Chapter 4

Pre-Processing from the Data Set

Databases are typically not made to support analysis with a automatically learning algorithm, this is why pre-processing of data is necessary.
The provided Data is labelled and saved in .csv format, that is Comma Separated Values document. The twenty attributes and the four different measures of Blood Pressure are distributed in columns, and each instance (we will call the rows as instances) is a learning sample or pattern.

Pre-processing relevant techniques are:

- **Data Cleaning:** remove inconsitences from the data
- **Feature Engineering:** find the right features/attribute set
  - Feature Transformation: bring attributes into a suitable form (e.g., normalization, discretization)
  - Feature Construction: apply mathematical transformations to the features, construct derived features
  - Feature Subset Selection: select appropiate feature subsets
- **Sampling**
  - select appropriate subsets of the data

All this tasks were carried out, and the results reported in this chapter.

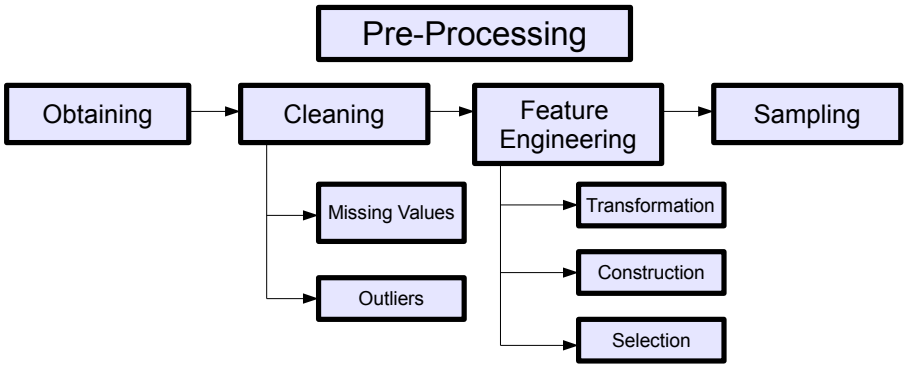

Figure 4.1: Pre-Processing Tasks

4.1 Data Cleaning

It was noticed that the Dataset will probably contain many inconsistent values as empty values (designed as '0'), negative values, and also outliers, which must be removed because they would cause a bad performance of the ANN.
The main purpose of this phase, according to [28], is the manipulation and transformation of the raw data, therefore the information contained in the set of data may be discovered, or easily accessible.
First the Data was represented in Histograms by means of Weka Software. The obtained Histograms for four different attributes can be observed in Figure 4.2. It was noticed the presence of outliers which affect negatively to the limitation of the input feature intervals.

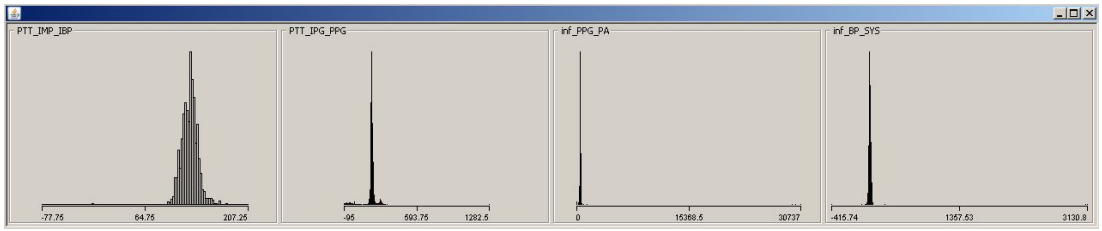

Figure 4.2: Histograms of four Features

Some filters included in Weka, could be applied to obtain a Dataset suitable for the feature selection phase. Within this set of filters, the most relevant in this research were

the next:

- weka.unsupervised.attribute.ReplaceMissingValues. It replaces all missing values for nominal and numeric attributes in a dataset with the modes and means from the training data.
- weka.unsupervised.attribute.Discretize. It discretizes a range of numeric attributes in the dataset into nominal attributes. Discretization is by simple binning. Skips the class attribute if set.
- weka.unsupervised.instance.Randomize. Randomly shuffles the order of instances passed through it. The random number generator is reset with the seed value whenever a new set of instances is passed in.
- weka.unsupervised.instances.Resample. Produces a random subsample of a dataset using sampling with replacement. The original dataset must fit entirely in memory. The number of instances in the generated dataset may be specified.

