CS-BIGS 4(2): 116-133
http://legacy.bentley.edu/csbigs/documents/kainulainen.pdf

Ensembles of Local Linear Models for Bankruptcy Analysis and Prediction

Laura Kainulainen
Aalto University, Finland

Yoan Miche
Insitut National Polytechnique de Grenoble, France

Emil Eirola
Aalto University, Finland

Qi Yu
Aalto University, Finland

Benoît Frénay
Université Catholique de Louvain, Belgium

Eric Séverin
University Lille 1, France

Amaury Lendasse
Aalto University, Finland, IKERBASQUE and University of the Basque Country, Spain

Bankruptcy prediction is an extensively researched topic. Ensemble methodology has been applied to it in past work. However, the interpretability of the results, so often important in practical applications, has not been emphasized. This paper builds ensembles of locally linear models using a forward variable selection technique. The method applied to four datasets provides information about the importance of the variables, thus offering interpretation possibilities.

1. Introduction

Bankruptcy prediction has gained increasing attention since the 1960s (Altman, 1968), and not without reason. Predicting the financial distress of firms benefits the company leaders by identifying internal problems, but also assists auditors in their work for finding potentially troubled firms. Above all, bankruptcy prediction produces information for investors and banks so that they can make sounder lending and investing decisions (Wilson & Sharda, 1994; Atiya, 2001). At present, the applied methods range from well-known statistical methods to

advanced soft computing techniques (Kumar & Ravi, 2007). Nevertheless, predicting the probability that a firm will fail is not sufficient, because it does not reveal the causes behind the event. This paper proposes to use a technique called ensemble of locally linear models combined with forward variable selection. It is able to assess the importance of the variables, thus providing more interpretability than "black box" models.

A bankrupt firm is one that is unable to meet its financial obligations (Beaver, 1966). In other words, it cannot pay back its debtors. Consequently, the firm enters a juridical process called bankruptcy. The legal definition makes it possible to define the starting point of bankruptcy accurately, which is important in research. However, the precise definition varies from country to country. Even then, most legal systems recognize two phases of corporate bankruptcy: reorganization and liquidation. Typically, reorganization can be considered as a second chance for the firm while liquidation stands for sale or cessation of the company (Pochet, 2002). Bankruptcy prediction research often aims at binary classification of the firms, yet predicting default probabilities would be more beneficial in many ways, although more difficult (Atiya, 2001). This paper adopts the binary classification viewpoint.

In everyday life, if we have the option of asking for a second and a third opinion -- why not do that? The intuitive idea behind ensemble modeling is to use the wisdom of crowds. Systems of multiple experts have bloomed in research (Kuncheva & Whitaker, 2003), because properly combining several diverse and independent classifiers into one classification output gives better results than a lone classifier. This also holds in bankruptcy prediction (Verikas et. al., 2010). In this paper, local linear models are built on multiple regions of the dataset using multiple combinations of variables. The aim is not only to create good ensembles and predict equally well or better than in previous research, but to provide interpretable results. The interpretability further benefits the users of bankruptcy prediction. Ensembles of locally linear models have not been applied intensively to bankruptcy prediction yet, although the idea of local linear models that are based on the Euclidean distance of the K nearest neighbors is relatively old (Cover & Hart, 1967). Furthermore, using ensemble techniques for choosing the parameters of the model (e.g. size of the local neighborhoods), is a novel approach. The proposed methodology is presented in Section 3.

The paper begins with Section 2 that shortly reviews ensemble modeling in general and in particular in bankruptcy prediction. It is followed by a presentation of the method used in this paper, Ensembles of Locally Linear (E-LL) models combined with forward search, in Section 3. The E-LL is compared to other methods, Linear Discriminant Analysis and Support Vector Machines with a random kernel, as presented in Section 4. Finally, test results are presented in Section 5, and their implications discussed in Section 6.

2. Ensembles in Bankruptcy Prediction

Bankruptcy prediction research has bloomed since the originating works in the late sixties (Beaver, 1966; Altman, 1968). Altman (1968) used multivariate discriminant analysis, and a decade later Olhson (1980) applied a logistic regression approach, both of them subsequently gaining popularity in practice and in academic studies (Atiya, 2001). Later, various techniques – which some authors claim are nearly all intelligent ones – have been applied to the problem (Kumar & Ravi, 2007). The techniques employed range from statistical techniques to soft computing approaches, passing by neural networks, case-based reasoning techniques, decision trees, evolutionary approaches and rough sets (Kumar & Ravi, 2007).

Moreover, Verikas et al (2010) have presented a review on hybrid and ensemble- based techniques applied to bankruptcy prediction. Without comparing the hybrid and ensemble-based techniques amongst themselves, the authors claim that properly designed hybrid or ensemble-based techniques outperform systems based on one predictor. In their opinion, a successful ensemble design requires a trade-off between the ensemble accuracy and the diversity of the ensemble members. They view genetic algorithms as prominent means to integrate feature selection, selection of the hyper-parameters, and training of the ensemble members. However, genetic algorithms might be computationally very time- consuming when used with large feature sets. Amongst the many issues discussed by these authors are the transparency of results in order to analyze the reasons behind the bankruptcy and the possibility of increasing prediction accuracy by including non-financial features.

Although assessing the performance of different methods is difficult, there is a consensus that intelligent approaches, such as neural networks, decision trees or ensembles, outperform stand-alone statistical techniques, like linear discriminant analysis or logistic regression (Kumar & Ravi, 2007). Besides, stand-alone techniques can always be added to the ensembles. However, many authors claim that their technique outperforms the previous ones, but a fair comparison of the results is challenging due to the different datasets used and the

imprecise presentation of the results (Verikas et. al., 2010). As a conclusion, it has been stated that intelligent techniques outperform traditional statistical ones and properly designed ensembles outperform stand-alone classifiers.

2.1.Ensemble terminology

Classifier ensembles, also called multiple classifier systems and consensus theory, or a special case of mixtures of experts (Polikar, 2006; Jacobs et. al., 1991), have attracted a lot of interest among the research community (Kuncheva & Whitaker, 2003; Rokach, 2010). The main idea is to create several classifiers and further combine them into one model. The approach is analogous to the wisdom of crowds: a combined opinion of many experts, which have independent and diverse opinions and whose opinions can be properly combined, is usually more accurate than the opinions of single individuals. Reviewing the extensive literature on ensemble modelling is out of the scope of this paper: only the most relevant parts are discussed. For tutorials and surveys already published on the topic, see e.g. (Polikar, 2007; Polikar, 2006; Kuncheva, 2004; Rokach, 2010).

The process of creating an ensemble of classifiers consists of two key components: (1) the diversity of individual classifiers and (2) a method for combining the classifiers obtained. The following paragraph presents the notation used in this paper, and the following sections discuss the two key components in ensemble creation. The ensembles of locally linear classifiers are presented in section 3.

Solving classification problems aims at assigning each object with a class label ω_k from the set of class labels denoted as $\Omega = \{\omega_1, ..., \omega_c\}$. In bankruptcy prediction, the classification problem is binary (size of Ω is 2): the company is classified either healthy or unhealthy (in a state of bankruptcy). The dataset on which the classifiers are built consists of entries $\mathbf{x}_j$ that have measured properties, features from the feature space R^n. Since this is an application of supervised learning, the dataset also includes the class label y_i for each of the entries. As a result, the dataset can be described as $\mathbf{X} = \{\mathbf{x}_i, y_i\} \in R^n \times \Omega$. In this context, a classifier is a function that maps an entry point to one of the classes, $M : R^n \to \Omega$ (Kuncheva, 2004). Each of the classifiers label the given data point to one of the possible classes in Ω. In ensembles, several classifier outputs are combined together. The output corresponding to a classifier or model M^m is denoted as $\hat{\mathbf{y}}_m$. The output of these combined models, yielding the ensemble classification, is denoted as $\hat{\mathbf{y}}$. Figure 1 illustrates an ensemble of classifiers. The classifiers or models $M^1, ..., M^m$ are created based on the dataset $\mathbf{X}$, each of the classifiers providing an output $\hat{\mathbf{y}}_m$ and combined into the final output $\hat{\mathbf{y}}$.

2.2.Importance of the diversity of ensemble members

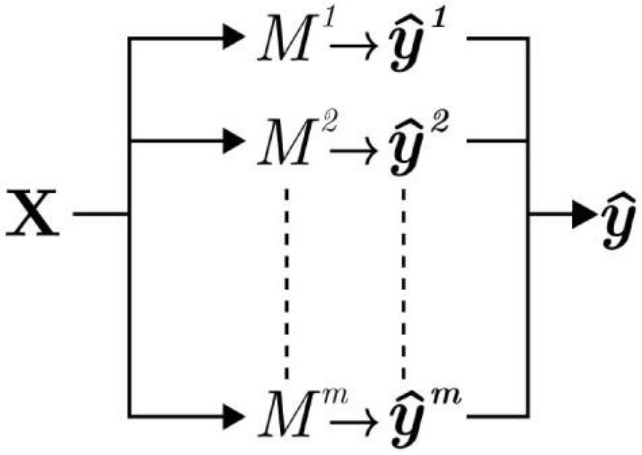

Figure 1. Ensemble process

According to various authors, diverse individual classifiers make errors on different instances. In other words, the decision boundaries of classifiers should be different (Polikar, 2006; Rokach, 2010). The variety is supposed to improve classification accuracy. Although the need for dissimilarity seems to be intuitive, the classification framework lacks a solid and commonly accepted theory of the benefits of classifier diversity (Brown et. al., 2005; Kuncheva & Whitaker, 2003). Diversity creation can be obtained in several ways, and the approaches to classify them vary (Brown et. al., 2005; Rokach, 2010). In the following paragraphs, two ways are considered.

Brown's et al. (2005) concept of explicit and implicit diversity methods is an interesting approach. As their names suggest, explicit methods explicitly ensure the diversity of the classifiers while implicit methods do not take into account the information of classification diversity.

