Bankruptcy Prediction using Hybrid Neural Networks with Artificial Bee Colony

Said Marso, Mohamed EL Merouani

Abstract—**Credit risk now is considered like the most important risk faced by banks and financial institutions. For this reason, predictive analytics became the barometer of financiers with a main interest to corporate bankruptcy prediction (CBP), or referred to as financial distress prediction. Predictive analytics techniques are subdivided into two groups: traditional statistical (e.g. discriminant analysis and logistic regression) and machine learning (e.g. artificial neural networks, support vector machines and random forest). Trained by metaheuristic algorithm such as artificial bee colony (ABC), an ANN model called ABCNN is applied in CBP and this new hybrid model is the contribution of our article. Our model is compared with two other models in order to investigate its efficiency: the first model is multiple discriminant analysis (MDA) and the second one is an ANN trained by the most common learning algorithm a back propagation (BPNN).**

The models mentioned above were applied at two different periods in order to evaluate their efficiency. The first period is one year before the bankruptcy, and the second one is three years before. Performance was evaluated in terms of several metrics: Accuracy, AUC-score, type I and type II errors. The experimental results indicate that ANN models, on average, are approximately 10% more accurate in relation to MDA in different periods. Our Model ABCNN led to 92% accuracy, whereas BPNN's and MDA's led to 91% and 81% accuracy, respectively in one year before the bankruptcy. However, when we apply in the period three years before the bankruptcy, we find that ABCNN's result is 80.94% followed by BPNN with 81.05% , then MDA - with 67.22%. The right data preprocessing applied has increased significantly the predictive power of all the models mentioned above. In conclusion, experimental results show that the ABC algorithm in an ABC model could be considered to predict potential corporate bankruptcy.

Index Terms—**Corporate bankruptcy prediction, artificial neural networks, artificial bee colony, training, imbalanced data, machine learning.**

I. Introduction

In recent years, machine learning (ML), a cross between Statistics and Computer Science and belonging to artificial intelligence (AI) field, has been in constant improvement due to a better accessibility of Big Data and low-cost computations [1]. The magic of machine learning lies in its ability to learn on its own and improve over time, but more importantly, machine learning focuses on exploiting data and not on the design of complex probabilistic models.

ML is a significant breakthrough in the financial sector; its applications are varied: stock and currency market prediction, trading, portfolio management, credit scoring and corporate bankruptcy prediction (CBP) to name but a few [2]. The real success stories of machine learning in the financial sector can be found in CBP [3]. This latter is a major issue in banks because of protection of creditor interests, or restoration of the sustainability of the firm, and the necessity of anticipation of serious economic and financial difficulties that a firm is likely to meet. In the framework of the Basel Committee, banks have to propose a systematic evaluation of risks. This implies an accurate estimation of the probability of default of their customers, and therefore a possible reworking of their evaluation methods [4]. Accordingly, CBP intends to solve the problem by classifying a firm into two groups: bankrupt or healthy firm during the following years. This classification is through financial ratios describing the firm situation over a given period.

For decades, there has been an abundant academic research literature concerning CBP. The earliest review in corporate bankruptcy domain focused on statistics-based techniques [5]. Two complete reviews of Bellovary et al. [6] and Kumar and Ravi [7], covering nearly the same period from 1966 to 2005, traced the historical summary of CBP studies and introducing some important trends for future research in this area. More recently, Sun et al. [8] published the latest progress on prediction models in the CBP domain. The article presents a full summary of these models; shortcomings and oversights of past researches were discussed, and some directions for further research suggested. Another article written by Chen et al. [9] summarize the traditional statistical techniques and state-of-the-art intelligent methods for CBP, with the emphasis on the most recent achievements as promising trend in CBP. More than this, some researchers emphasized specific predictive models in their review papers. For example, Verikas et al. [3] showed a comprehensive review of hybrid techniques applied to CBP; Wozniaka et al. [3] discussed how to construct multiple classifier systems; Brabazon et al. [10] introduced the utilizing of natural computing in the widespread financial problems and some studies focused on application of ANNs in CBP context [11], [12].

From articles cited previously, we learn that prediction algorithms are classified in two groups: those belonging

Manuscript received October 4, 2019; revised March 11, 2020.
Said Marso (corresponding author, email:said.marso@gmail.com) and M. El Merouani (email:m_merouani@yahoo.fr) are with the Department of Mathematics,Abdelmalek Essaadi University - Faculty of the Sciences, Tetouan 93002, Morocco.

to traditional statistical methods and those to artificial intelligence. In AI, methods include ANN, fuzzy set theory (FST), decision trees (DTs), case-based reasoning (CBR), support vector machines (SVMs), rough set theory (RST), genetic programming (GP) and hybrid learning among others. Moreover, these methods help to improve the accuracy of statistical techniques without making any restrictive assumptions.

Among AI models, ANNs are among the most used in financial engineering and especially in CBP [3], [13]. ANNs have a strong mapping ability, based on the network structure, to detect complex and nonlinear relationships, which is useful to find intrinsic patterns in complex financial data. These are incremental models allowing new data to be trained in order to update the previous training result. Comparing with statistical techniques, ANNs have often reached a better performance in terms of prediction and classification accuracy [10], [8], [14].

However, there are some disadvantages influencing ANNs performance. First, their 'black box' nature makes their causal interpretation of financial ratios effects and connection weights more complex. Second, training is time consuming and required large data sets to achieve optimal model performance, which may diminish the predictive power of the model. Finally, ANNs training/learning/optimization require technical expertise to find an appropriate network structure (the function) and the weights (the parameters of the function).

Researches on hybrid ANN for CBP, focused on improving and eliminating drawback of ANN. These Researches began to boom especially in the recent last decade. Bellovary et al. [6] note that a trend was visible in which stand-alone models were losing popularity, whereas hybrid models were gaining attention and showing improvements in performance. This example of the power of hybrid models is more broadly justified by Sun et al. [8]. Sun et al. compare a collection of hybrid models, where they focus on why and how different techniques were combined. The hybridization of ANN can be realized, generally, by three mode: integration mode (selecting relevant features or searching optimal weights), cascade mode, and clustering combination mode. A more detailed discussion on building a hybrid system is in [12], [15], [3].