To proceed solving the problem of achieve a better representation of the Data, it would be contrasted some possibilities. First of all, due to its facility is to delete the instances with outliers or empty values, the next one, would be to replace empty values and outliers for means of the attributes, or even replace all the instances, that do not provide good information. In some of this cases, must be considered the lost of information, but in spite of everything, it will be better than having bad instances.

Applying the filter 4.1, the empty values were substituted for the mean values of each feature. But the suspected problem after applying this filter was, that the supposition that the empty values should be these mean values could probably not be true, so it was being introduced false information. The other possibility was then to remove the lines with missing values or outliers, this could be also a bad solution, because of lost of information, but after this solution was tested with the Dataset, good results were obtained.

It was designed a little algorithm (Appendix C) in *Pearl Programming Language (postscript)*. It was applied to eliminate the incomplete or incorrect instances, what after the first Histograms were observed it is the same as removed the instances which contains negative and empty values.
After applying this code, and introducing the resulting Data in Weka, it was obteined the Histograms in figure 4.3, were it can be observed a better interpretation of the

provided Data.

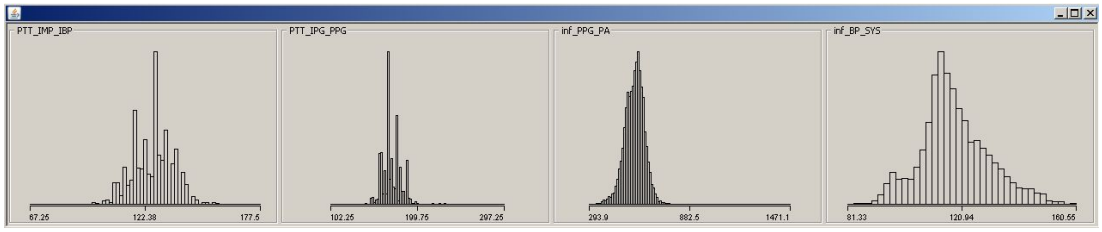

Figure 4.3: Histograms of four Features

4.2 Feature Engineering

4.2.1 Feature Transformation

Many datasets, as in the presented situation, contain continues domain attributes, so it could be indispensable the previous application of any method to reduce the possible set of output values, obtaining a finite number of output intervals. This can be carry out with the filter called weka.unsupervised.attribute.Discretize, which will divide the output in a different number of bins depending on the selected output feature. It is known that a big amount of learning algorithms operate with discrete classes. In this research this solution was not contempled, but it could be the starting point for future researchs.

The normalization of Data was also considered. This proceedment is mentioned as necessary in [16] and [26], and [29] is proved that with normalized data, it is possible to obtain better results constructing ANNs. The normalization, was carried out with a function from matlab called *mapminmax*, which returns the normalized Data in the interval $[-1, +1]$, and also relevant information, for denormalizing after simulations.

4.2.2 Feature Construction

It is demostrated in the Thesis [29] that a mathematical transformation to the data like logarithmus or n^{th} power, could improved the results. This idea was also developed, providing then, a bigger number of input features.
They were applied four different transformations: $x^2, x^3, \sqrt{x}, log(x)$. So at this time, there were available ninety-five input features.

It could be also a good idea, the construction of some groups of features, $x_1 - x_2, x_1 + x_2, x_1 \cdot x_2$..., but they were not created in this research, because the set of features was already too big.
At last, there is an algorithm, which will be introduced, in the next sections. It is Principal Component Analysis, with this method, it can be transformed an n-dimensional space into a lower-dimensional subspace, without losing much information. After carrying out a ranking of the attributes, also with Principal Component Analysis, it was created some features subsets construction with this algorithm and the results simulated using Weka. This was reported in the next section.

4.2.3 Features Subset Selection

The selection of relevant features, and the elimination of irrelevant ones, is a central problem in machine learning [20]. Practical machine learning algorithms are known to degrade in performance (prediction accuracy) when faced with many features that are not necessary for predicting the desired output. These features can be irrelevant, uninsteresting or redundant. So it can be conclude that are needed some techniques to recognise unnecessary attributes.

