Besides that, and since the predictors of the car data set were also categorical variables, the features were transformed to numerical values using ordinal encoding.

For the Parkinson's data set, the best accuracy was obtained using the PSO with the features selected by the Chi-square method. However, it is possible to perceive that this combination required the highest number of hidden units. This may call into question if the difference between the accuracy achieved using a higher number and the accuracy achieved using a lower number of hidden units may or not compensate a longer and more complex training. As an example, LMA–Chi2 combination, achieved an approximate accuracy but needed less than half of the hidden units used in the combination PSO–Chi2. Moreover, The Chi-square-based feature selection process seemed to be superior to mRMR in all cases, in terms of test accuracy.

A similar scenario was found in the car data set, where, on average, the best accuracy is obtained using the combination PSO–Chi2. Notwithstanding, it is also the combination with the highest standard deviation, meaning that in some cases, this combination may not be stable between executions, in terms of optimising weights. In this view, the mRMR revealed to be a more stable algorithm when compared with the Chi-square. On the other hand, in this data set, the accuracy seems to be unrelated to the training algorithms, since, for the two training algorithms, the mRMR and Chi-square obtained individually similar results. Although the best accuracy was found by the combinations LMA–Chi2 and PSO–Chi2, the choice for the best combination should fall on the PSO algorithm, since it required a lower number of hidden neurons.

Taking into consideration the red wine data set, the best combination in terms of maximum and average accuracy is PSO–Chi2; though, the Chi-square-based feature selection method required more hidden units when compared with the mRMR. Nevertheless, all combinations were not able to predict the wine classification with high accuracy; this situation can depict that neither the two optimisation algorithms nor the two feature selection processes are good approaches for this data set, or even that the relationship between independent variables and the output variable is very complex to model.

The last data set is the data set that contains assessments of the quality of red wines. This data set is an interesting test since the mRMR selected 5 features and the Chi-square only 3. Thus, the model created with the features selected by the mRMR will be more complex, aiming at an increase in the precision of the created models. However, as can be seen by Table III, this increase in complexity is not much more beneficial when compared to the simpler models created by Chi-square, taking into account that on average all models had 45% of accuracy. Nonetheless, the combination that found the maximum accuracy was PSO–Chi2, that in this case required the lowest number of hidden units, but still revealed not to be stable between executions.

It is important to note that in some cases, e.g., LMA–Chi2, PSO–Chi2 for red wine data set, PSO–Chi2 for white wine data set, the best and worst accuracy was achieved using the same number of hidden neuron units, i.e., 15, 20 and 4, respectively, which depicts the randomness of the optimisation algorithms in terms of converging to optimised weights.

Making a general assessment only in terms of the optimisation algorithms, the PSO appeared to more the most suitable algorithm for these databases, especially if we considered the maximum accuracy; however, at the expense of longer training time, since it requires a higher number of hidden units when compared with the LMA, that performs best when fewer hidden units are used. In terms of stability, both algorithms showed similar behaviour. Red and white wine data sets have achieved the lowest accuracy among all data sets ranging from 35% to 62%. This may be due to a relatively higher number of classes than other data sets. Further, the authors of the original paper [1] of the wine data set have also claimed differences in the training phases of ANNs when compared to support vector machines (SVM). This is due to difference of cost function used for both algorithms, since SVM penalises the large errors linearly whereas ANN tries to minimise the sum of squared errors. Since SVM is less sensitive to outliers, it results in higher accuracy, unlike ANNs.

Receiver Operating Characteristics (ROC) curves depict the prediction ability of the classification system. It is a graphical

TABLE III: Performance of feature selection–optimisation algorithm combinations.