Rokach (2010) classifies the diversity generation methods into five categories: manipulation of the training sample, manipulation of the inducer (e.g., the way the classifier is formed from the data set), changing the target attribute representation, partitioning the search space and hybridization techniques. (i) Bagging (a.k.a. bootstrap aggregating) of Breiman, is one of the earliest

representatives of the methods manipulating the training samples. It obtains diversity by using randomly drawn training data subsets, each of them used to train a different classifier (Polikar, 2006; Breiman, 1996). (ii) Manipulating the inducer could mean for example changing the number of neurons in a single layer feed-forward neural network (Rokach, 2010), whereas (iii) Changing the target attribute could be predicting an "easier" task before targeting the actual one. (iv) Partitioning the search space is based on the idea of dividing the instance space into several sub-spaces. For example, the feature subset-based methods are a popular approach. In those methods, each classifier is given a feature subset. These subsets can be created in various ways, including random-based, reduct-based and collective-performance-based strategies (Rokach, 2010). It is proposed that feature subset-based ensembles might help avoiding the pitfalls of feature selection techniques, such as choosing the same features for all the classifiers, especially with high dimensional data (Tumer & Oza, 2003). (v) An example of hybridization technique is to use several types of inducers or classifiers together (Rokach, 2010).

2.3.Importance of merging the opinions of experts

The second aspect of creating ensembles of classifiers is the method for merging the classifiers. These methods can be classified in several ways. For example, they can be partitioned to classifier fusion and selection techniques, or to trainable and non-trainable practices, or to weighting methods and meta-learning methods. The choice of the method also depends on the type of the output. If the output consists of class labels, methods such as majority voting or weighted majority voting might be appropriate. For continuous outputs, many kinds of algebraic combiners, such as weighted average, or decision templates can be used (Kuncheva, 2004; Polikar, 2006; Rokach, 2010. It must be noted, though, that continuous outputs can be converted to labeled output simply by using an appropriate threshold, which is however not obvious to choose.

Techniques to combine continuous outputs are numerous, including among others algebraic combiners, decision templates and application of the Dempster-Shafer Based Combination (Polikar, 2006). The Non-negative least-squares combiner used in this paper is closest to algebraic combiners, which are presented briefly in the following paragraphs.

Algebraic combiners contain, among others, the mean rule, weighted average, median rule, product rule, and generalized mean. In general, they are non-trainable combiners. Different combiners are actually functions that combine the outputs of individual classifiers $(\hat{\mathbf{y}}_1,...,\hat{\mathbf{y}}_m)$ into one general output $\hat{\mathbf{y}}$. As a comparison, decision templates compare the decision profile of each test point to the decision templates of each class and chooses the most similar (Kuncheva, 2004; Polikar, 2006).

For example, a trimmed mean classifiers removes the most pessimistic and optimistic classifiers before calculating the mean. Weighted average combination methods assign the classifiers a weight before averaging the results. There are several ways to obtain these weights (Kuncheva, 2004; Polikar, 2006). In a way, some algebraic combiners, such as mean rule, rely more on the power of the crowd. At the same time, some give more importance to the classifiers which perform better than the others. In such approaches, the performance of individual classifiers when shown new data must be estimated.

3. Ensembles of Locally Linear Models

The method presented in this paper is an ensemble of locally linear models. Multiple locally linear classifiers are created and further combined into one ensemble. Even though the base classifiers are locally linear, the global model is non-linear.

3.1.Locally Linear Models

According to Bontempi et. al. (2001), global models and local models differ in two aspects. Global models consider all the operating conditions of the modeled system, which is coherent if the system is assumed to be driven by a physical-like law. Also, the problem of input-output mapping is treated as a function estimation question. In contrast, the local models relax one or both of these aspects (Bontempi et. al., 2001). The ensemble of locally linear models (E-LL) is not modeling a global classification border, but classifying each data point of the test set based on a model built on its nearest neighbours in the training set. As a result it adopts the idea of memory-based learning (Cover & Hart, 1967), where the training data is not discarded but used for classification in the test phase.

The idea of using locally linear classifiers that are based on K nearest neighbors can be found in the work of Bottou and Vapnik (Bottou & Vapnik, 1992). In this case, the number of neighbors used (K) was fixed. Later, the same principle was used for regression, but the configuration of the model was chosen with leave-one-out cross-validation using the PRESS statistic (Bontempit

et al., 2001). These concepts are essential also in the method presented in this paper.

In the k-fold cross-validation technique, the data set are divided into k blocks, and each of the blocks is of size N/k, where N is the total number of observations. Each block is used in turn as a calibration set and the remaining k-1 blocks as a training set. The leave-one-out method is a special case of k-fold cross-validation, where the training set consists of all observations except one, which is used for validation. It means that k is equal to N (Polikar, 2007). In ensembles of locally linear models, the leave-one-out cross-validation contributes to building a more accurate ensemble, since the models that are estimated to perform the best with new data are favored in the ensemble formation. This also reduces the risk of over-fitting.

However, computing the leave-one-out output might be time-consuming. The PRESS statistic allows to exactly calculate the LOO error very efficiently. The main idea is to divide the normal error by a correction term and thus obtain the leave-one-out error. Formula 1 is used to calculate the ε_i^{PRESS} error, which is the leave-one-out error for sample *i*,

$$\varepsilon_i^{PRESS} = \frac{y_i - \mathbf{x}_i \mathbf{b}_i}{1 - \mathbf{x}_i \mathbf{P} \mathbf{x}_i^T} \quad (1)$$

where $\mathbf{P}$ is defined as $\mathbf{P} = (\mathbf{X}^T\mathbf{X})^{-1}$ and $\mathbf{X}$ is the matrix containing the data points in a linear system $\mathbf{Xb} = \mathbf{y}$. For a detailed explanation of the method, see (Myers, 1990; Bontempi et. al., 1998[2]; Miche et. al., 2008).

A locally linear model is a regression model that is performed for each observation in the dataset, as a linear combination of its nearest neighbors (Bontempi et. al., 1998[1]). The idea is that locally enough, all problems can be considered linear. The original KNN algorithm uses the K nearest neighbors of an observation to define its class. The observation is labeled with the class which dominates among these neighbors. In this method, the distance between two samples is defined as the Euclidean distance (Kuncheva, 2004). However, instead of the pure KNN algorithm, here the locally linear regression predicts the class label of each observation. The nearest neighbors are used only as a basis to build the regression model. The number K of neighbors used has to be at least the number d of dimensions plus one, because otherwise linear regression cannot be performed (Miche et. al., 2010). In this paper, the lower bound of K was d+2 and the maximum number of K is d+18 due to computational time constraints. These boundaries were chosen to make possible the computation of a linear regression model while still keeping the computational time reasonable.

3.2.Diversity creation and combination methods

As other ensembles, ensembles of locally linear models also consist of two key components. First, the diversity of individual classifiers is created from two different sources. Both different features and different numbers of neighbors are used in the local linear classifier. Some models use both different variables and multiple K as the bases for the linear regression models. The second aspect is to combine the models, which is done with a non-negative least-squares algorithm. Both aspects are discussed in the following sections.

3.2.1.Multiple values of K in the K Nearest Neighbors method create diversity

As discussed in Section 3.1, the K Nearest Neighbors method is used as a basis for the linear regression model. Depending on the dataset, problem, and variables, the optimal number K of neighbors changes as well as the resulting classification output. Thus, the first source for diversity comes from multiple numbers of neighbors. Figure 2 illustrates a situation where multiple values of K, $d + 2$, $d + 3$ . . . $d + limit$, are used to create multiple models $M^1, M^2, ..., M^m$. In this paper, the limit equals 18, in order to maintain a reasonable computational time while still producing enough models for the creation of the linear regression model.

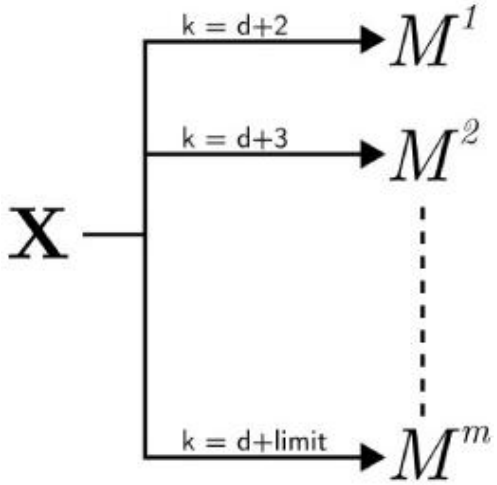

Figure 2. Creating diversity with multiple values of K.

3.2.2.Multiple sets of variables create diversity

In data analysis, variable selection is an important but a

complicated task, which has been addressed in several ways. The properties of ensembles can be utilized to solve the problem. One way to obtain diverse classifiers is to build them on different variable sets. The combination method of the ensemble emphasizes the ones performing the best. Since the nearest neighbor method is based on Euclidean distance, changing the variables also changes the distances and the neighbors. Figure 3 explains this principle. Multiple variable sets *1, 2, … , n* are used to create multiple models $M^1, M^2, ..., M^n$.

In total, *d* variables result in 2^d-1 possible subsets of the variables. The dimension of the datasets used is at smallest 28, which results in approximately $2^{28}-1$ subsets. It is already too much for an exhaustive search. That is why a strategy for selecting the variable sets used in Figure 3 has to be developed.