Removing these attributes could: increase efficiency, improve accuracy and prevent overfitting. The Feature Subset Selection techniques try to determine appropiate features automatically, that maximizes accuracy of machine algorithms.

The problem of feature subset selection consist of finding a subset of the original features of a dataset, such that a machine learing algorithm that is run on data containing only these features, generates a classifier that minimizes the trade-off in accuracy complexity.

The first thing made in this section, was to pay attention to the different possibilities offerted for carrying out the selection. There are two main groups of reduction mechanisms:

- Transformation from the Dataset to a lower dimension space.
  - This is the reduction of a set of variables into a smaller set while keeping most of the information content. To pursue this Non Supervised techniques for dimension reduction, there are mainly two methods, namely, principal component analysis and factor analysis.

- Obtaining of the attributes, subgroups more adequate for the prediction.
  - This is the reduction of candidate independent variables with respect to a dependent variable. The Supervised techniques available are: Filters and Wrappers. It should be noted that these feature selection methods choose a set of features from existing features, and does not construct new ones; there is no feature extraction or construction (like there is with PCA).

In this report, it has been tried techniques from both groups. First of all, it was applied PCA. And then it has also been used a reduction from attributes using Wrappers. The PCA and the Wrappers were applied using *Weka*, and the results reported in Sections 4.2.3.1 and 4.2.3.2 respectively.

4.2.3.1 Non-Supervised Techniques: Principal Component Analysis

The central idea of Principal Component Analysis (PCA) is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. This is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables, and next components contain less and so on [17].
PCA is a widely used multivariate analytical statistical technique that can be applied to reduce the set of dependent variables to a smaller set of underlying variables (called components) based on patterns of correlation among the original variables.
With the Pulse Transit Time attributes and the other parameters in the Dataset, several dependent variables may be correlated with one another. PCA identifies patterns

of correlation among dependent variables and substitutes a new variable (component), for the group of original attributes that were correlated.

Due to the fact that it is a non-supervised technique, now it will not be neccesary to take account the output features, but only the inputs. It is also irrelevant which of the five different constructions ($x^2, x^3, \sqrt{x}, log(x)$) is introduced into the analysis, due to the fact that all the sets own the same information, it does not matter if they are constructed sets or not.

Now the algebraic properties of this PCA will be defined. x_k represents a vector with all the instances for an input feature, then the PCA starts computing the covariance matrix (Σ) for the k vectors ($k = 1, 2, \ldots, p$). From this convariance matrix, the eigenvectors and eigenvalues will be computed, which fulfill the next equation:

$$\Sigma A = A\Lambda \tag{4.1}$$

being Λ a diagonal matrix whose k^{th} diagonal element is λ_k, the k_{th} eigenvalue of Σ, where A is the orthogonal matrix whose k^{th} column, α_k, is the k^{th} eigenvector of Σ.
Now the Principal Components will be defined as the linear transformation resulting in Equation 4.2, where the variance of each principal component, (z_k), will be λ_k. So after they are found they will be ordered attending to their variance values, as it was before discussed.

$$\mathbf{z} = A^T\mathbf{x} \tag{4.2}$$

The kth principal component can be computed as follows:

$$z_k = \alpha_k^T x = \alpha_{11}x_1 + \alpha_{12}x_2 + \ldots + \alpha_{1p}x_p = \sum_{j=1}^{p} \alpha_{1j}x_j \tag{4.3}$$

In Appendix A the Tables A.1,A.2 and A.3 represent the cumulative variance vector, the eigenvectors and the resulting principal components respectively, for one of the simulations carried out with four input features and 1000 patterns.
The orthonormal linear transformation of x, which defines z, has some optimal properties, which can be found in [17].

An important parameter in this analysis is the quantity of information, that is desired. This is measured by the cumulative variance, which is the sum of the variances of the different principal components, form the first one (with the bigger variance), to the last component that will be in the output subset. The normal selections are between

80% and 90%. This parameter will establish the number of principal components that will be in the output of the analysis, according to the cumulative variance vector.