| **Algorithm** | | **Parkinson's (%)** | **Car (%)** | **Wine (Red) (%)** | **Wine (White) (%)** |
|---|---|---|---|---|---|
| LMA–mRMR | No. hidden (max/min) | 7/12 | 4/20 | 7/20 | 7/20 |
| | Max | 86.2069 | 68.7259 | 58.5774 | 49.1826 |
| | Min | 27,5862 | 63.3205 | 35.1464 | 41,2807 |
| | Mean ± standard deviation | $56.8966 \pm 29.3103$ | $66.0232 \pm 2.7023$ | $46.8619 \pm 11.7155$ | $45.2316 \pm 3.9509$ |
| LMA–Chi2 | No. hidden (max/min) | 4/7 | 10/7 | 15/15 | 4/20 |
| | Max | 89.6552 | 74.1313 | 61.0879 | 51.49864 |
| | Min | 10.3448 | 67.9537 | 43.5146 | 40.0545 |
| | Mean ± standard deviation | $50 \pm 39.6552$ | $71.0425 \pm 3.0888$ | $52.3013 + /8.7866$ | $45.7766 \pm 5.7228$ |
| PSO–mRMR | No. hidden (max/min) | 12/15 | 12/4 | 12/4 | 20/10 |
| | Max | 86.2069 | 69.1120 | 55.6485 | 51.0899 |
| | Min | 62.0690 | 64.0927 | 39.3305 | 42.5068 |
| | Mean ± standard deviation | $74.1379 \pm 12.0690$ | $66.6023 \pm 2.5097$ | $47.4895 \pm 8.1590$ | $46.7984 \pm 4.2915$ |
| PSO–Chi2 | No. hidden (max/min) | 12/20 | 7/12 | 20/20 | 4/4 |
| | Max | **93.1035** | **74.9035** | **62.3431** | **54.6322** |
| | Min | 58.6207 | 66.4093 | 47.6987 | 35.9673 |
| | Mean ± standard deviation | $75.8621 \pm 17.2414$ | $70.6564 \pm 4.2471$ | $55.0209 \pm 7.3222$ | $45.2997 \pm 9.3324$ |

tool that visually represents the relationship between sensitivity/True Positive rate and (1-specificity)/False Positive rate, which, in turn, tests the ability of classification system under different decision-making thresholds for data classification. ROC along with Area under curve (AUC) gives a good insight about the prediction system.

The wine data set has achieved the worst accuracy for all feature selection-optimisation algorithms combinations, unlike other data sets. So, it is vital to understand the reason behind achieving the worst accuracy using other performance measures (ROC-AUC analysis). The ROC curves for the best ANN architectures w.r.t white and red wine data set are presented in Fig. 9 and 10, respectively.

For red wine data set, the model was not able to classify class 1 and 2 which is due to relatively less number of samples (10 and 53, respectively). Due to less number of samples, the model could not learn to classify class 1 and class 2. Moreover, the AUC for the class 1 and class 2 are very less than other classes. For white wine data set, the number of classes is 1 more than red wine. Class 1, 2 and 6 were poorly classified due to limited samples 20, 53 and 18, respectively. The AUC for these classes are far more less than other classes with an adequate number of samples for model training. Due to classes imbalance, the white and red wine data set did not perform well in terms of accuracy and ROC curves (using specificity and sensitivity values).

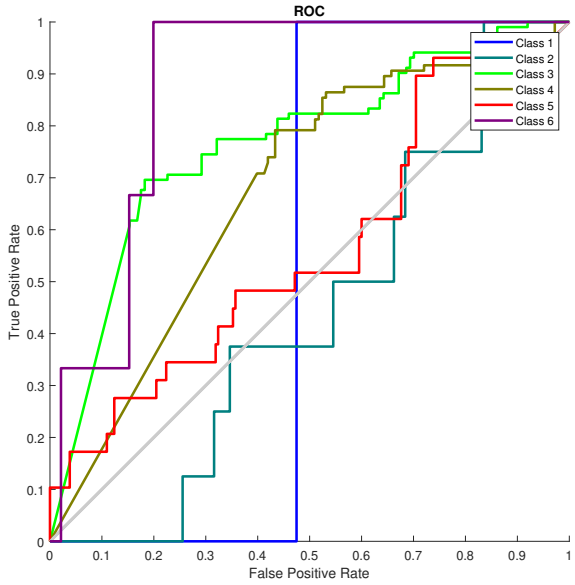


Fig. 9: ROC curves of the best ANN behaviour for the red wine data set.

## IV. Conclusion

This study highlighted the use of ANN for classification tasks. Four data sets were used and subject to two different feature selection processes: Minimum Redundancy and Maximum Relevance (mRMR) and Chi-square tests. Besides that, the experiments were conducted also using different optimisation algorithms, using different network architecture. The two optimisation algorithms considered were the Levenberg-Maquardt algorithm and Particle Swarm Optimisation.

The accuracy of an ANN model is dependent upon the network architecture, training algorithm and the feature selection process used. These three parameters should be chosen (most of the times by a trial and error approach) for better training and better accuracy. In this work, we presented a comparative analysis of feature selection and optimisation algorithms for classification tasks, in order to obtain the combination of approaches that leverage the accuracy of a given model.