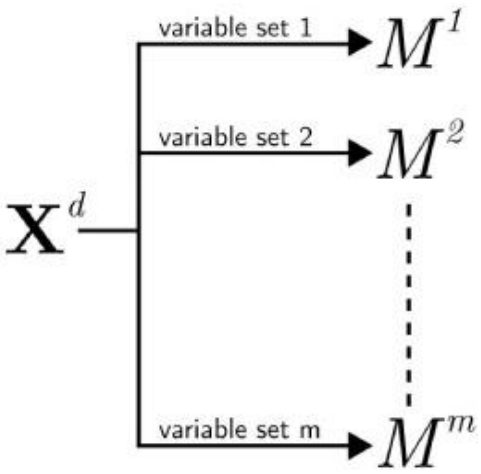

Figure 3. Creating diversity with multiple combinations of variables.

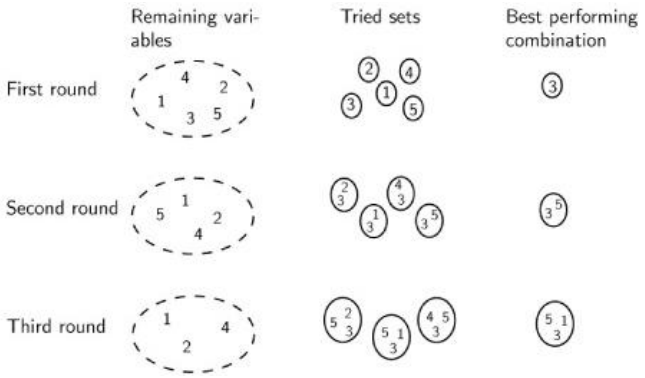

Figure 4. Forward selection: the sets of variables selected from the previous round are used as a basis for the following round.

3.2.3 Forward search

How are these variable sets selected? There are numerous possibilities for variable selection, such as random subspaces or forward search (Polikar, 2006). Forward search enables to keep the number of the variables rather small, which improves interpretation possibilities. In forward search, the models are initialized on all possible variables one by one. The variable triggering the best performing model is selected. Second, the rest of the variables are combined one by one with the variable that was selected from the first round. The best combination is saved. On every round, one more variable is added to the combination, until no further improvements can be found or the process is stopped (Rossi et. al., 2006).

Figure 4 illustrates the forward method. One of the five variables is selected and all the variables one by one are added to the set and tested. The set that obtains the highest accuracy (percentage of correct classification) is selected as the basis of the next round.

3.2.4 Multiple values of K and multiple combinations of variables create diversity

The previous sections present how to obtain diversity of the classifiers from two different sources: multiple values of K in the K nearest neighbor method and multiple combinations of variables. This section presents how to use both of the sources in diversity creation.

The process is divided into four steps that are executed in series. The starting point is a subset of the original data set. It is obtained from the forward search. On each round of the forward search, a new variable is added to the set of the selected variables. For example, at this stage the starting point set might contain variables 3, 5 and 1. One by one, each of the remaining variables is added to this set in order to assess its performance. The set to be evaluated can be for example {1, 2, 3, 5}. Figure 5 starts from this point. It is also the middle column in Figure 4, "Tried sets", each of the sets being a "starting point" of Figure 5. In the first phase, that variable set is split into all possible subsets, except the empty set (in this case $P(\{1,2,3,5\}) \setminus \varnothing = \{\{1\},\{2\},\{3\},\{1,2\},\{1,3\},...\}$, where P is the notation for a power set). In Figure 5 these sets are represented as $X^{set\,i}$. Each of these subsets is used as a basis for an ensemble. In the end, there will be in total 2^d - 1 ensembles, *d* being the dimension of the set to be evaluated. In Figure 5 their outputs are represented as $\hat{\mathbf{y}}^i_{\mathrm{LOO}}$, and $m = 2^d - 1$. In the example set of 4 variables, there will be $2^4 - 1 = 15$ ensembles. But how are these ensembles built?

Phases two and three in Figure 5 consist of building the ensembles that are based on the subsets obtained in phase 1. The classifiers of these ensembles are models that are built on multiple values of K, as presented in Section 3.2.1. In Figure 5, this is presented in phase 2. The values of K vary between $d + 2$ and $d + limit$, d being the dimension and *limit* being 18 in the experiments presented here. It means that one ensemble of this phase consists of *limit* - 1 models. The models are linear regression models, but since their bases vary, they vary as well. The output of these models is the leave-one-out-output (see 3.1), which helps avoid over-fitting. These models are merged into an ensemble (see Figure 5, phase 3). As noted in the previous paragraph, in the end there are $2^d - 1$ ensembles that are combined into the final output. In Figure 5, this is represented by the final phase, number 4. The ensemble creation procedure presented in Figure 5 is repeated for all the variables added to the set from the previous round, similarly to the example of the set {*1,2,3,5*} at the starting point.

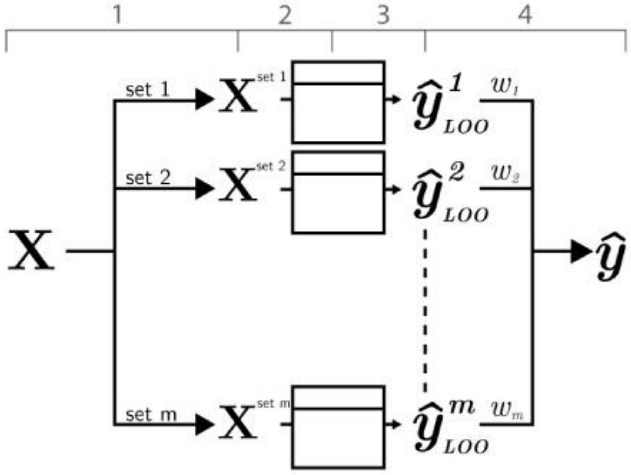

Figure 5. Two ensembles in a series with diversity from two sources: K Nearest Neighbors and variable sets.

3.2.5.Combining multiple models into an ensemble

There still remains the question of combining the diverse classifiers obtained in the previous sections. As seen in section 2.3, there are various possibilities to merge the classifiers. In this paper, the classifiers are combined with a weighting system. Furthermore, the prediction performance of the ensemble is assessed with a leave-one-out method applied to a linear system. In the training phase, the actual outputs are known, since the training of the classifiers is supervised. The process explained in the following paragraphs applies for the training phase. When the model is used for testing, the weights obtained in training are used. As seen in Section 3.2.4, this ensemble method actually consists of two layers of ensembles in series. The combination process is the same in general terms for both of the layers (phases 3 and 4 in Figure 5).

The weights for the classifiers that form the ensemble are obtained from the non-negative least-squares (NNLS) problem that is solved between the classifier outputs and the actual output. According to Miche et al. (2010), the advantage of NNLS is that it is efficient and fast. The square of the difference between the actual output and the weighted leave-one-outputs of the classifiers is minimized under the constraint that the weights are positive, as seen in Equation 2.

$$\min_{w_j} \left\| y - \sum_j y^j_{LOO} w_j \right\|^2_L, \text{s.t. } w_j \geq 0. \tag{2}$$

In reducing over-fitting, the leave-one-output and the positive weights play an important role. Leave-one-output estimates the performance of the model when each of the data points is used as validation set at time (see Subsection 3.1). The constraint that the weights be positive reduces over-fitting, because the model cannot be fitted too tightly to the training data (Miche et. al., 2010). Because there are "two ensembles in series", the leave-one-output is accomplished in two ways that require closer study.

First, in the third phase of the process explained in Section 3.2.4 and seen in Figure 5, several local linear regression models are combined into an ensemble. In this phase, the leave-one-output of the classifiers comes from the fact that the linear regression models to classify each point are built on the point's neighborhood. The point itself is not used in the process. The regression model based on the neighbors is used to predict the class of that particular output. Second, when the weights are calculated with the NNLS algorithm, the classifiers obtaining a zero weight are pruned out. The leave-one-output of the ensemble is calculated with the PRESS statistics (see Section 3.1). In other words, the remaining classifiers and the actual output are treated as a linear system with weights obtained from the NNLS algorithm. Thus, in the first layer of ensembles, the leave-one-output is used in two different ways. In the second layer, as seen in the fourth phase in Figure 5, there are classifiers that are built on different subsets of the subset of variables in question. The output of these models comes from the ensemble output from the previous layer, obtained with the PRESS statistics. A similar process is used to combine these classifiers into the final output, because that output is used in the forward search, and it

is better to choose an ensemble that performs well when shown to new data.

To conclude, the overall process of combining classifiers into ensembles works in the following way:

1. There are classifiers that have to be combined. The output of the classifiers is leave-one-out because (i) either they are local lineal regression models (ii) or they are ensembles themselves and the output was computed with PRESS statistics (see 3).

2. Weights between the classifiers and the actual output are computed. The positive weights minimize the squared difference between the actual output and the weighted combination of the classifiers.

3. The classifiers with zero weights are pruned out. PRESS statistics are used to compute the leave-one-out output of the system between the remaining classifiers and the actual output.

4. The ensemble output is leave-one-out. It is used either as a basis for the second layer of ensembles (see 1) or to assess the performance of the variable set that the two-layer ensemble is based on.

4. Comparison to other methods

In order to evaluate the performance of the proposed method, it has to be compared to other methods. Linear Discriminant Analysis (LDA) was chosen because it has been traditionally used in bankruptcy prediction. Even though the research community has found more accurate prediction methods (see Section 2), LDA is still used in practice. Moreover, Support Vector Machines (SVM) have obtained good results in classification. Although not very widely applied to bankruptcy prediction, they offer a good reference point. The rest of this section explains both methods and their application.

4.1.Linear Discriminant Analysis

In Linear Discriminant Analysis, the main idea is to calculate a score that would describe the risk of a company to go bankrupt. The classification of the scores to bankrupt or healthy is performed according to the chosen threshold. This score is calculated as a linear combination of the explanatory variables. In other words, each variable is given a weight and then summed. The weights are defined to separate the means of the two classes (Fisher, 1936). The whole idea with discriminant analysis is to give more weight to the variables that separate best the means of the two groups and are the most similar within the groups. Altman also tested whether the year when the data were collected has influence on the prediction performance. He concluded that even though the accuracy is lower, a data set collected two years prior to the bankruptcy can be used for the prediction (Altman, 1968).