After introducing the set of data in Weka, it was applied the Principal Component Analysis algorithm from the Select Attributes tab within the explorer menu. Were made two different simulations, because it might be possible that the output be a linear transformation of the input features (*"transformBackToOriginal"* set to false) or a subset of the features (*"transformBackToOriginal"* set to true). In the first case, the output of the PC Analysis was a set of components formed by means of linear combinations of the correlated attributes. And in the second analysis, the output was a subset of the input set of parameters, which offer the majority of the information.

Index	**Feature**	**Index**	**Feature**
1	PTT_QRS_IBP	11	nic_IMP_Area
2	PTT_IPG_PPG	12	PTT_QRS_IMP_f
3	PTT_QRS_IPG	13	nic_IPG_Area
4	PTT_QRS_IMP	14	nic_IPG_PA
5	PTT_IMP_IBP	15	inf_HR
6	PTT_IBP_PPG	16	PTT_QRS_PPG
7	PTT_IPG_IBP	17	nic_HR
8	PTT_IMP_IPG	18	inf_PPG_Area
9	PTT_IMP_PPG	19	inf_PPG_PA
10	nic_IMP_PA		

Table 4.1: PCA Ranking of the Features

The next steps were: first of all it was obtained a ranking of all the features which can be observed in Table 4.1, providing which of them are more representative in the Dataset. With this information, the set was divided in six subsets of $4, 6, 8, 10, 12, 14$ attributes, which were simulated direct with Weka and with the next configuration of Multilayer Perceptron in the control panel of Weka:

- Learning Algorithm: Backpropagation with Momentum
- Training Set: 60% (600 patterns)
- Validation Set: 40% (400 patterns)
- Hidden Layers: 4, 6, 8, 10, 12

- Learning Rate: 0.2
- Momentum Constant: 0.3
- Normalized Input Data: True
- Normalized Output Data: True
- Training Time: 2000 Epochs
- Validation Threshold: 20

The results of the simulations with **x** input patterns are reported in Table 4.2. The simulations of the other data sets are in Appendix A.

	8 Hidden Layers		**10 Hidden layers**	
Input Features	MAE *(mmHg)*	STD *(mmHg)*	MAE *(mmHg)*	STD *(mmHg)*
4	3,5570	4,3356	3,6436	4,4809
6	3,5794	4,4614	3,6025	4,4753
8	3,6082	4,6126	3,5570	4,5960
10	3,6104	4,5360	3,5650	4,6557
12	3,2681	4,2929	3,1075	4,1651
14	3,3247	4,9947	3,3266	4,7792
$\mathbf{10}_b$	3,1646	4,2377	3,0830	4,1396
$\mathbf{11}_b$	3,3608	4,2640	3,0735	4,0869
$\mathbf{12}_b$	3,3284	4,5888	3,2531	4,3354
$\mathbf{13}_b$	3,2461	4,4898	3,3245	4,3657

Table 4.2: Weka Multilayer Perceptron Simulations with x input ranked patterns.

It is noticed in the results of Table 4.2, and also in the others from Appendix A, that there is no decrease of error, by adding from eight to ten parameters. This could have happenned, because the added features, do not introduce more information, or in any other case, they are damaging to the Dataset.
After this was noticed, the idea was to create four new subsets without adding one of both two parameters (PTT_IMP_PPG and nic_IMP_PA), which are not contributing any improvement:

- 10_b.csv. 10 features from ranking without taken both.
- 11_b.csv. 11 features from ranking without taken *PTT_IMP_PPG*.
- 12_b.csv. 12 features from ranking without taken both.
- 13_b.csv). 13 features from ranking without taken *PTT_IMP_PPG*.

The results for these subsets are shown in the same tables.

Now was the time to make the same simulations, but this time with the linear transformations from Principal Components Analysis. And it will be tested if these new sets of attributes could provide any improvement. Then again more PC Analysis, one for each subset of features, choosing always to obtain 90% of the variance. From each set were obtained a number of components (Table 4.3), to cover this at least 90% of the variance. These components as it was already mentioned belong to linear transformations applied to the input data.