Searching optimal weights (i.e. training ANN) has traditionally been carried out using the back-propagation (BP) algorithm [12], despite it has a number of undesirable features impacting its performance. Indeed, it has a tendency to fall in local minima, to have a computational intensity and a slow convergence speed, an absence of formal theory, and finally, algorithm depends on initial values (connection weights, learning rate, momentum).

Recently, many researchers have shown that the performance of learning models can be significantly enhanced by the hybridization of metaheuristic algorithms, and this could be considered as a good alternative to overcome the limitations of BP [12], [16], [17]. Unlike gradient-based algorithms, which require a continuous and differentiable function, and lack exploration ability, the metaheuristic algorithms overcome these restrictions due to their ability to address complex, nonlinear and non-differentiable problems in one hand, and their fine balance between exploration and exploitation of a search space on the other hand. Particle Swarm Optimization (PSO), Tabu Search, Ant Colony Optimization (ACO), and Evolutionary Algorithms (EAs)- which include Genetic Algorithms (GA) and Differential Evolution (DE)- are typical examples of metaheuristics. In this work, the Artificial Bee Colony (ABC) algorithm is proposed in training ANN to allow our model to predict bankruptcy, and the performance of the algorithm is compared with BP algorithm and another statistical classification model MDA.

The paper will proceed as followed: the next section is devoted to a literature review and models widely applied in CBP. Section 3 describes how the ABC algorithm is implemented in the training ANN. In Section 4, we expose the different steps used for the realization of our empirical study. Experiments results will be given in section 5. And, finally, this article will end with a discussion, some conclusions and axes of improvement for future research.

II. Literature Review

In 1966, one of the earliest models for predicting bankruptcy is developed by Beaver [18], through the use of one-dimensional ratio [18]. From 1968, Altman states that the single variable analysis method does not fully account for complexity of the failure process. Therefore, he constructed the famous Z-score model which is a MDA function with five financial ratios. Altman [19] analysis found that its predictive power at the year before bankruptcy was significantly better than one conducted by Beaver. More than that, the superiority of MDA has been proven then by Deakin [20]. Even though, several shortcomings are related to the use of MDA because the validity of results is dependent on restrictive assumptions.

In parallel, other models were developed such as the logistic regression (LR). Unlike MDA, the LR does not require assumptions such as linearity and normality of the predictive variables, and homoscedasticity of the two classification groups. But, LR model, like MDA, relies on other assumptions, as no multicollinearity among the predictive variables [21]. The seminal study using the LR model in CBP was developed by Ohlson [22], then Aziz et al. in 1988, they employed LR to predict firm failure [6]. The results of Altman and Ohlson have been compared empirically by several studies [6] including the one of Begley et al. [23], who concludes Ohlson's model is the most successful in CBP. However, Begley et al. [23] also indicated that the proposed models based on Altman and Ohlson do not perform well on more recent data in comparison with the originally estimated models. Also, Lo [24] built the CBP model with MDA and logistic regression. The results showed that if predictive variables satisfied normal distribution assumption then the predictability of MDA model will be higher than LR, and vice versa. MDA and LR statistical parametric techniques are widely used in corporate bankruptcy prediction (CBP) [25]. Their advantage is to estimate directly the contribution of each financial ratio to the risk of the firm helping users to identify key issues for a failing firm. To face the many constraints associated with statistical techniques, machine learning models including ANN have been exploited by academics and practitioners.

Research on using ANN in CBP started around 1990. One of the first attempts is described by Odom and Sharda [26]. Many studies have focused on comparing the performance of ANN with the statistical techniques mentioned above. Coats and Fant in 1992 [6] compare the performance of ANN with the MDA of Altman. Their conclusion was that ANN gives better results than MDA and in the same year, Tam and Kiang made another comparison between LR, K-Nearest neighbor, decision trees and ANN. They report that the predictive power of the ANN applied to CBP outperforms all other models [6]. However, Altman et al. [27] found that MDA is more efficient than ANN. Also, Pompe and Feelders [28] report that ANN do not outperform MDA and decision trees. During the last decade, most research was focused on improving and comparing existing models. And according to the reviews of literature already cited, Kumar and Ravi [7] display that, among all models, ANN were the most popular option. According to Bellovary et al. [6], analysis of accuracy of the models suggests that ANN is the most promising model for CBP. In addition, Chen and DU [29] found that ANNs also performed better than data mining techniques. Other research came to the same conclusion on ANNs [8], [13].

The process by which ANN updates its free parameters to capture the patterns is called the training i.e. searching optimal weight. The most common training algorithm in reviewed of CBP was the BP [12]. A different approach from the ones outlined above, is the family of methaheuristic. This research field, with its roots copied from the biological evolution process, has also received attention in CBP [12].

Abdelwahed and Amir [30] developed a two step technique for designing a CBP tool based on genetic algorithm (GA) and an ANN. In the first step, GA is used to select a subset of relevant predictive variables. Then, in the second step, GA is applied to optimize the topology of the ANN. Hu [31] developed an ELECTRE-based single-layer perceptron (SLP) approach and then, GA was designed to determine its connection weights. Its application to CBP demonstrated that it performed better than the traditional SLP, the MLP, and some other single classifier models. In the same year, Chauhan et al. [32] applied differential evolution (DE) to train a wavelet ANN Called DEWNN in CBP. The results showed that DEWNN outperformed compared to other models and it is a very effective machine learning model for classification problem. Likewise, Ravisankar et al. [33] used ANNs, RST and Genetic Programming (GP) to construct different two-phase hybrids in such paradigm, each phase is taking one technique as feature selection method followed by another one as classifier. They showed the GP-GP performed better compared to other hybrids in terms of accuracy, sensitivity, specificity and AUC.