4.2.Support Vector Machines with an Extreme Learning Machine (ELM) kernel

Support Vector Machines have gained popularity in classification due to the high accuracy they obtain. Sometimes, non-linear problems are laborious to train. A novel approach of combining Extreme Learning Machines with Support Vector Machines has obtained very promising results.

4.2.1.Support Vector Machines

Support vector machines (SVM) were introduced by Boser, Guyon and Vapnik (1992). The book by Vapnik further explain the idea, (Vapnik, 1998) and for various tutorials and books see e.g. (Hearst, 1998; Cristianini & Shawe-Taylor, 2000; Burges, 1998). With much research using and developing them in various application areas, support vector machines can be considered as a state-of-the-art method in classification in terms of accuracy.
The main idea of SVMs is to solve non-linear classification problems in two steps: first the input vectors are mapped into a high-dimensional feature space with a non-linear function. Second, the classification problem is solved in that space by defining a hyperplane that separates the classes. The separating hyperplane aims at maximizing the margin between the training data and the classification boundary. However, not all the data points are used for defining the margin. Only the points closest to the boundary, support vectors, are considered (Boser et. al., 1992; Kumar & Ravi, 2007; Cristianini & Shawe-Taylor, 2000).

The two steps, mapping of input vectors and definition of the hyperplane, can be combined with a kernel function. A kernel function is a function that takes in original data points and outputs their inner product in the high-dimensional feature space. This can be done because of the dual representation of the problem that enables to evaluate the decision rule, using inner products between the test point and the training points. This means that the actual mapping function does not have to been known or computed. There are multiple possibilities for the kernel function, even though they have to fill Mercer's conditions (Cristianini & Shawe-Taylor, 2000; Burges, 1998).

Although popular in other fields, support vector machines have not been used intensively in bankruptcy prediction (Kumar & Ravi, 2007). Min and Lee (2005) compared SVMs with different linear, radial basis function, polynomial and sigmoid kernels, and chose to use the RBF kernel. They concluded that the SVM outperformed multiple discriminant analysis, logistic regression analysis, and a back-propagation neural network. Shin et al. (2005) obtained similar results, also using an RBF kernel. RBF kernels are most frequently used in SVM classification, even though they require the tuning of an additional meta-parameter, which has to be performed simultaneously with the selection of the other meta-parameters.

A new meta-parameter-free kernel has been proposed by Frénay and Verleysen (2010[1]; 2010[2]); it does not suffer from large number of meta-parameters but still maintains the performance of RBF kernels. For that reason it has been used in this paper.

4.2.2.Support vector machines with a random kernel

Tuning the parameters for a kernel might be very time consuming. The new method presented in (Frénay & Verleysen, 2010[2]), combines the SVM classifiers with a random kernel. Consequently, the method combines the Extreme Learning Machine (Huang et. al., 2006) with Support Vector Machines methodologies (Frénay & Verleysen, 2010[2]).

The extreme learning machine (ELM) is an algorithm used for training of single-layer feed-forward networks. It randomly chooses the hidden nodes to be used and optimizes analytically only the output weights. The method is based on the idea that the input weights, that is to say, the selection of the hidden nodes as well as the biases on the hidden layer, if used, can be randomly chosen if the activation function is infinitely differentiable. The single-layer feed-forward network (SLFN) is defined in Formula 3.

$$\sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{t}_j,\ j = 1,...,N \tag{3}$$

In the formula, Ñ denotes the number of the hidden nodes and N the number of the samples. The bias is defined as b_i, the weights as $\mathbf{w}_i$ and the output $\mathbf{t}_j$. The hidden layer output matrix H is defined in Formula 4.

$$\mathbf{H}(\mathbf{w}_1,...,\mathbf{w}_{\tilde{N}},b_1,...,b_{\tilde{N}},\mathbf{x}_1,...,\mathbf{x}_N) = \begin{pmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_N + b_{\tilde{N}}) \end{pmatrix} \tag{4}$$

Each column in **H** represents the output of one hidden node (Huang et. al., 2006). The ELM algorithm consists of three steps in the training phase. At first, the random input weights $\mathbf{w}_j$ and bias b_i are assigned. Second, the hidden layer output matrix **H** is calculated. Third, the output weight β is obtained by solving the linear system between the hidden layer nodes and the output. The meta-parameter-free random kernel combines ELM and SVM. Instead of using a kernel function that requires tuning of an additional meta-parameter, the data points are transferred to a new space similarly to the ELM. The hidden layer of ELM becomes the space where the margin-maximization problem is solved. In other words, a support vector machine using an ELM kernel can be seen as linear SVM in the space of the hidden layer of ELM.

5. Testing the model with four datasets

How well do the methods perform compared to each other? In bankruptcy prediction, obtaining data sets that are publicly available is a problematic task. The datasets are laborious, and above all, very expensive to obtain. There are more data sets for credit scoring analysis, but even though some authors use credit score datasets for testing bankruptcy prediction methods, an analysis on the importance of the variables could not have been performed. Also, the sizes of the data sets are often very limited.

The limited size of the datasets further affects the estimation of the performance of the methods. A good practice is to divide the dataset into training, validation and testing sets. The models are built in the training phase based on the information that the training set contains. The results are validated and the best model selected. Finally, the model is tested in a test set that was not used for building the model. Since the datasets are small, the performance estimation becomes challenging. As a result, Monte-Carlo cross-tests are used. As seen in Section 3, the leave-one-out cross-validation is used with locally linear models, which makes the ensemble creation more accurate: the models that are estimated to perform best on the new data are favored.

Monte-Carlo methods refer to various techniques. The adopted Monte-Carlo cross-test consists of two steps. First, the dataset is divided into training and testing sets. The training set is formed by drawing without replacement a certain number of observations. The

testing set comprises the rest of the observations. However, the proportion of each class is maintained. Second, the model is trained with the training set and then tested with the testing set. These steps are repeated several times (Lendasse et. al., 2003). In this case, the training set contains 75% of the samples and the testing set the rest. These two steps are repeated 200 times due to time limitations. In one dataset, the same partition to training set and test set is used for all the methods, which makes the comparison fair.

The test results are presented in the next Section dataset by dataset. The authors would like to thank Dr. Atiya, Dr du Jardin and Dr. Pietruszkiewicz for their help in providing the four datasets.

5.1.Atiya dataset

The data set developed by Amir Atiya consists of 983 firms; 607 of them were solvent and 376 had defaulted, but the prediction for the defaulted firms was performed at two or three points in time before default. The observations in the defaulted group come from a time period of 1 month to 36 months before bankruptcy, the median time being 13 months (Atiya, 2001). In total, there were 63 variables. The data were standardized to 0 mean and variance 1 before performing the classification task. The variables of the Atiya dataset are presented in Tables 1 and 2. Since the Atiya dataset is unbalanced with regards to the number of healthy and bankrupt companies, a different measure for mean accuracy is used. That measure is defined in Equation 5.

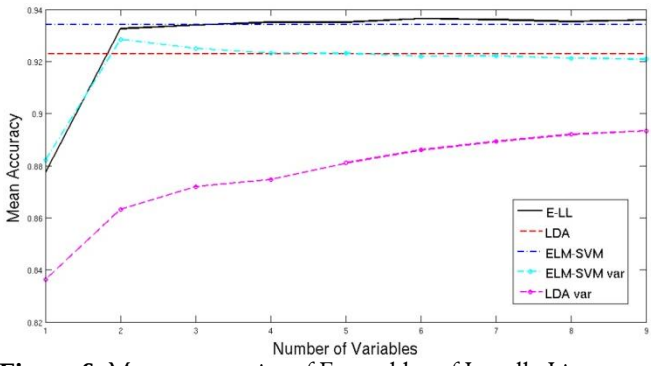

Figure 6. Mean accuracies of Ensembles of Locally Linear models (E-LL in black solid line), Linear Discriminant Analysis (LDA in red dashed line) with same variable selection as with E-LL (marked as LDA var, drawn as magenta dashed line with dots), Extreme Learning Machine Support Vector Machines (ELM-SVM, blue dash-dot line), also with variable selection (ELM-SVM var, cyan dash-dot line with dots). Atiya dataset, Monte-Carlo cross-test repeated 200 times.

$$\frac{\frac{\text{True positive}}{\text{Total positive}} + \frac{\text{True negative}}{\text{Total negative}}}{2} \quad (5)$$

Table 1. Variables used in the Atiya dataset, part 1.

X1	cash/tot assets
X2	working capital/tot assets (TA)
X3	working capital/curr assets
X4	equity (EQ)/TA
X5	1-(long term debt/TA)
X6	rate of chg of cash flow per share (CFPS)
X7	rate of chg (ROC) of earnings per share (EPS)
X8	ROC(EPS from cont. operations)
X9	ROC(gross operating income GOI)
X10	ROC(net oper. Inc NOI)
X11	ROC(sales)
X12	ROC(gross profit margin)
X13	ROC(net profit margin)
X14	a measure of share price chg
X15	a measure of chg of gross oper mgn
X16	one year chg in net profit mgn
X17	ROC(TA)
X18	one year chg in EQ
X19	other ROC(CFPS) (other measure of chg)
X20	other ROC(EPS)
X21	other ROC(EPS cont oper)
X22	other ROC(GOI)
X23	other ROC(NOI)
X24	other ROC(sales)
X25	gross profit mgn
X26	net profit mgn
X27	a measure of dividend incr/decr
X28	cash flow (CF)/TA
X29	earnings/TA
X30	earnings cont oper/TA
X31	GOI/TA
X32	NOI/TA
X33	sales/TA
X34	PE ratio
X35	P/CF ratio
X36	price sales ratio
X37	price book value ratio
X38	return on assets ROA
X39	return on equity
X40	current ratio

Note: ROC=rate of change (usually over 4 year period), CFPS=cashflow per share, EPS=earning per share, GOI=gross operating income (i.e. before taxes, interest and other deductions), profit mgn=profit margin, TA=total assets, gross profit mgn=profit margin as related to GOI, EQ=shareholders equity (also called book value), NOI=net operating income (after taxes, etc), P/CF=price cash-flow ratio, PE = price earnings ratio.