PCA_Input_Data	**Output Components**	**Variance Covered**
PCA_ranked_4.csv	3	0,9328
PCA_ranked_6.csv	4	1,0000
PCA_ranked_8.csv	4	1,0000
PCA_ranked_10.csv	4	0,9212
PCA_ranked_12.csv	5	0,9362
PCA_ranked_14.csv	6	0,9389

Table 4.3: PC Analysis for Ranked Data

The simulations of these new data sets were carried out with the same configuration as before, but this time only for 8 and 10 hidden layers, due to the next. It has been already researched ([25]) that the configuration of the Network itself is not so important as the selection of the input set of parameters. And it will be demonstrated in the next chapters, that it is absolutely true. Because it was noted much more improvement in the results, varying the input feature selection algorithm, as varying learning algorithm, number of hidden layers, training time,...
Also in this moment it is not so interesting to find the best configuration, as finding the

best input subset of features. A more exhaustive research, of the possible configurations, will be carried out in the next chapter.

The simulations were carried out for estimation of the SBP. The results of the simulations for **x** input patterns are shown in Table 4.4, and for the other transformations of the data set are presented in Appendix A.

	8 Hidden Neurones		10 Hidden Neurones	
PCA Inputs	MAE ($mmHg$)	STD($mmHg$)	MAE($mmHg$)	STD($mmHg$)
4	5,1950	6,8847	5,1627	6,7131
6	3,5993	4,6406	3,5441	4,6698
8	3,6856	5,7974	3,6552	5,2805
10	3,6417	5,0999	3,7479	5,7884
12	3,7391	4,9108	3,4541	4,9220
14	3,3606	4,4311	3,3308	4,5715

Table 4.4: Weka Simulations with PCA inputs.

All these simulations were made to observe which of the subsets responds better to the simulations. Now the selection of the input subset will be made attending not only to the Mean Absolut Error, and the Standard Deviation of the estimations, but also to which is the best with the less number of parameters. It will be also better to find a good subset with the lowest number of features possible, because it will mean also a lower computational cost in their obtention. This comparation will be carried out in Section 4.2.4

4.2.3.2 Supervised Techniques: Wrappers

As it was already mentioned, the supervised feature subsets selection techniques use information about the learning task (e.g.:look at the relation of an attribute to class attribute).

The idea behind the Wrapper approach, shown in Figure 4.4, is simple: the induction algorithm is considered as a black box with tunable parameters (in this research an ANN, with a particular learning algorithm). The induction algorithm is run on the

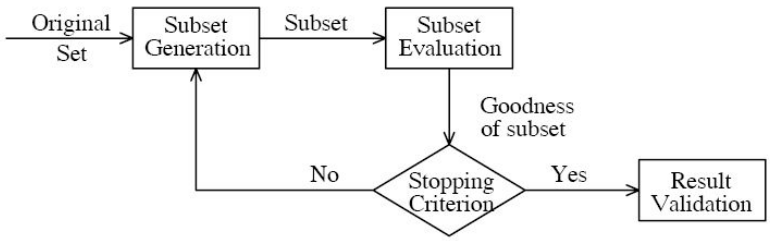

Figure 4.4: Four key steps of supervised feature subset selection

output subset, usually partitioned into internal training and test sets.
There are two components into the wrapper approach: a search component and a evaluation component. The search component repeteadly suggests subsets of the data set. The evaluation component evaluates these subset by running the induction algorithm several times and getting an estimate of our objective function, usually accuracy. The subset with the highest estimated value will be chosen as the final subset on which to run the induction algorithm.

The goal of this approach is to improve accuracy, and whenever there are two models with approximately the same accuracy, it is prefered the simper one (e.g., less features used). For accuracy estimation, it were used repeated runs of five-fold cross-validation.
They test subsets of features until find the most suitable (computationally less expensive). For the evaluation, they use a learning algorithm. In this case was used Backpropagation with Momentum. It is not possible to do an exhaustive search, due to the long time that the analysis would take. Therefore different search methods can be used.