Making a general assessment, the best combination that, on average, found more accurate results were the use of PSO with the features selected by the Chi-square process, although it required most of the times a higher number of hidden units. Moreover, the concept of feature selection and optimisation algorithm combination have not worked in case of mRMR since both for LMA and PSO in combination with mRMR produces similar values of accuracy. On the other hand, both optimisation algorithms had similar stability problems. Furthermore, in some cases, the same neural network architecture

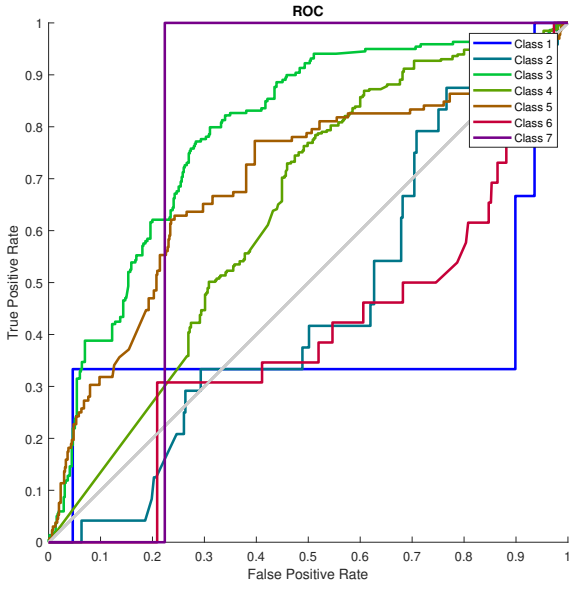


Fig. 10: ROC curves of the best ANN behaviour for the white wine data set.

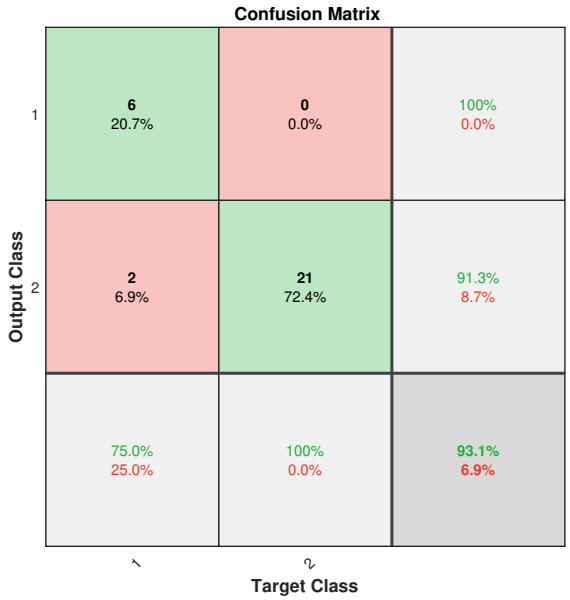


Fig. 11: Confusion matrix of the best ANN behaviour for the Parkinson's data set.

gave best and the worst results which depict the dependence of optimisation algorithms to selection of parameter, e.g., $\lambda$ in LMA and $\omega$ in PSO. In Appendix A, the confusion matrices of the best and worst ANN behaviour are presented.

Finally, it is important to note that for the white wine data set, the mRMR selected more features than the Chi-square. This situation created more complex models to be trained and did not result in a significant increase in the accuracy. Overall, both wine data sets did not produce significant results due to a large number of classes and imbalance between the number of samples of each class.

The code-base of this paper will be available in the following GitHub repository: https://github.com/Dntfreitas/PSORNN_Classification

## Appendix A
## Confusion Matrices of the Best and Worst ANN Behaviour

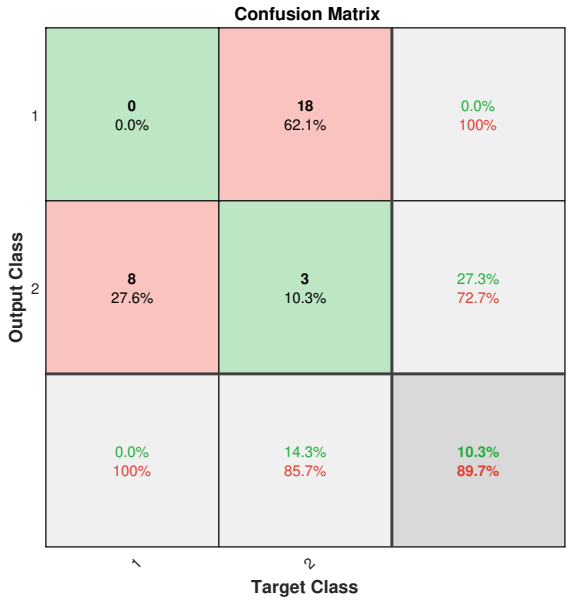


Fig. 12: Confusion matrix of the worst ANN behaviour for the Parkinson's data set.