There are no articles in the literature reviewing which uses ABC algorithm to trained ANN in CBP area. This is confirmed by Solder-Dominguez et al. [34]. Their recent article sketches an accurate image of the works related to the use of metaheuristics in solving problems in the finance sector. Several studies have dealt with the discussion of strengths of ABC algorithm to train an ANN in many different disciplines. Karaboga and Akay in 2007 and Omkar and Senthilnath in 2009 in signal processing applications. Hsieh et al. (2011) in the integration of a system capable of combining wavelet transformations and the ABC-based recurrent neural network (called ABC-RNN) for reliable price forecasting of equities . And Shah et al. (2011) in the time series showed that MLP-ABC is better than MLP-BP [35].

III. Hybrid Neural Network

ANNs are computational structures imitating the mechanism of the accumulation of information using by the central nervous system. The components involved in ANN models are the nodes, weights, and layers. There are different types of ANN models, including MLP (in general FNN) which will be develop in this study. The structural representation of an MLP makes it appealing because it allows perceiving a computational model in a network form. Moreover, several researchers praised MLP for its universal approximation ability of any function [36].

A. Architecture of MLP

MLPs consist of a population of neurons interconnected through complex signaling pathways. They use this structure to analyze complex interactions between a group of measurable variables in order to predict an outcome. MLPs possess layers of neurons connected by synaptic (weights) links (Fig. 2). These layers are arranged in layer-by-layer basis: input layer, one or more middle hidden layers, and an output layer. Neurons in the input and output layers correspond to the predictive variables (financial ratios) and binary variable bankrupt/healthy, respectively. Neurons in adjacent layers communicate with each other through activation functions, which convert the weighted sum of a neuron's inputs into an output (Fig. 1).

MLPs can be more or less complex i.e. composed of a variable number of hidden layers. When a MLP has two or more hidden layers, it is called a deep neural network (DNN). Let's note that the input layer is excluded when counting the number of layers in an MLP because inputs (represented by squares in Fig. 2) are not neurons: they do not achieve any information processing.

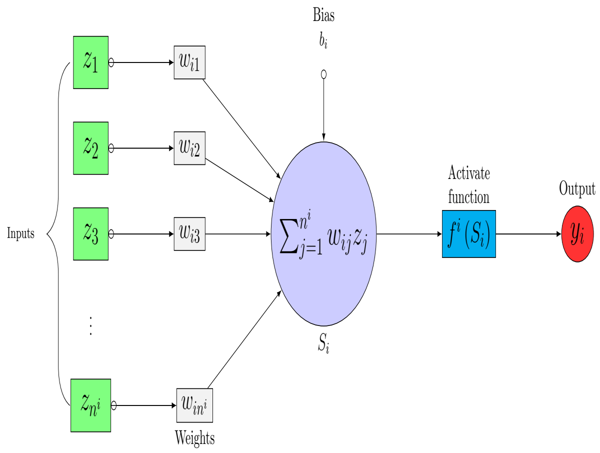


Fig. 1: Node in the network

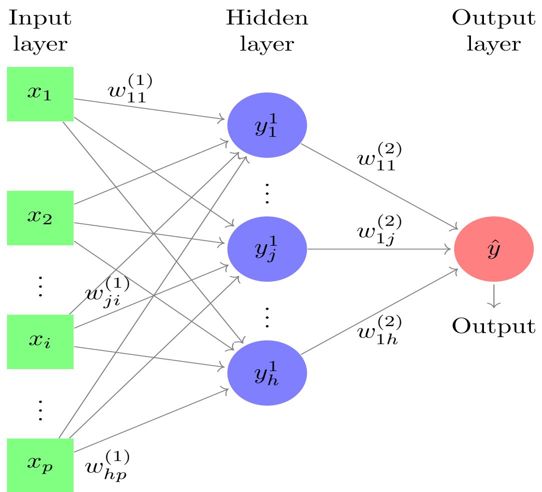


Fig. 2: architecture of a multilayer perceptron neural network, where input layer has p input nodes, one hidden layer, and output layer has one node.

A node in an MLP deals with information processing coming from connection weights (Fig. 1). Mathematically, for a node i the output y_i (excitation) is written as the following:

$$y_i = f_i\left(S_i\right) = f_i\left(\sum_{j=1}^{n^i} w_{ij} z_j - b_i\right) \tag{1}$$

where n^i is the total incoming connections, z is the input, w^i is the weight, b_i is the bias, S_i is called potential (inspired by the biology) and $f_i(.)$ is the activation function at the i^{th} node to limit the amplitude of the output into a certain range.

Fig 2 illustrates an MLP structure. i.e., a phenotype of a function $f(x, w)$, which is parameterized by a p input (financial ratios) vector $x = \langle x_1, x_2, \cdots, x_p \rangle$ and a d-dimensional real-valued weight vector $w = \langle w_1, w_2, \cdots, w_d \rangle$. The function $f(x, w)$ is computed according to the following expression:

$$\hat{y} = f(x, w) = f^{(2)}\left(\sum_{j=1}^{h} w_{1j}^{(2)} y_j^{(1)} - b_1^{(2)}\right) \tag{2}$$

where

$$y_j^{(1)} = f^{(1)}\left(\sum_{i=1}^{p} w_{ji}^{(1)} x_i - b_j^{(1)}\right) \tag{3}$$

where $w_{ji}^{(l)}$ is the weight linking the i neuron of the layer $l-1$ to j neuron in the layer l and $f^{(l)}$ is an activation function of the layer l.

Therefore, solving a problem using an MLP consists of finding an appropriate structure/architecture (numbers of layers and nodes, activation functions, etc) and an appropriate weight vector w (the weights optimization) by using some learning algorithm. This paper focuses on the weights optimization by keeping structure fixed to the initial choice.

B. Model Training

Once the structure is selected, the weights, through a supervised learning approach, are fitted on a training set labeled (X, Y) with N input-output pairs where $X = (x_1, x_2, \cdots, x_N)$ and $Y = (y_1, y_2, \cdots, y_N)$. Each input $x_k = \langle x_{k1}, x_{k2}, \ldots, x_{kp} \rangle$ has a corresponding 1-dimensional desired output vector y_k (bankrupt or healthy).