The search method used in this research is called *Best First*, it is a *Backtracking algorithm* which is a method to find the best solution for a computational problem without checking all the possibilities. It works covering a graph (Figure 4.5), where each node represent a solution for the problem. So the algorithm has success when travelling in the graph, it finds a solution which solve the problem. In this case the stopping criterion is the number of consecutive non-improving nodes allowed. It will be "8" in all the simulations. There are three variants of this search algorithm attending to how evolve the search:

- **Forward Selection**. Start with the empty set and search forward.
- **Backward Elimination**. Start with the full set of attributes and search backward.
- **Bidirectional Search**. Start at any point (random subset), and search in both directions.

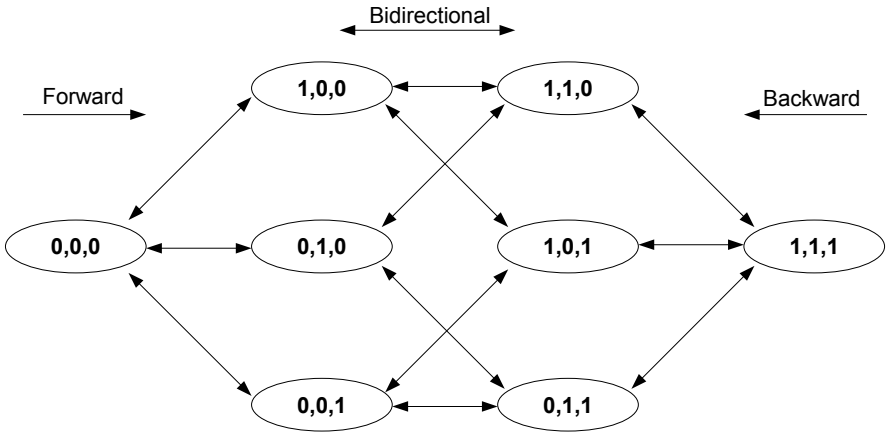

Figure 4.5: Backtracking Search Method. For a set of 3 features, where "1" means selected and "0" not selected.

For simplicity to express the results, all the features will be numbered as in the Table 4.5, and the constructions as a letter next to the number so x construction will be "a", x^2 will be "b", x^3 will be "c", and $\sqrt{x}, logx$ will be "d" and "e".

Then were carried out the wrapper simulations for SBP estimations, six for each construction varying the number of hidden layers in the induction box, and the search strategie. The parameters of the learning algorithm were the same for all the simulations:

- **Evaluator**
  - Learning Algorithm: Backpropagation with Momentum
  - Evaluate in all training data.
  - Hidden Layers: 6, 8, 10

Index	Feature	Index	Feature
1	PTT_QRS_IBP	11	PTT_QRS_IMP_f
2	PTT_IPG_PPG	12	nic_HR
3	PTT_QRS_IPG	13	inf_HR
4	PTT_QRS_IMP	14	nic_IPG_PA
5	PTT_IMP_IBP	15	nic_IPG_Area
6	PTT_IBP_PPG	16	nic_IMP_PA
7	PTT_IPG_IBP	17	nic_IMP_Area
8	PTT_IMP_IPG	18	inf_PPG_PA
9	PTT_IMP_PPG	19	inf_PPG_Area
10	PTT_QRS_PPG		

Table 4.5: Wrappers Numeration

  - Learning Rate: 0.2
  - Momentum Constant: 0.1
  - Normalized Input Data: True
  - Normalized Output Data: True
  - Training Time: 500 Epochs
  - Validation Threshold: 20

- **Search Method**
  - Best Search
  - Start set: No Attributes
  - Search Direction: Forward, Backward
  - Number of Node expansions: 8

The results for all the constructions are in Appendix A, here are only presented in Table 4.6 the results for the x construction. In these tables are shown also the results of the simulations with the subsets resulting on the wrapper analysis.