Table 2. Variables used in the Atiya dataset, part 1.

X41	Quick ratio
X42	market capitalization/(long term debt LTD)
X43	relative strength indicator
X44	gross profit mgn
X45	net profit mgn
X46	one-year rel chg of CF
X47	one-year rel chg of GOI
X48	one-year rel chg og NOI
X49	4 yr ROC(CF)
X50	4 yr ROC(GOI)
X51	4 yr ROC(NOI)
X52	3 yr ROC(CF)
X53	3 yr ROC(GOI)
X54	3 yr ROC(NOI)
X55	TA
X56	sector default prob
X57	one year ROC(price)
X58	4 yr ROC(price)
X59	3 yr ROC(price)
X60	price
X61	a measure of ROC(price)
X62	volatility
X63	3 yr ROC(EQ)

Note: ROC=rate of change (usually over 4 year period), CFPS=cashflow per share, EPS=earning per share, GOI=gross operating income (i.e. before taxes, interest and other deductions), profit mgn=profit margin, TA=total assets, gross profit mgn=profit margin as related to GOI, EQ=shareholders equity (also called book value), NOI=net operating income (after taxes, etc), P/CF=price cashflow ratio.

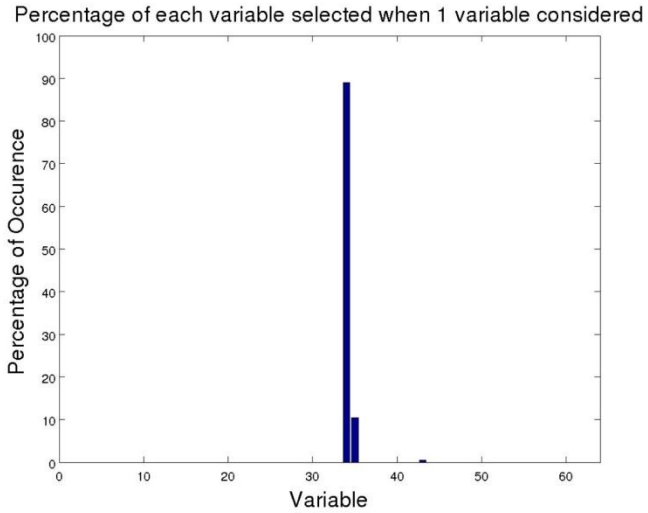

Figure 7. Percentage of the first variable chosen; Atiya dataset

Figure 7 displays the percentage of each variable to be chosen first, and it can be seen that variables 34 and 35 gain the largest importance in predicting the bankruptcy.

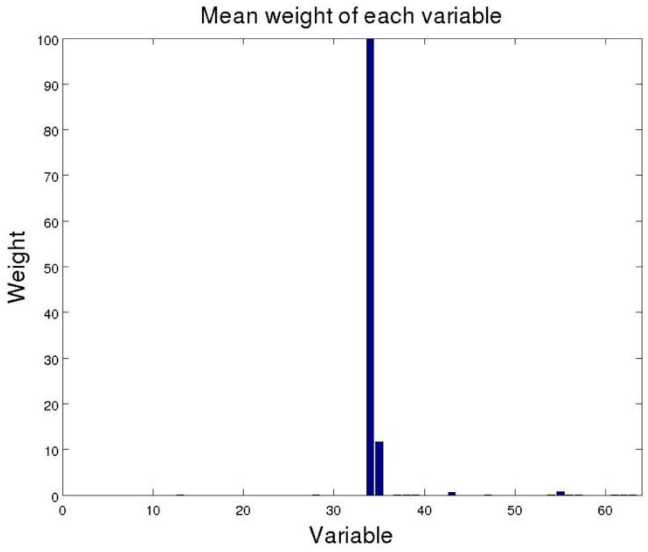

Figure 8. Weight added to each variable. The weight was calculated as a sum of additional test accuracy that the variable brought and scaled to the largest weight; Atiya data.

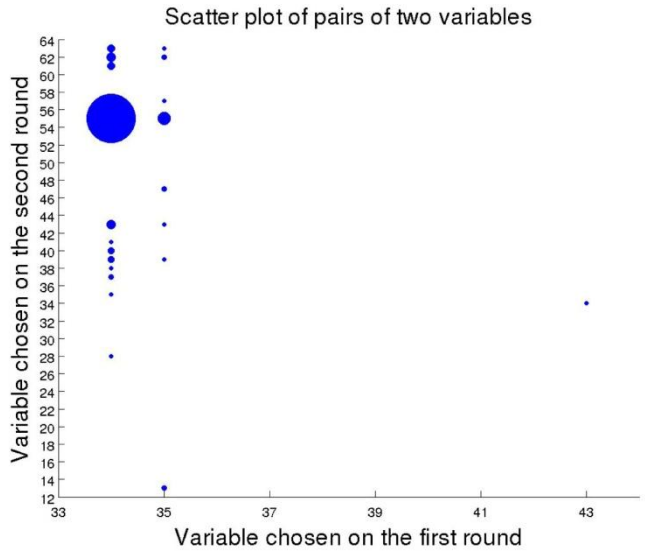

Figure 9. Scatter plot of first and second variable chosen, first variable is on x-axis and second variable on y-axis. The dots and their size represent the number of variable pairs found when two variables were selected. Atiya data set

This can also be noted from Figure 8, which displays a weight for each variable. The weight was calculated as a sum of additional test accuracies that the variable contributed, scaled to the largest weight. The bigger the weight, the more the variable increased the overall accuracy when added.

Looking at the two first variables, Figure 9 plots the pairs that occur the most often. It can be noted that the pairs

34 and 55, and 35 and 55 are the most important. These variables also explain the very good performance of E-LL: the values of bankrupt and healthy companies are grouped. The problem is very non-linear, but an easy classification problem. The E-LL is able to find such variables and thus performs well.

5.2.Philippe du Jardin datasets

The second and third data sets are somewhat similar. They were both used in the thesis by Philippe du Jardin. The 2002 dataset comprises companies that have accounting data from the year 2002 and net equity data from the year 2001. The bankruptcy decisions, or more accurately, decisions of reorganization or liquidation, are from the year 2003. The 2003 dataset was constructed similarly. In both datasets, the proportion of healthy and bankrupt companies is 50:50. In total, there were 500 and 520 observations respectively. The companies are all from the trade sector and they have a similar structure, from the juridical and assets points of view. In addition, healthy companies were still running in 2005, and had activities at least during four years. The ages of the companies were also considered, in order to obtain a good partition of companies of different ages (Jardin, 2007). Both datasets have 41 variables. The labels of the variables are presented in Table 3. Outliers were placed at the theoretical limits of the variables in question.

The mean test accuracies obtained from the Monte-Carlo Cross-test are presented in Figures 10 and 11. The 2002 dataset was notably easier to fit: the mean accuracies for all the methods are higher than in 2003. This can be due to the environment when the information was collected. For example, global economic phenomena are not modeled in this paper, but their effects show on individual companies. There might also be unpredictable causes for bankruptcy, such as the events of September 11th 2001 that might show in the du Jardin 2003 data. The standard deviation in du Jardin 2002 is around 0.02 for all the methods except LDA with variable selection, which is close to 0.04. In du Jardin 2003, the standard deviation is around 0.03, except again for LDA with variable selection, which has standard deviation close to 0.04.

The E-LL achieves a similar performance to that of the comparison methods, which is a good start. However, the interpretability of the variables is one goal of this paper. Figures 12 and 13 present the percentage of occurrence of each variable when the first chosen variable is considered.

We note that variables 16, 17, and 18 are chosen more often than other variables. Some of the variables are not chosen at all.

Table 3. Variables used in the du Jardin datasets.

X1	Profit before Tax/Shareholders' Funds
X2	Net Income/Shareholders' Funds
X3	EBITDA/Total Assets
X4	EBITDA/Permanent Assets
X5	EBIT/Total Assets
X6	Net Income/Total Assets
X7	Value Added/Total Sales
X8	Total Sales/Shareholders' Funds
X9	EBIT/Total Sales
X10	Total Sales/Total Assets
X11	Gross Trading Prot/Total Sales
X12	Operating Cash Flow/Total Assets
X13	Operating Cash Flow/Total Sales
X14	Financial Expenses/Total Sales
X15	Labor Expenses/Total Sales
X16	Shareholders' Funds/Total Assets
X17	Total Debt/Shareholders' Funds
X18	Total Debt/Total Assets
X19	Net Operating Working Capital/Total Assets
X20	Long Term Debt/Total Assets
X21	Long Term Debt/Shareholders' Funds
X22	(Cash + Marketable Securities)/Total Assets
X23	Cash/Total Assets
X24	(Cash + Marketable Securiti es)/Total Sales
X25	Quick Ratio
X26	Cash/Current Liabilities
X27	Current Assets/Current Liabilities
X28	Quick Assets/Total Assets
X29	Current Liabilities/Total Assets
X30	Quick Assets/Total Assets
X31	EBITDA/Total Sales
X32	Financial Debt/Cash Flow
X33	Cash/Total Debt
X34	Cash/Total Sales
X35	Inventory/Total Sales
X36	Net Operating Working Capital/Total Sales
X37	Accounts Receivable/Total Sales
X38	Accounts Payable/Total Sales
X39	Current Assets/Total Sales
X40	Change in Equity Position
X41	Change in Other Debts

Note: EBITDA = Earnings Before Interest, Taxes, Depreciation and Amortization

However, if a similar analysis is performed for a larger number of selected variables, such differences are not visible, mainly due to the reason that when a variable has been chosen in previous rounds, it cannot be chosen again. Also, as it can be seen on Figures 10 and 11, the

test accuracy does not improve much when more variables are added. Thus the importance of variables chosen later is presented in Figures 14, and 15, created by summing the additional accuracies that each variable brings to the test accuracy, and scaling them to the largest sum. Consequently, the variable bringing the highest accuracy has 100 percent weight.