Initial connection weights are generally selected randomly. Then, for all $k = 1$ to N MLP predict an output $\hat{y}_k$, which is then compared with the desired output y_k using a loss/error/cost function. The utility of training data set is to minimize the loss function (Eq. (4)) by iterating, so parameters/weights will win in precision.

$$F(\hat{y}, y, w) = -y \ln \hat{y} - (1-y)\ln(1-\hat{y}) + \frac{\alpha}{2}\|W\|_2^2 \tag{4}$$

Loss function F (Eq. (4)) is, in general, composed of an error term and a regularization term. The error term evaluates how an MLP fits the data and the regularization term $\frac{\alpha}{2}\|W\|_2^2$ is used to prevent over-fitting by controlling the effective complexity of the MLP.

C. Conventional optimization approaches

Finding a suitable algorithm for the MLP optimization has always been a difficult task. MLP optimization using conventional gradient based algorithms is viewed as an unconstrained optimization problem [37]. Gradient-descent algorithm is an iterative numerical method to optimize unconstrained problem. This approach starts with an initial weight w_0 and generates a sequence of weight vector $\boldsymbol{w_1, w_2, \ldots}$ such that $\boldsymbol{F}$ is reduced in each iteration. The connection weights at iteration t are updated as:

$$\boxed{w^{t+1} = w^t - \gamma \nabla F(\hat{y}, y, w^t)} \tag{5}$$

where $\boldsymbol{\gamma}$ is the learning rate and $\boldsymbol{\nabla F(\hat{y}, y, w^t)}$ the gradient of lost function $\boldsymbol{F}$. The weights update using (Eq. (5)) is knowing as the gradient descent method.

Backpropagation (BP) algorithm performed by gradient descent is widely used to train an MLP. Briefly, this algorithm consists to update connection weights for minimizing the loss function (weight adjustment) by computing error values from the last layer neuron (back-propagation), starting by the output and heading toward the input. BP algorithm is conceptually simple, computationally efficient, and works in most cases [38]. However, the classical BP algorithm is slow and works better to find local minima than global minimum. Modified basic BP algorithms were suggested to overcome this limitation.

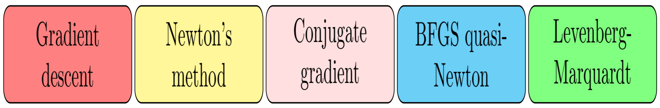


Fig. 3: The most important backpropagation algorithms

LeCun et al. [38] presented an excellent review of various BP algorithms (see Fig. 3) applied to MLP. What emerges from this study is that *Levenberg-Marquardt* algorithm provides best outcomes comparing to other algorithms, and requires the smallest number of training iterations. But, because of his computational demand, *BFGS quasi-Newton* and *Conjugate gradient* are more suitable for large networks. Despite of their successful results, these algorithms require important memory storage and computation, and there still exists an issue with local minima. For a detailed description of BP and its variants, we suggest reading the book of Simon Haykin [37] and the literature review of Ojha et al. [16].

D. Artificial Bee Colony

ABC algorithm is a simulation of honey bee foraging behavior, introduced by [39] is one approach that has been used to find an optimal solution for numerical optimization problems. In our proposed study ABC algorithm is used to find a combination of weights and biases which yield minimization of the loss function (Eq. (4)). The main advantage of this type of metaheuristic its does not use the gradient of the loss function, which means ABC algorithm does not require that the loss function be differential and continuous such as other optimization algorithms (Fig. 3).

1) Concept of Artificial Bee Colony:

As in the minimal model of forage selection of real honey bees, the colony of artificial bees contains three groups of bees: employed, onlookers and scout bees. Initially, all food source positions are discovered by scout bees. Thereafter, the information (i.e, richness, distance and direction from the hive) about the nectar of food sources exploited is shared in the form of a waggle dance from the employed bees to the onlooker bees responsible of evaluating it. Onlookers watch the dances of employed bees and choose a food source, according to the probability proportional to the quality of the food source. Consequently, a good food source position attracts more bees than a bad one. This foraging process is called local search method, as the method of choosing the food source depends on the experience of the employed and onlookers bees. When the food source has been visited fully, the employed bee associated with it abandons it. The employed bee whose food source has been abandoned becomes a scout and starts to search for finding a new food source randomly without using experience. This foraging process is called global search.

2) Formalism:

In ABC algorithm, the position of a food source $\boldsymbol{X_i^t = (x_{i1}^t, \ldots, x_{id}^t, \ldots, x_{iD}^t)}$ at iteration $\boldsymbol{t}$, such as $\boldsymbol{D}$ is the dimension of the loss function (Eq. (4)), represents a possible solution to the problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. $\boldsymbol{i \in \{1, 2, \ldots, SN\}}$ refers to the solution where $\boldsymbol{SN}$ denotes the size of population. ABC was mathematically modeled as follows:

Initially, at iteration $\boldsymbol{t = 0}$, ABC generates a randomly distributed initial population of $\boldsymbol{SN}$ solutions, calculated by the following expression:

$$\boxed{x_{i,j} = x_j^{min} + rand[0, 1] \times \left(x_j^{max} - x_j^{min}\right)} \tag{6}$$

where $\boldsymbol{x_j^{min}}$ and $\boldsymbol{x_j^{max}}$ as lower and upper bounds of the $\boldsymbol{j^{th}}$ dimension, respectively.

After, each iteration $\boldsymbol{t}$ process requires cycle of three steps described below:

1 Number of bees employed is equal to the number of food sources since each employed bee is associated with one and only one source of food. This employed bee combines her memorized positional information of this food source and randomly chooses another neighbor food source according to the following equation:

$$\boxed{v_{i,j} = x_{i,j} + \Phi_{i,j} \times (x_{i,j} - x_{k,j})} \tag{7}$$

where $\boldsymbol{\Phi_{i,j}}$ is a random number between $[-1, 1]$, $\boldsymbol{k \neq i}$ and both are $\in \{1, 2, \ldots, SN\}$

Once the new neighbor solution $\boldsymbol{V_i}$ is generated, a greedy selection is used. If the fitness value of $\boldsymbol{V_i}$ is better than the old food source $\boldsymbol{X_i}$, then $\boldsymbol{X_i}$ is updated with $\boldsymbol{V_i}$; otherwise we keep $\boldsymbol{X_i}$ unchanged.