Although it can be observed that the results are good enough, it may be interesting to build another set, but this time with features from all the data constructions. Were selected parameters from the sets with the best results in the simulations, and was built

Search Direction		6 Hidden Layers	8 Hidden Layers	10 Hidden Layers
Forward	Subset	1a, 2a, 3a, 4a, 5a, 6a, 7a, 8a, 10a, 11a, 12a, 13a, 16a, 17a, 18a, 19a.	1a, 2a, 3a, 4a, 5a, 6a, 7a, 8a, 10a, 11a, 12a, 13a, 16a, 17a, 18a, 19a.	5a, 7a, 8a, 10a, 11a, 12a, 13a, 18a, 19a.
	MAE	2,9297	2,6248	2,5673
	STD	3,7526	4,0208	3,6116
Backward	Subset	1a, 2a, 3a, 4a, 5a, 6a, 8a, 9a, 10a, 11a, 12a, 13a, 14a, 15a, 16a, 18a, 19a.	2a, 3a, 4a, 7a, 8a, 10a, 11a, 12a, 13a, 17a, 18a, 19a.	4a, 9a, 11a, 12a, 13a, 17a, 18a, 19a.
	MAE	2,5581	2,5616	2,5355
	STD	3,6421	3,7390	3,5622

Table 4.6: Subset from Wrapper analysis for x input patterns.

another set, this time with many redundant features. Then it was introduced again the set into the wrapper analysis, to reduce the number of attributes. The analysis was made with the next parameters:

- **Evaluator**
  - Learning Algorithm: Backpropagation with Momentum
  - Evaluate using Cross Validation 5-Folds.
  - Hidden Layers: 10
  - Learning Rate: 0.2
  - Momentum Constant: 0.1
  - Normalized Input Data: True
  - Normalized Output Data: True
  - Training Time: 1000 Epochs
  - Validation Threshold: 20
- **Search Method**
  - Best Search
  - Start set: Random
  - Search Direction: Bidirectional
  - Number of Node expansions: 8

The resuls obtained in Weka after this analysis were presented in a percentage, which informs of which parameters were selected in what percentage of the 5-Folds. In Table 4.7 are presented the results, and the final feature subset selected.

After simulate the chosen subset in the last wrapper analysis with the parameters of the estimator, varying only the training strategie to validation split 60% and the number of epochs to 2000, were obtained $MAE(error) = 2,2788$ and $STD(error) = 3,2108$, which are much better than the previous results, with only one feature construction data. It is also interesting to mention, that although the selected subset has 13 input features (more than the PCA subset selected) only 6 from the 13 are neccesary to measure, because the rest are mathematical constructions, so the subset is a better estimator, and also need less computational cost.

Feature	Folds (%)	Decision	Feature	Folds (%)	Decision
1b	20	no	13a	80	yes
3b	0	no	13b	60	no
3c	60	no	13c	80	yes
4a	60	no	13d	100	yes
9a	40	no	13e	40	no
10b	20	no	17d	20	no
10c	40	no	18a	80	yes
10e	100	yes	18b	80	yes
11a	80	yes	18c	80	yes
11c	60	no	18d	0	no
11d	20	no	18e	80	yes
11e	100	yes	19a	40	no
12a	100	yes	19b	0	no
12b	80	yes	19c	20	no
12d	20	no	19d	80	yes
12e	60	no	19e	60	no

Table 4.7: Wrapper analysis with all the constructions.

4.2.4 Conclusions in Subset Selection procedures

Observing the table for the ranked input patterns, the best results were for 10_b`.csv` with no construction, in other words, with x input features.
After comparing the results of ranked input data and with principal components input data. It is observed that the data with principal components transformation, do not provide any improvement. Hence this construction will not be taken into account to the next phase.
From the wrapper analysis, as it was already mentioned, will be taken the resulting subset from the last simulation. Which is expected to be much better than PCA ranked subset. This set will be called `wrap.csv`.
The estimation results could be understood as random, due to the fact that the ANN intialize their weights and bias before any new simulation in a random way. So it is impossible, for an ANN to obtain the same estimation results in two simulations. Despite that, the selected subsets were assumed to be the best.

4.3 Sampling

The next phases of the research will need different sets of patterns for to test them, and to prove the generalization capability of the Neural Network to estimate the SBP. So they will be sampled different comma separated values files from each subset of features. All these files will be sampled using the filter `weka.unsupervised.instances.Resample` from Weka simulator. The results will be 8 files, for each subset of features, with 100, 200, 400, 800, 1000, 2000, 4000 and 8000 patterns. They will be called `#wrap.csv` and 10_b`#.csv`, for the wrapper subset and the principal component analysis ranked data respectively.