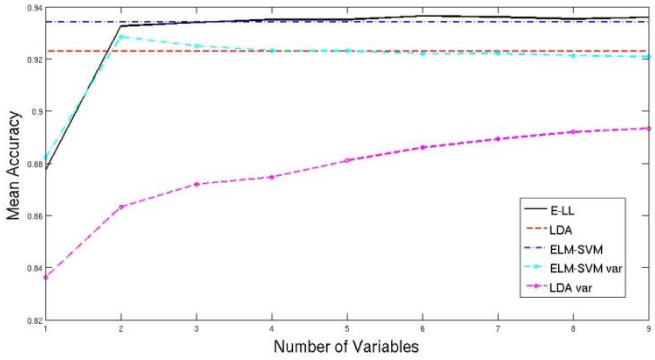

Figure 10. Mean accuracies of Ensembles of Locally Linear models (E-LL in black solid line), Linear Discriminant Analysis (LDA in red dashed line) with same variable selection than with E-LL (marked as LDA var, drawn as magenta dashed line with dots), Extreme Learning Machine Support Vector Machines (ELM-SVM, blue dash-dot line), also with variable selection (ELM-SVM var, cyan dash-dot line with dots). Du Jardin 2002 dataset, Monte-Carlo cross-test repeated 200 times

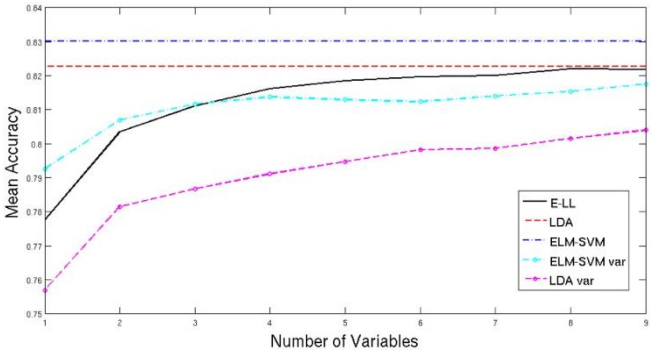

Figure 11. Mean accuracies of Ensembles of Locally Linear models (E-LL in black solid line), Linear Discriminant Analysis (LDA in red dashed line) with same variable selection than with E-LL (marked as LDA var, drawn as magenta dashed line with dots), Extreme Learning Machine Support Vector Machines (ELM-SVM, blue dash-dot line), also with variable selection (ELM-SVM var, cyan dash-dot line with dots). Du Jardin 2003 dataset, Monte-Carlo cross-test repeated 200 times

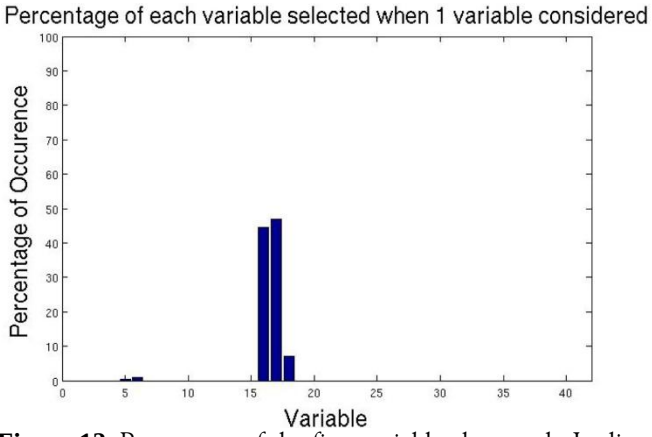

Figure 12. Percentage of the first variable chosen, du Jardin 2002

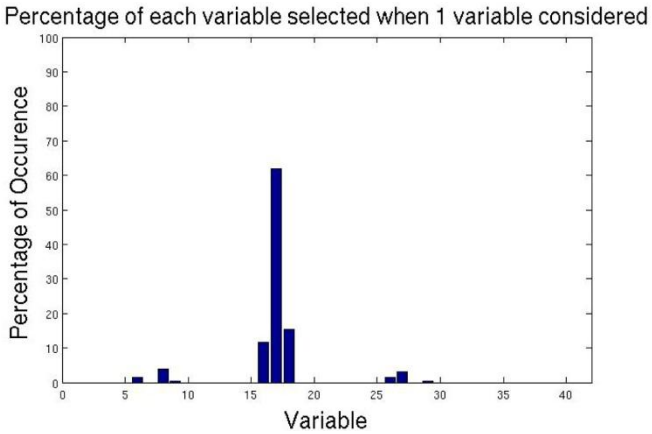

Figure 13. Percentage of the first variable chosen, du Jardin 2003

Figures 14 and 15 reveal that variables 16, 17 and 18 are important again. Also variables 1 to 6 have some importance in both of the datasets. The 2003 dataset also has other variables occurring, the most important being 8 and 27. However, this is not sufficient. Figures 16 and 17 represent the co- occurrence of the two first variables. The x-axis has the variable that was chosen the first. The y-axis has the variable selected the second. The bigger the dot, the more often the pair occurred. From these Figures it can be noted that in 2002, variables 16, 17 occurred with variables 1 to 3 and 6, and variable 18 with variable 1. In 2003, the situation is more diverse. Variable 17 occurs with variables 2, 3, 5, 9, 26, 31 and 34. Also, variable 16 occurred with variable 31.

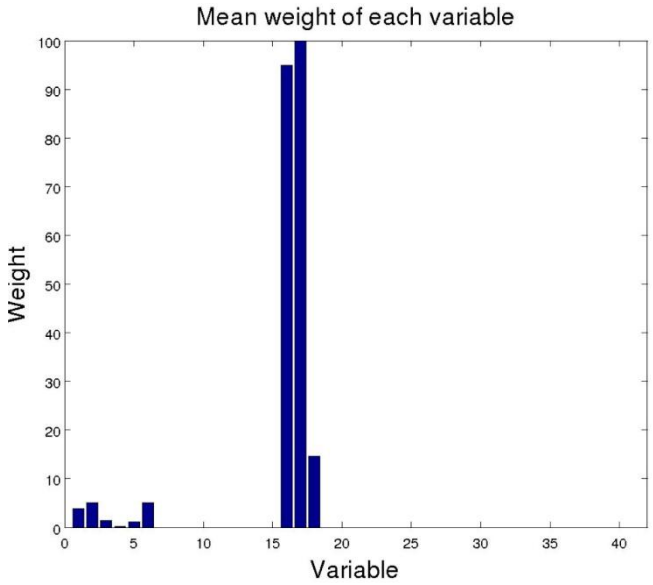

Figure 14. Weight added to each variable. The weight was calculated as a sum of additional test accuracies that the variable brought and scaled to the largest accuracy, du Jardin 2002 data

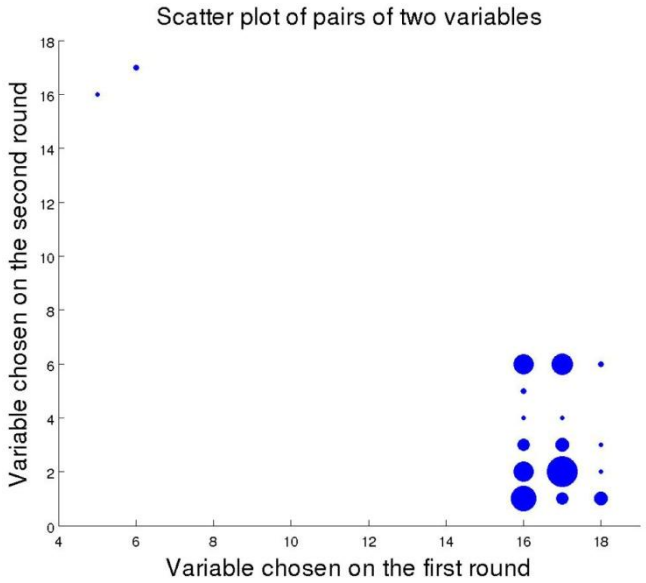

Figure 16. Scatter plot of first and second variables chosen, first variable is on x-axis and second variable on y-axis. The dots and their size represent the number of times a variable pair was selected. Du Jardin 2002 data

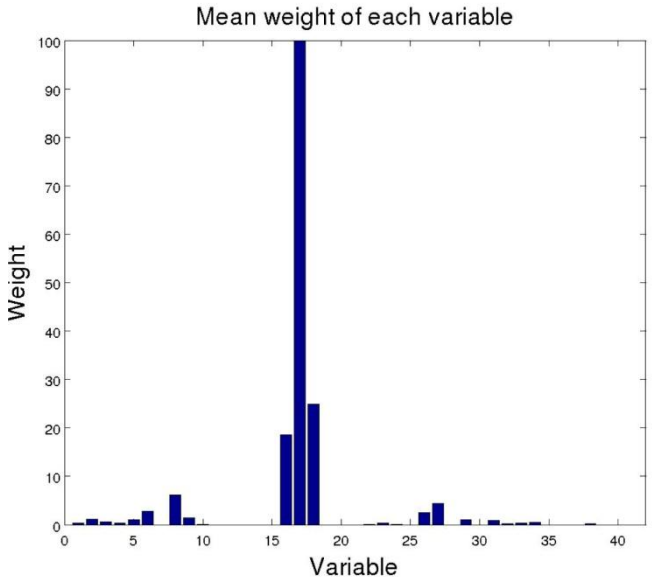

Figure 15. Weight added to each variable. The weight was calculated as a sum of additional test accuracies that the variable brought and scaled to the largest accuracy, du Jardin 2003 data