2 In the same way, the onlooker bees generate a neighbor food source from the old one in her memory (Eq. (7)). Except that the onlooker bees choose their own memorized positional information depending on the probability value associated with that food source. The calculation of the probability value of the i food source is given by following equation:

$$\boxed{P_i = \frac{\mathrm{fit}_i}{\sum_{j=0}^{SN} fit_j}} \tag{8}$$

where $\boldsymbol{fit_i}$ is the fitness value of the solution $\boldsymbol{i}$ which is proportional to the nectar amount of the food sources. This means as seen, the better the solution $\boldsymbol{i}$, the higher the probability of the $\boldsymbol{i^{th}}$ food source is selected.

3 During these two previous steps, when a new solution $\boldsymbol{V_i}$; is generated from a solution $\boldsymbol{X_i}$; does not improve the latter, a counter of visits is incremented for the solution $\boldsymbol{X_i}$. When this counter reaches a predetermined number of visits, the solution is abandoned. Then, the scout bee replaces $\boldsymbol{X_i}$ and discovers a new food source according (Eq. (6)).

In ABC, there are three main control parameters: (1) $\boldsymbol{\Phi_{i,j}}$ is the weight of the difference between memorized positional information and randomly selected food source. This parameter maintains the proper diversity mechanism of the search procedure in ABC. (2) The population size of ABC $\boldsymbol{SN}$, the number of onlooker bees equals the number of employed bees, and where a scout bee is one of the employed bees whose food source is exhausted. From literature, the algorithm produces better results when the population size increases. But from a certain value, any increment does not improve the performance of the ABC algorithm significantly. However, it was suggested that a size of 50-100 can provide an acceptable convergence. (3) The last crucial control parameter is responsible to explore new areas of search space called limit and represents the number of visits. It should be equal $\left(\frac{\text{population size}}{\mathbf{2}}\right) \times \boldsymbol{D}$ [40].

IV. Research Design and Methodology from a Concrete Case

Our executive route and methodology of the CBP is described in Fig. 4. The data comes from Emerging Markets Information Service (EMIS). After data preprocessing (outliers, missing data, imbalanced data, normalization), feature selection serves for dimensionality reduction. Afterwards, our algorithms/models/classifiers are applied to explore the patterns from the transformed data. Finally, the prediction results are evaluated in terms of accuracy for the sake of comparison among different classifiers.

Fig. 4: Pipeline of corporate bankruptcy prediction

A. Data

The dataset is about CBP of Polish firms. The data was collected from (EMIS), which is a database containing information on emerging markets around the world. The source of database includes various information such as financial, political, macroeconomic of firms [41]. Table I reports criteria used to select data for our study.

TABLE I: The process of selection data.

Name	Criterion	Selection
Database	Availability of databases	EMIS
Sector	The largest number of bankrupt firms in the sector compared to other sectors	Manufacturing sector
Forecasting period	Two most chosen horizons	Bankruptcy after one year and after three years
Predictive variables	Used in the integrated models and financial analysis	64 financial ratios were analysed [41]

B. Data preprocessing

Real-world financial data are incomplete, inconsistent, and likely to contain many errors [42]. And as we know, Machine learning algorithms learn from the data provided to it. Therefore, if data quality is poor (i.e. incorrect or incomplete or redundant), so, the resulting algorithm will itself be quite poor, since it supposedly reflects what it is seeing in the data. This is why, it is imperative to preprocess our data before feeding the algorithm by this them. Therefore, data preprocessing is a procedure in which the raw data is cleaned up and transformed to increase the predictive power of models. This includes the detection of missing and outlier values, imbalanced data and a selection of the most relevant financial ratios.

1) Missing and outliers values:

TABLE II: Before and after data preprocessing.

Firms	Messy Data			
	1 year		3 years	
	Before	After	Before	After
Bankrupt	308▲	0	388▲	0
Healthy	2571✗	0	5230✗	0
Total	2879	0	5618	0

[▲: imputation technique KNN algorithm ✗: remove entire rows containing missing values].

Inappropriate handling of missing data may cause false results on classification, especially if the number of missing values is large. The strategy adopted for healthy firms was to remove entire rows containing missing values, and for bankrupt firms to impute with the KNN algorithm; Removing technique was not an option due to the low number of bankrupt firms. The advantage of this algorithm over others (imputation by mean and mode for example) is the imputation of missing values which are influenced only by the most similar cases rather than by all cases. This imputation is more suitable also in case of a large number of missing data [43].

In statistics, an outlier is an observation point that is distant from other observations. The outliers can significantly influence machine learning modelling, as machine learning models are to a great extent sensitive to occurrence of outliers. Therefore, it is appropriate to consider exclusion or replacement of such extreme values from our data not to distort the resulting prediction models. To find and replace significant univariate outliers. Hampel's test wil be conducted (for detailed information about this test, see [44]).

2) Imbalanced data and feature selection:

TABLE III: Before and after data preprocessing.

Firms	Imbalanced Data			
	1 year		3 years	
	Before	After	Before	After
Bankrupt	410	2785▼	495	4601▼
Healthy	5500	2785▼	10008	4601▼
Total	5910	5570	10503	9202

[▼: SMOTE + Tomek Link method].

Classification models perform unsatisfactorily on imbalanced data (see Table II). This issue distorts the model prediction capability with the consequence that models tend to classify badly bankrupt firms in healthy firm (the majority class). Sun et al. [8] presented a comprehensive introduction addressing their advantages and constraints related to this problem. From this literature review, solutions for imbalanced data were categorized in three groups: over-sampling, under-sampling, and hybrid. For our study, (SMOTE + Tomek Link) method [45], [46], [47], a hybrid model was preferred to take advantage of both over-sampling and under-sampling. This method uses first the over-sampling technique, SMOTE, on the minority class, and then ambiguous borderline instances stem from the majority class are removed by T-Link method. A detailed list and classification of data balancing methods are provided by Arafat et al. [47].