## References

[1] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[2] Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *J. Bioinf. and Comput. Biol.*, 3(2):185–205, Apr., 2005, doi: 10.1142/S0219720005001004.

[3] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proc. of the 6th International Symposium on Micro Machine and Human Science (MHS)*, pages 39–43, Nagoya, Japan, Oct. 1995.

[4] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. of the International Conference on Neural Networks (ICNN)*, volume 4, pages 1942–1948, Perth, Australia, Nov. 1995.

[5] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

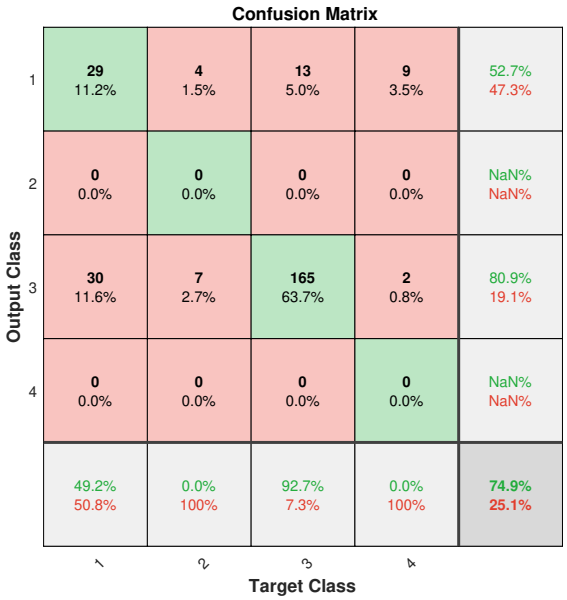


Fig. 13: Confusion matrix of the best ANN behaviour for the car data set.

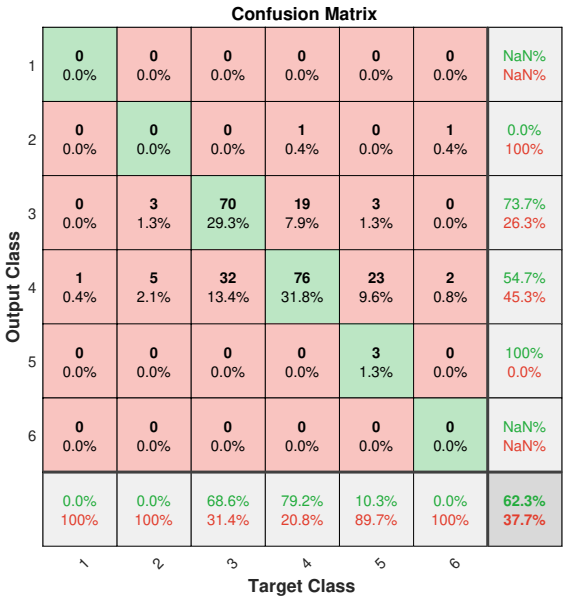


Fig. 15: Confusion matrix of the best ANN behaviour for the red wine data set.

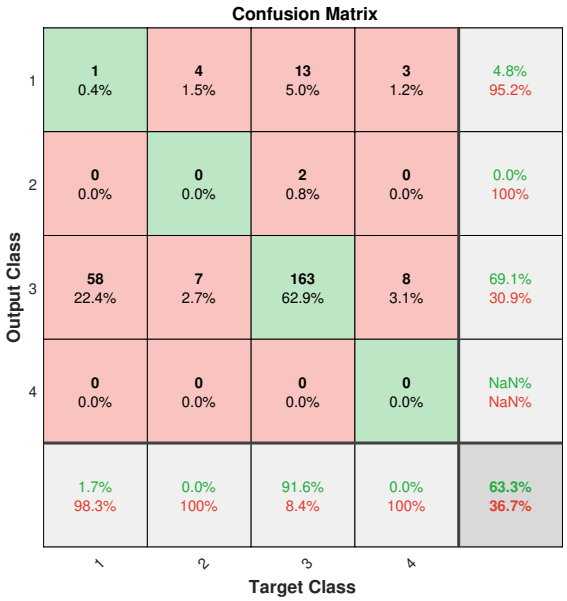


Fig. 14: Confusion matrix of the worst ANN behaviour for the car data set.