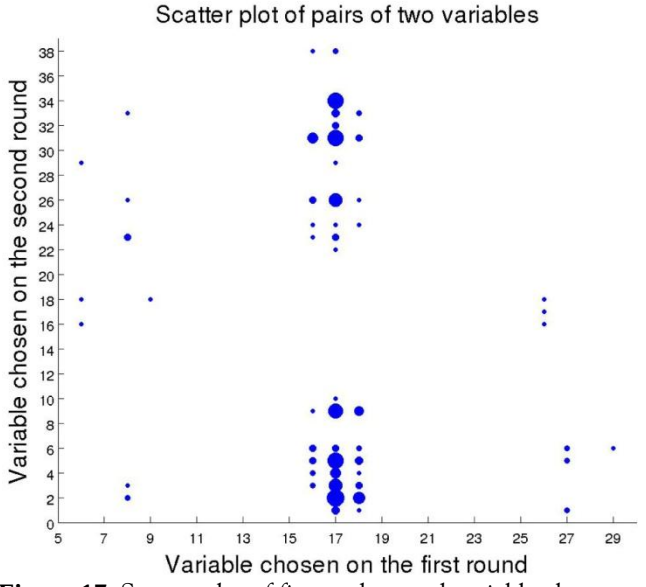

Figure 17. Scatter plot of first and second variable chosen, first variable is on x-axis and second variable on y-axis. The dots and their size represent the number of times a variable pair was selected. Du Jardin 2003 data

5.3 Pietruszkiewicz dataset

Pietruszkiewicz has developed a data set of 240 cases of which 112 are bankrupt and 128 healthy. In total there are 120 companies; the data come from two years in a row. Bankruptcies occurred from two up to five years after the observations (Pietruszkiewicz, 2004, 2008).

The 30 variables consist of ratios of different financial variables. However, variables 8 and 16 (sales/receivables) as well as 17 and 20 (sales/total assets), were exactly the same. The duplicates, 16 and 20, were removed. The remaining variables with their new labels are presented in Table 4.

Table 4. Variables used in the Pietruszkiewicz dataset

X1	cash/current liabilities
X2	cash/total assets
X3	current assets/current liabilities
X4	current assets/total assets
X5	working capital/total assets
X6	working capital/sales
X7	sales/inventory
X8	sales/receivables
X9	net profit/total assets
X10	net profit/current assets
X11	net profit/sales
X12	gross profit/sales
X13	net profit/liabilities
X14	net profit/equity
X15	net profit/(equity + long term liabilities)
X16	sales/total assets
X17	sales/current assets
X18	(365*receivables)/sales
X19	liabilities/total income
X20	current liabilities/total income
X21	receivables/liabilities
X22	net profit/sales
X23	liabilities/total assets
X24	liabilities/equity
X25	long term liabilities/equity
X26	current liabilities/equity
X27	EBIT (Earnings Before Interests and Taxes)/total assets
X28	current assets/sales

The Pietruszkiewicz and du Jardin datasets are fairly similar in terms of variables. Both of them use financial ratios. The ratios are not exactly the same in all the cases, but very similar.

Figure 18 presents the mean accuracies obtained with the Pietruszkiewicz dataset. The standard deviation is around 0.06 except for ELM-SVM and LDA, for which it is 0.05. Consequently, the results are comparable to each other. E- LL achieves a similar performance than that of all the other methods except ELM-SVM with only three variables. ELM-SVM obtains a better test accuracy than E-LL, but it also uses the information from the whole dataset. When tried with less variables, its performance is not as good.

Figure 19 represents the percentage of the time that each variable was chosen as the first one. Variables 9, 10, 11, 15 and 22 obtain the most important percentages. When the added accuracy is considered in Figure 20, again the same variables seem to be the most important. That is due to the fact that even with one variable, the E-LL achieves a test accuracy of 70%. When pairs of the variables are considered, as seen in Figure 21, it can be noted that variables 9, 10 and 11 are combined with variable 17, and variable 9 with 1 and 28 and variable 11 with 28.

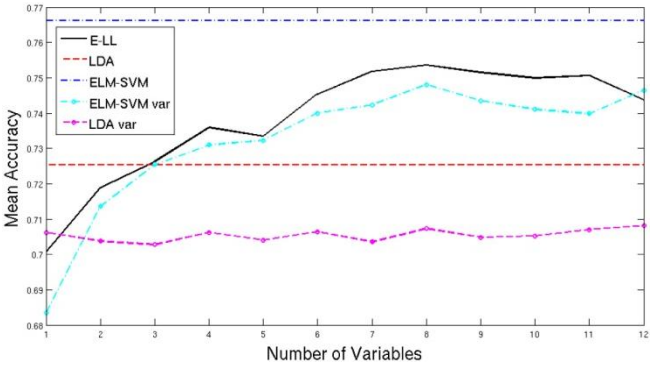

Figure 18. Mean accuracies of Ensembles of Locally Linear models (E-LL in black solid line), Linear Discriminant Analysis (LDA in red dashed line) with same variable selection than with E-LL (marked as LDA var, drawn as magenta dashed line with dots), Extreme Learning Machine Support Vector Machines (ELM-SVM, blue dash-dot line), also with variable selection (ELM-SVM var, cyan dash-dot line with dots). Pietruszkiewicz dataset, Monte-Carlo cross-test repeated 200 times.

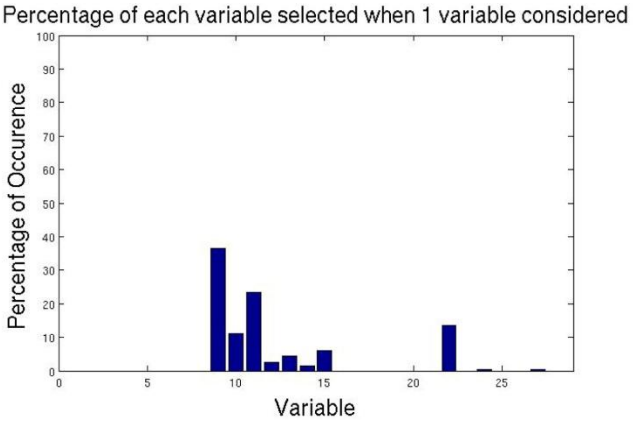

Figure 19. Percentage of the first variable chosen, Pietruszkiewicz data.

6. Financial Discussion

It can be observed that E-LL does not clearly outperform the comparison methods in the du Jardin datasets and in the Pietruszkiewicz dataset, but it does not do worse either. In the Atiya dataset it outperforms all the other methods. Also, ELM-SVM and LDA use information from the whole dataset, meaning all the variables, while E-LL in these methods uses only 1 to 9 or 12 variables. Thus, with less information, similar results can be

obtained, which opens possibilities for interpreting the results.

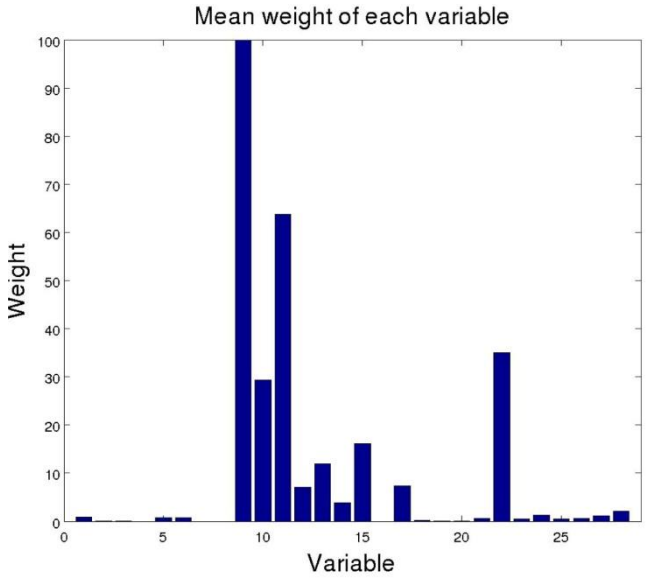

Figure 20. Weight added to each variable. The weight was calculated as a sum of additional test accuracy that the variable brought and scaled to the largest accuracy, Pietruszkiewicz data.

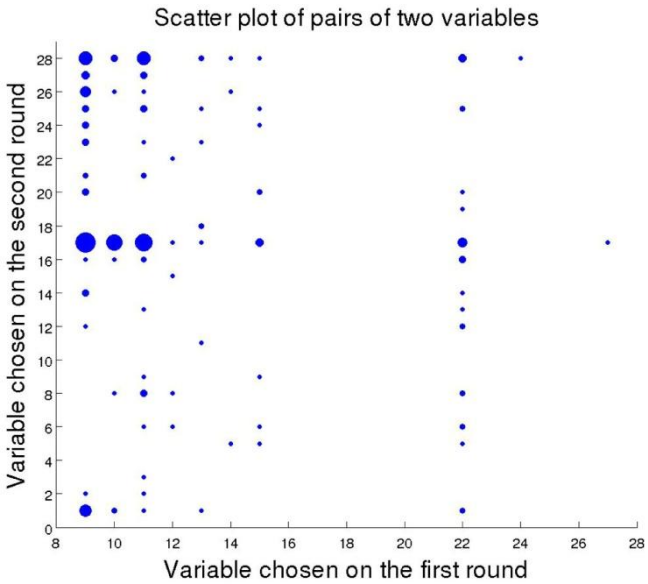

Figure 21. Scatter plot of first and second variable chosen, first variable is on x- axis and second variable on y-axis. The dots and their size represent the number of variable pairs found when two variables were selected. Pietruszkiewicz data.