Feature selection is the process of selecting a subset of relevant financial ratios related to CBP. Some recent studies have explicitly highlighted the important role of features selection with the conclusion that selection of representative variables increases prediction performance [8], [9]. Generally, there are two types of featuring methods: filter and wrapper. Filter approaches are independent of the base classifier and use human experiential expertise or statistical information of the data set to carry out feature selection. Wrapper approaches, on the other hand, use classification accuracy, or criteria derived from the classifier, to rank the discriminative power of all (or a part) of the possible feature subsets and select the subset that produces the best performance [8]. Examples of feature selection/extraction are t-test, correlation matrix, PCA or FA for the filter approach, and stepwise regression like stepwise MDA or stepwise Logit for the wrappe approach. Over the past few years, new methods have been proposed to address feature selection in presence of thousands variables. Recursive feature elimination is one of the prominent feature selection methods. Hence, in this paper, we applied the Random Forest-Recursive Feature Elimination (RF-RFE) algorithm to identify predictive variables.

V. Results and Discussion

A. Model selection

Data were subdivided randomly into training and testing samples. The training set was used to establish our model and the testing set was used to evaluate the model and its robustness. Performance of our model, the MLP associated to ABC algorithm called ABCNN was compared to MDA and to the MLP associated to the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm called BPNN. All models were implemented with the *Scikit-learn* machine learning library in Python programming language. *Linear Discriminant Analysis* and *MLP classifier* are used to implement MDA and BPNN, respectively. For ABCNN, we wrote our own program based on a MLP classifier.

Moreover, our models require different hyper-parameters and tuning them is a crucial part of model selection in order to prevent over-fitting. We applied a grid-search cross-validation algorithm using *GridSearchCV* of the *Sklearn* package, which uses multiple training and test sets to measure the influence of given hyper-parameter

B. Performance metric

TABLE IV: Confusion matrix for a two-class classification problem.

Actual	Predicted	
	Bankrupt	Healthy
Bankrupt	True Positive (TP)	False Negative (FN)
Healthy	False Positive (FP)	True Negative (TN)

Usually, the result of CBP is considered as a binary value: bankrupt or healthy. Accordingly, the outcome of a classifier can be represented as a $2D$ contingency matrix shown in Table IV, where: TP is True Positive, that is to say, bankrupt firms are classified correctly, TN is True Negative, that is to say, healthy firms are classified correctly, FN is False Negative, that is to say, bankrupt firms are classified wrongly and FP is False Positive, that is to say, healthy firms are classified wrongly. The accuracy metric was calculated to evaluate our models:

Three performance metrics were calculated to evaluate models: accuracy (the number of firms which are correctly classified and divided by the total number of firms in the test set), type I error (misclassification of a bankrupt firm as healthy) and type II error (misclassification of a healthy as bankrupt). In CBP context, there is a preference for lower type I error because this translates into losses for lenders, whereas type II error is the threshold for gain [48].

$$\boxed{\text{Accuracy} = \frac{TN+TP}{TN+FP+FN+TN}}$$

$$\boxed{\text{Type I} = 1 - \frac{TP}{TP+FN}}$$

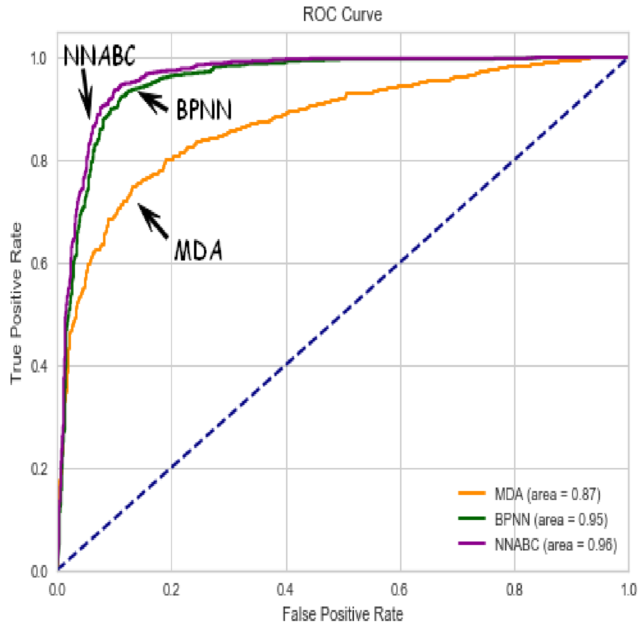
$$\boxed{\text{Type II} = 1 - \frac{TN}{TN+FP}}$$

C. Empirical results

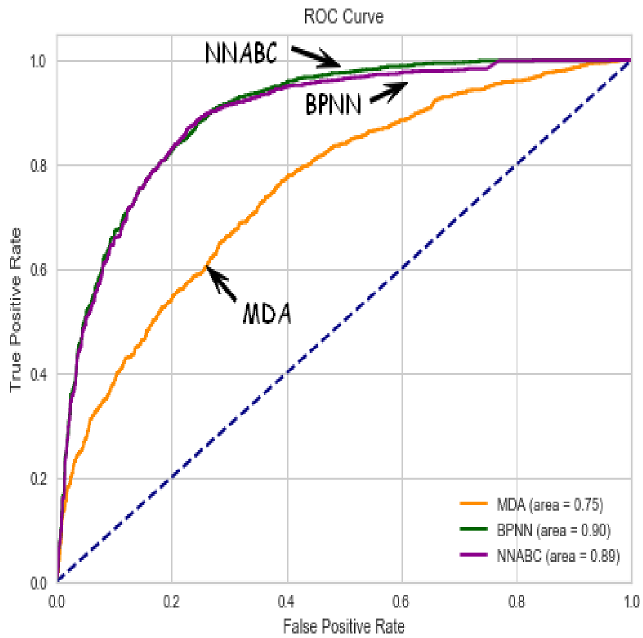
Table V reports performance metrics (type I and type II errors, accuracy and AUC) for all models from the testing data set for the two forecasting period of one year before bankruptcy and three years before bankruptcy.

TABLE V: The results in (%) of prediction models on the testing data set.