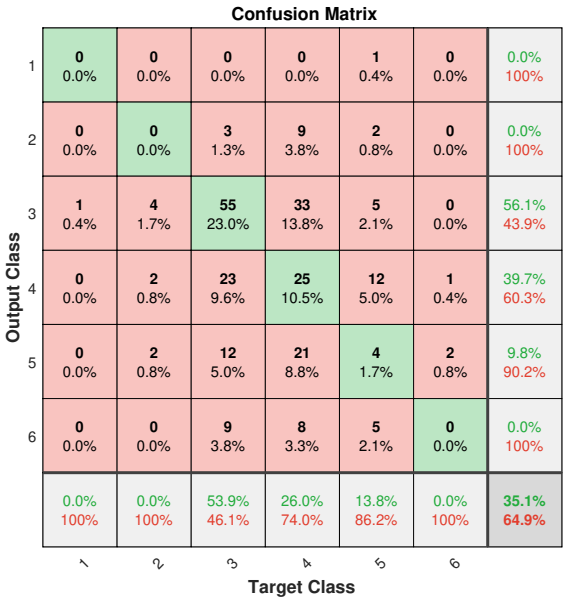


Fig. 16: Confusion matrix of the worst ANN behaviour for the red wine data set.

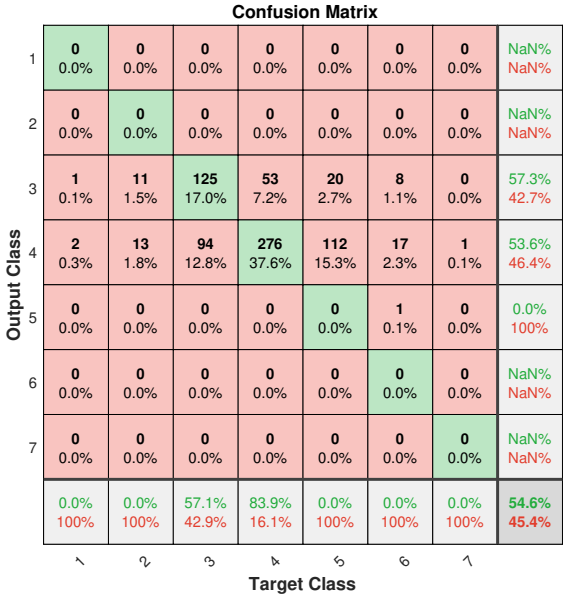


Fig. 17: Confusion matrix of the best ANN behaviour for the white wine data set.

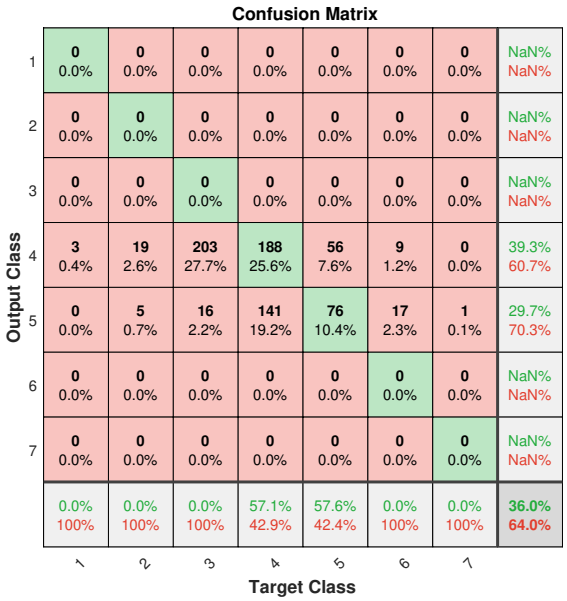


Fig. 18: Confusion matrix of the worst ANN behaviour for the white wine data set.


[6] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[7] Y. Shi and R. C. Eberhart. A modified particle swarm optimizer. In *Proc. of the IEEE World Congress on Computational Intelligence (WCCI)*, pages 69–73, Anchorage, AK, USA, May 1998.