In the Atiya dataset (see Tables 1 and 2), variables 34 (PE ratio) and 35 (price cash-flow ratio) were the most important when considered alone. Especially variable 34 was often combined with variable 55 (total assets). In both du Jardin datasets (see Table 3), variables 16 (Shareholders' funds/Total Assets), 17 (Total Debt/Shareholders' Funds) and 18 (Total Debt/Total Assets) were important. In the 2002 dataset they were often combined with variables 1 (Profit before tax/Shareholders' funds), 2 (Net Income/Shareholders' Funds) and 6 (Net Income/Total Assets), but in 2003 the results were more diverse, adding importance also to variables 5 (EBIT/Total Assets), 9 (EBIT/Total Sales), 26 (Cash/Current Liabilities), 31 (EBITDA/Total Sales) and 34 (Cash/Total Sales). In the Pietruszkiewicz dataset (see Table 4, variables 9 (net profit/total assets), 10 (net profit/current assets), 11 (net profit/sales), 15 (net profit/(equity + long term liabilities)) and 22 (current assets/sales) seem to be the most important ones. Variables 9, 10, and 11 were often combined with variable 17 (sales/current assets), but also with variable 28 (current assets/sales).

The different results highlight several relationships. Starting from the Atiya dataset, it can be observed that the main indicator for predicting bankruptcy is the cash flow (variables 34 and 35). Indeed, the main (and first) difficulties met by the firm come from the problem of cash and liquidity. To face this problem, a firm tries to engage some measures called in the financial literature 'restructuring measures'. If they fail, the liquidity ratios deteriorate. That can be seen in the Pietruszkiewicz dataset in particular with variable 22. Consequently, the 'activity cost' (through the policy of trade receivable and accounts payable) increases and a decrease in profitability can be observed, especially through variables 10 and 11.

At the end, the drop in profitability leads to a decrease in financial structure as can be seen in the results of du Jardin dataset, especially through variables 16, 17 and 18. Of course, the relationship between profitability and financial structure is highlighted by the combination between, on the one hand, variables 1, 2 and 6 and, on the other hand, variables 16, 17 and 18.

In fact, the different results highlight all the stages in a bankruptcy. First is the cash problem (Atiya), second is the problem of profitability (Pietruszkiewicz) and the last stage (just before bankruptcy) is the impact of profitability on the financial structure (du Jardin). The Atiya and Pietruszkiewicz datasets have financial data from earlier stages than du Jardin datasets, which has data only from the previous year before bankruptcy.

7. Conclusions

The results obtained from tests on four datasets show that the performance of Ensembles of Locally Linear models is

comparable to that of the other methods tested, ELM-SVM and LDA, especially regarding accuracy. The main advantage of E-LL is that the variable selection embedded into the method provides a good interpretability of the results. From a financial point of view, the variables selected by the E-LL are relevant. The problem of cash, leading to a problem of profitability and later resulting in the impact of profitability on the financial structure can be observed from the datasets. Different datasets highlight different variables, which might also be due to the fact that the data are not collected from equally long periods before bankruptcy. The datasets with data close to the bankruptcy highlight variables describing the final stage before bankruptcy. Being able to extract important variables also opens visualization possibilities.

Further work is needed both in applications to both finance and ensemble methodology. First, the interpretation of variables should be further discussed. Could some of the variables be left out at the beginning of the analysis? That would save efforts and money in data collection, and lead to better profits in reality. Also, could the E-LL methodology be applied to other fields? Second, ensemble creation techniques are numerous. Would the results change if a different merging method was used? Research on ensembles of locally linear models in bankruptcy prediction could be continued for example by following these paths.

References

Altman, E.I. 1968. Financial ratios, discriminant analysis and the pre- diction of corporate bankruptcy. *The Journal of Finance,* 23(4):589–609.

Atiya, A.F. 2001. Bankruptcy prediction for credit risk using neural net- works: A survey and new results. *Neural Networks, IEEE Transactions on,* 12(4):929–935.

Beaver, W.H. 1966. Financial ratios as predictors of failure. *Journal of Accounting Research,* 4(3):71–111.

Bontempi, G. Bersini, H. & Birattari, M. 2001. The local paradigm for modeling and control: from neuro-fuzzy to lazy learning. *Fuzzy Sets and Systems,* 121(1):59–72.

Bontempi, G. Birattari, M. & Bersini, H.1998[1]. Local learning for data analysis. In F. Verdenius and W. van den Broek, editors, Proceedings of the 8th Belgian-Dutch Conference on Machine Learning, Benelearn'98:62–68.

Bontempi, G. Birattari, M. & Bersini, H. 1998[2]. Machine Learning: ECML-98, volume 1398 of Lecture Notes in Computer Science, chapter Recursive lazy learning for modeling and control:292–303.

Boser, B.E. Guyon, I.M. & Vapnik, V.N. 1992. A training algorithm for optimal margin classifiers. In COLT'92: Proceedings of the fifth annual workshop on Computational learning theory:144–152.

Bottou, L. & Vapnik, V. 1992. Local learning algorithms. *Neural Computation,* 4(6):888–900.

Breiman, L. 1996. Bagging predictors. *Machine Learning,* 24(2):123–140.

Brown, G. Wyatt, J. Harris, R. & Yao, X. 2005. Diversity creation methods: a survey and categorisation. *Information Fusion,* 6(1):5-20.

Burges, C.J.C. 1998. A tutorial on support vector machines for pat- tern recognition. *Data Mining and Knowledge Discovery,* 2(2):121–167.

Cover, T. & Hart, P. 1967. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on,* 13(1):21-27.

Cristianini, N. & Shawe-Taylor, J. 2000. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press.

Fisher, R.A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics,* 7(2):179–188.

Frénay, B. & Verleysen, M. 2010[1]. Parameter-free kernel in extreme learning for non-linear support vector regression. To appear in Neurocomputing Special Issue: Advances in ELM.

Frénay, B. & Verleysen, M. 2010[2]. Using svms with randomised feature spaces: an extreme learning approach. In ESANN2010: 18th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning:28–30.

Hearst, M. A. 1998. Support vector machines. *IEEE Intelligent Systems,* 13(4):18–28.

Huang, G.B. Zhu, Q.Y. & Siew, C.K. 2006. Extreme learning machine: Theory and applications. *Neurocomputing,* 70:489–501.

Jacobs, R.A. Jordan, M.I. Nowlan, S.J. & Hinton, G.E. 1991. Adaptive mixtures of local experts. *Neural Computation,* 3(1):79–87.

Jardin, P.D. 2007. Prévision de la défaillance et réseaux de neurones: l'apport des méthodes numériques de sélection de variables. PhD thesis, Université de Nice-Sophia-Antipolis.

Kumar, P.R. & Ravi, V. 2007. Bankruptcy prediction in banks and firms via statistical and intelligent techniques - a review. *European Journal of Operational Research,* 180(1):1–28.

Kuncheva, L.I. 2004. Combining Pattern Classifiers. Wiley-Interscience.

Kuncheva, L.I. & Whitaker, C.J. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning,* 51:181–207.

Lendasse, A. Wertz, V. & Verleysen, M. 2003. Model selection with cross-validations and bootstraps - application to time series prediction with RBFN models. In Kaynak, O. Alpaydin, E. Oja, E. & Xu, L. editors, ICANN 2003, Joint International Conference on Artificial Neural Networks, Istanbul (Turkey), volume 2714 of Lecture Notes in Computer Science:573–580.

Miche, Y. Eirola, E. Bas, P. Simula, O. Jutten, C. Lendasse, A. & Verleysen, M. 2010. Ensemble modeling with a constrained linear system of leave-one- out outputs. In Verleysen, M. editor, ESANN2010: 18th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning:19–24.

Miche, Y. Bas, P. Jutten, C. Simula, O. & Lendasse, A. 2008. A methodology for building regression models using extreme learning machine: OP-ELM. In Verleysen, M. editor, ESANN 2008, European Symposium on Artificial Neural Networks.

Min, J.H. & Lee, Y.C. 2005. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert Systems with Applications,* 8(4):603–614.

Myers, R.H. 1990. Classical and Modern Regression with Applications. Duxbury.

Ohlson, J.A. 1980. Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research,* 18(1):109–131.

Pietruszkiewicz, W. 2004. Application of Discrete Predicting Structures in an Early Warning Expert System for Financial Distress. Ph.d. thesis, Faculty of Computer Science and Information Technology, Szczecin University of Technology.

Pietruszkiewicz, W. 2008. Dynamical systems and nonlinear kalman filtering applied in classification. In Proceedings of 2008 7th IEEE International Conference on Cybernetic Intelligent Systems:263–268.

Pochet C. 2002. Institutional complementarities within corporate governance systems: A comparative study of bankruptcy rules. *Journal of Management & Governance,* 6(4):343–381.

Polikar, R. 2007. Bootstrap - inspired techniques in computation intelligence. *Signal Processing Magazine, IEEE,* 24(4):59–72.

Polikar, R. 2006. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine,* 6(3):21–45.

Rokach, L. 2010. Ensemble-based classifiers. *Artificial Intelligence Review,* 33(1-2):1–39.

Rossi, F. Lendasse, A. Francois, D. Wertz, V. & Verleysen, M. 2006. Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometrics and Intelligent Laboratory Systems,* 80(2):215–226.

Shin, K.S. Lee, T.S. & Kim, H.J. 2005. An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications,* 28(1):127–135.

Tumer, K. & Oza, N.C. 2003. Input decimated ensembles. *Pattern Analysis and Applications,* 6(1):65–77.

Vapnik, V.N. 1998. Statistical Learning Theory. Wiley.

Verikas, A. Kalsyte, Z. Bacauskiene, M. &. Gelzinis, A 2010. Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: A survey. *Soft Computing,* 14(9):995–1010.

Wilson, R.L. & Sharda, R. 1994. Bankruptcy prediction using neural networks. *Decision Support Systems,* 11(5):545–557.

Correspondence: amaury.lendasse@aalto.fi