Models	Type I Error	Type II Error	Accuracy	AUC
	One year before the bankrupt			
MDA	14.54	24.47	80.67	87.18
BPNN	9.12	8.07	91.38	95.36
ABCNN	6.81	9.19	92.04	96.26
	Three years before the bankrupt			
MDA	23.26	42.15	67.22	75.38
BPNN	18.81	19.06	81.05	89.52
ABCNN	17.28	20.79	80.94	88.91



(a) One year before the bankrupt



(b) Three years before the bankrupt

Fig. 5: Roc curve of all models.

D. Discussion

On the testing data set, the ANN models outperform significantly the MDA model that belongs to traditional statistical techniques. The linear structure applied by statistical techniques to separate two classes (bankrupt face to non-bankrupt) causes this weak performance. Thus, non-linear models are necessary when more variables are used to predict bankruptcy. The table (V) also shows that the predictive power of the three models has been orienting downwards by three years compared to one year, indicating that more financial distress events are imminent, more than the overall accuracy achieved by all the algorithms is high. We also note that the

MDA and BPNN models are largely optimal in terms of calculating time when we implement the training of our model ABCNN.

When using data from one year before the bankrupt, our model ABCNN had the lower type I error (6.81%) and the best total accuracy rate (92.04%), followed by BPNN with a type I error of 9.12% and a total accuracy rate of 91.38% of. But, regarding type II error, ABCNN was of 9.19% so, higher than BPNN with 8.07%. The ANN model's errors and prediction rates are all better than those of the MDA model. MDA performs acceptably in misclassification of solvent companies (14.51%), but it is a poor model for classifying bankruptcy firms (24.47%). The result of these different models mentioned below before three years are in the same order for errors I and II with 17.28% and 20.79% respectively for ABCNN and 18.81% and 19.06% for BPNN. The only difference was in accuracy value. BPNN was slightly lower than ABCNN with a value of 80.94% for ABCNN and 81.05% for BPNN.

Fig. 5 shows the ROC curve and AUC of each model. The ANN models show significant superiority over MDA. However, it is difficult to confirm a single preferred model from the curves between BPNN and ABCNN.

VI. Conclusion

corporate bankruptcy prediction (CBP) is a class of an interesting and important problem. A better understanding of these causes will have tremendous financial and managerial consequences. While traditional statistical techniques work well for some situations, they may fail miserably when statistical assumptions are not met. Prediction models from machine learning such as ANN are promising alternative tools that should be given much consideration when solving real problems like CBP.

In this paper, we proposed a hybrid model to predict corporate bankruptcy. We processed in three stages. In the first stage, we applied a data preprocessing: (1) Missing data were imputed with meaningful values by using imputation technique (KNN algorithm) or by removing the entire observation containing missing values, (2) predictive variables were adjusted due to outliers by applying Hampel's test, (3) imbalanced data were corrected with the SMOTE + Tomek Link method (4) and a feature selection technique (RF-RFE) was used to extract the most important features. In the second stage, we examined predictive ability of three different models, namely MDA, BPNN, and the hybrid model based on an advanced Particle Swarm Optimization algorithm (PSONN), to seek the optimal parameters of the MLP model. In the last stage, an In-sample performance validation was used to enhance the generalization of the model.

The performance was evaluated not only in terms of prediction accuracy but also the type I and type II errors and AUC-score were used. The results of our experiments indicate better performance of the hybrid ABCNN, followed by the traditional back-propagation BPNN, with their flexible non-linear modeling capability, over other multivariate statistical methods. Again, the results pointed out that when we try and identify bankruptcies within one year before the bankruptcy, performance is better than that of three years.

In a future work, different modification can be made: (1) Apply deep neural networks (DNN) model to our problem. Indeed, DNN can produce more useful results than other models. In particular, deep learning can detect and exploit invisible interactions in data. (2) Incorporate macroeconomic variables with financial ratios for the construction of predictive models in order to improve their prediction. (3) using a parallel processing architecture that increases the computing performance of the system by leveraging the power of graphics processors (GPUs). This is because GPUs offer capabilities for parallelism that are not found in general purpose processors which happen to be a good match for operations such as matrix calculations and large-scale hashing. This can be used to considerably reduce training time of our model PSONN, which increases the number of experiments and iterations that can be run while tuning a model. (4) Consider another metaheuristic method on the MLP model or look at whether a combination of two or more metaheuristic algorithms may offer a more accurate solution than a single one.

References

[1] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International journal of information management*, vol. 35, no. 2, pp. 137–144, 2015.

[2] A. Mochón, D. Quintana, Y. Sáez, and P. Isasi, "Soft computing techniques applied to finance," *Applied Intelligence*, vol. 29, no. 2, pp. 111–115, 2008.

[3] A. Verikas, Z. Kalsyte, M. Bacauskiene, and A. Gelzinis, "Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: a survey," *Soft Computing*, vol. 14, no. 9, pp. 995–1010, 2010.

[4] E. I. Altman, "Corporate distress prediction models in a turbulent economic and basel ii environment," 2002.

[5] A. I. Dimitras, S. H. Zanakis, and C. Zopounidis, "A survey of business failures with an emphasis on prediction methods and industrial applications," *European Journal of Operational Research*, vol. 90, no. 3, pp. 487–513, 1996.

[6] J. L. Bellovary, D. E. Giacomino, and M. D. Akers, "A review of bankruptcy prediction studies: 1930 to present," *Journal of Financial education*, pp. 1–42, 2007.

[7] P. R. Kumar and V. Ravi, "Bankruptcy prediction in banks and firms via statistical and intelligent techniques–a review," *European journal of operational research*, vol. 180, no. 1, pp. 1–28, 2007.

[8] J. Sun, H. Li, Q.-H. Huang, and K.-Y. He, "Predicting financial distress and corporate failure: A review from the state-of-the-art definitions, modeling, sampling, and featuring approaches," *Knowledge-Based Systems*, vol. 57, pp. 41–56, 2014.

[9] N. Chen, B. Ribeiro, and A. Chen, "Financial credit risk assessment: a recent review," *Artificial Intelligence Review*, vol. 45, no. 1, pp. 1–23, 2016.

[10] A. Brabazon, J. Dang, I. Dempsey, M. O'Neill, and D. Edelman, "Natural computing in finance–a review," *Handbook of Natural Computing*, pp. 1707–1735, 2012.

[11] T. G. Calderon and J. J. Cheh, "A roadmap for future neural networks research in auditing and risk assessment," *International Journal of Accounting Information Systems*, vol. 3, no. 4, pp. 203–236, 2002.

[12] M. Tkáč and R. Verner, "Artificial neural networks in business: Two decades of research," *Applied Soft Computing*, vol. 38, pp. 788–804, 2016.

[13] H. A. Alaka, L. O. Oyedele, H. A. Owolabi, V. Kumar, S. O. Ajayi, O. O. Akinade, and M. Bilal, "Systematic review of bankruptcy prediction models: Towards a framework for tool selection," *ESWA*, vol. 94, pp. 164–184, 2018.

[14] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, 2014.

[15] T.-H. Lin, "A cross model study of corporate financial distress prediction in taiwan: Multiple discriminant analysis, logit, probit and neural networks models," *Neurocomputing*, vol. 72, no. 16-18, pp. 3507–3516, 2009.

[16] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 60, pp. 97–116, 2017.

[17] C.-F. Wang and Y.-H. Zhang, "An improved artificial bee colony algorithm for solving optimization problems," *IAENG International Journal of Computer Science*, vol. 43, no. 3, pp. 336–343, 2016.

[18] W. H. Beaver, "Financial ratios as predictors of failure," *Journal of accounting research*, pp. 71–111, 1966.

[19] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The journal of finance*, vol. 23, no. 4, pp. 589–609, 1968.

[20] E. B. Deakin, "Distributions of financial accounting ratios: some empirical evidence," *The Accounting Review*, vol. 51, no. 1, pp. 90–96, 1976.

[21] R. H. Jackson and A. Wood, "The performance of insolvency prediction and credit risk models in the uk: A comparative study," *The British Accounting Review*, vol. 45, no. 3, pp. 183–202, 2013.

[22] J. A. Ohlson, "Financial ratios and the probabilistic prediction of bankruptcy," *Journal of accounting research*, pp. 109–131, 1980.

[23] J. Begley, J. Ming, and S. Watts, "Bankruptcy classification errors in the 1980s: An empirical analysis of altman's and ohlson's models," *Review of accounting Studies*, vol. 1, no. 4, pp. 267–284, 1996.

[24] A. W. Lo, "Logit versus discriminant analysis: A specification test and application to corporate bankruptcies," *Journal of econometrics*, vol. 31, no. 2, pp. 151–178, 1986.

[25] M. Adnan Aziz and H. A. Dar, "Predicting corporate bankruptcy: where we stand?" *Corporate Governance: The international journal of business in society*, vol. 6, no. 1, pp. 18–33, 2006.

[26] M. D. Odom and R. Sharda, "A neural network model for bankruptcy prediction," in *1990 IJCNN International Joint Conference on neural networks.* IEEE, 1990, pp. 163–168.

[27] E. I. Altman, G. Marco, and F. Varetto, "Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the italian experience)," *Journal of banking & finance*, vol. 18, no. 3, pp. 505–529, 1994.

[28] P. Pompe and A. Feelders, "Using machine learning, neural networks, and statistics to predict corporate bankruptcy," *Computer-Aided Civil and Infrastructure Engineering*, vol. 12, no. 4, pp. 267–276, 1997.

[29] W.-S. Chen and Y.-K. Du, "Using neural networks and data mining techniques for the financial distress prediction model," *Expert Systems with Applications*, vol. 36, no. 2, pp. 4075–4086, 2009.

[30] T. Abdelwahed and E. M. Amir, "New evolutionary bankruptcy forecasting model based on genetic algorithms and neural networks," in *Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on.* IEEE, 2005, pp. 5–pp.

[31] Y.-C. Hu, "Bankruptcy prediction using electre-based single-layer perceptron," *Neurocomputing*, vol. 72, no. 13-15, pp. 3150–3157, 2009.

[32] N. Chauhan, V. Ravi, and D. K. Chandra, "Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7659–7665, 2009.

[33] P. Ravisankar, V. Ravi, and I. Bose, "Failure prediction of dotcom companies using neural network–genetic programming hybrids," *Information Sciences*, vol. 180, no. 8, pp. 1257–1267, 2010.

[34] A. Soler-Dominguez, A. A. Juan, and R. Kizys, "A survey on financial applications of metaheuristics," *ACM Computing Surveys (CSUR)*, vol. 50, no. 1, p. 15, 2017.

[35] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (abc) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.

[36] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[37] S. Haykin, *Neural Networks and Learning Machines, 3/E.* Pearson Education India, 2010.

[38] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade.* Springer, 2012, pp. 9–48.

[39] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer , Tech. Rep., 2005.

[40] J. C. Bansal, H. Sharma, and S. S. Jadon, "Artificial bee colony algorithm: a survey," *International Journal of Advanced Intelligence Paradigms*, vol. 5, no. 1-2, pp. 123–159, 2013.

[41] M. Zieba, S. K. Tomczak, and J. M. Tomczak, "Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction," *Expert Systems with Applications*, vol. 58, pp. 93–101, 2016.

[42] H. Son, C. Hyun, D. Phan, and H. Hwang, "Data analytic approach for bankruptcy prediction," *Expert Systems with Applications*, vol. 138, p. 112816, 2019.

[43] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review," *Neural Computing and Applications*, vol. 19, no. 2, pp. 263–282, 2010.

[44] M. Mihalovic, "Performance comparison of multiple discriminant analysis and logit models in bankruptcy prediction," *Economics & Sociology*, vol. 9, no. 4, p. 101, 2016.

[45] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[46] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[47] M. Arafat, S. Hoque, S. Xu, and D. M. Farid, "Machine learning for mining imbalanced data," *IAENG International Journal of Computer Science*, vol. 46, no. 2, pp. 332–348, 2019.

[48] F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications*, vol. 83, pp. 405–417, 2017.