




Article

Cost-Sensitive Metaheuristic Optimization-Based Neural Network with Ensemble Learning for Financial Distress Prediction

Salah Al-Deen Safi ¹, Pedro A. Castillo ¹ and Hossam Faris ^{2,3,*}

¹ Department of Computer Architecture and Technology, ETSIT-CITIC, University of Granada, 18011 Granada, Spain; sasafi@correo.ugr.es (S.A.-D.S.); pacv@ugr.es (P.A.C.)

² King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan

³ Research Centre for Information and Communications Technologies of the University of Granada (CITIC-UGR), University of Granada, 18011 Granada, Spain

* Correspondence: hossam.faris@ju.edu.jo or inv.hfaris@ugr.es

Abstract: Financial distress prediction is crucial in the financial domain because of its implications for banks, businesses, and corporations. Serious financial losses may occur because of poor financial distress prediction. As a result, significant efforts have been made to develop prediction models that can assist decision-makers to anticipate events before they occur and avoid bankruptcy, thereby helping to improve the quality of such tasks. Because of the usual highly imbalanced distribution of data, financial distress prediction is a challenging task. Hence, a wide range of methods and algorithms have been developed over recent decades to address the classification of imbalanced datasets. Metaheuristic optimization-based artificial neural networks have shown exciting results in a variety of applications, as well as classification problems. However, less consideration has been paid to using a cost sensitivity fitness function in metaheuristic optimization-based artificial neural networks to solve the financial distress prediction problem. In this work, we propose ENS_PSONN_{cost} and ENS_CSONN_{cost}: metaheuristic optimization-based artificial neural networks that utilize a particle swarm optimizer and a competitive swarm optimizer and five cost sensitivity fitness functions as the base learners in a majority voting ensemble learning paradigm. Three extremely imbalanced datasets from Spanish, Taiwanese, and Polish companies were considered to avoid dataset bias. The results showed significant improvements in the g-mean (the geometric mean of sensitivity and specificity) metric and the F1 score (the harmonic mean of precision and sensitivity) while maintaining adequately high accuracy.

Keywords: financial distress; cost-sensitive; ensemble learning; imbalanced classification; metaheuristic; neural networks



Citation: Safi, S.A.-D.; Castillo, P.A.; Faris, H. Cost-Sensitive Metaheuristic Optimization-Based Neural Network with Ensemble Learning for Financial Distress Prediction. *Appl. Sci.* **2022**, *12*, 6918. <https://doi.org/10.3390/app12146918>

Received: 23 May 2022

Accepted: 5 July 2022

Published: 8 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The phrases bankruptcy and insolvency are frequently used interchangeably in the literature [1]. Bankruptcy is a legal financial procedure in which an individual or an organization declares that they are unable to pay their obligations. As an outcome of this legal position, the debtor's assets are liquidated to repay some of their debts, while the remainder of their debts are ignored [2]. Insolvency is defined as the failure to pay or the scenario in which a corporation, another legal entity or an individual cannot meet their financial commitments by the maturity date [1]. Hence, financial distress (i.e., bankruptcy or insolvency) prediction is a critical tool within the financial industry that serves as an aid for making appropriate business decisions [3]. The successful forecasting of this challenge provides a broader view of the business's health and assists decision-makers in anticipating occurrences before they happen.

As a result, there has been a significant effort in the literature to construct statistics- and artificial intelligence-based models that can accurately estimate a company's financial

state. In general, the previous evaluations of the company's condition, whether it has had financial distress or not, are examined as a binary classification problem from a machine learning perspective.

The challenge in dealing with financial distress datasets is that they are highly imbalanced. When there are significantly more samples from one class than other classes, the dataset is said to be imbalanced. Due to the effects of the majority class on the traditional training criteria, classifiers may have a high accuracy for the majority class but an extremely low accuracy for the minority class(es). The goal of most original classification algorithms is to reduce the error rate or the percentage of erroneous class label predictions [4].

There are two primary techniques for dealing with imbalanced datasets: at the data level, by resizing the training datasets (undersampling or oversampling), and at the algorithmic level, by using cost-sensitive classifiers [4]. In this work, we evaluated the algorithmic-level approach using a metaheuristic optimization-based artificial neural network (MHOANN) as our classifier, which was based on a particle swarm optimizer (PSO) [5] and a competitive swarm optimizer (CSO) [6] with a cost sensitivity fitness function. We then improved the capabilities of our model using homogeneous majority voting ensemble learning.

Evolutionary neural networks (ENNs) [7–12] are a subset of neural networks (NNs) in which evolution is a key type of adaptation, in addition to learning. Connection weight training, architectural design, learning rule adaption, input feature selection, connection weight initialization, rule extraction from NNs, and other activities are performed using evolutionary algorithms (EAs) [13].

MHOANNs are a subset of artificial neural networks (ANNs) in which the selection of weights and biases is performed using metaheuristic optimization algorithms [14]. Inspired by the collective behavior of social animals, swarm-based algorithms have been developed into a strong family of optimization approaches. The collection of potential solutions to the optimization issue is characterized in a PSO as a swarm of particles that flow across the parameter space, establishing trajectories that are driven by their own and their neighbors' best performances [15]. On the other hand, a CSO is a recent variation of a PSO in which a pairwise competition mechanism is implemented that causes the losing particle to learn from the winner and update its location [6].

This paper proposes using a cost-sensitive MHOANN to improve the prediction of minor classes in a financial distress dataset and then applying majority voting ensemble learning to create a strong learner out of several weak learners. The cost-sensitive component is used to improve the prediction of the minority classes, whereas the majority voting attempts to mitigate the negative influences of cost on the prediction of the majority class. Applying a cost sensitivity fitness function in an ensemble learning paradigm is different from existing cost-sensitive methods because it reduces the effects of the bias toward the minority classes, which is caused by the costs that are associated with the misclassification of minor class instances in the classical cost-sensitive methods. Moreover, the evolutionary nature of the utilized metaheuristic algorithms provides the accuracy and diversity that are required by ensemble learning to achieve a high prediction capability that exceeds the prediction capability of a single learner. The reason for selecting a PSO and a CSO as the optimization techniques in this work was that, compared to other metaheuristic algorithms, a PSO requires a small number of parameters and a correspondingly lower number of iterations [16]. On the other hand, a CSO is a relatively recent variation of a PSO that was designed to be used for large-scale optimization problems because half of the population is updated during each iteration [17].

To validate this, we used three different datasets from Spanish, Taiwanese, and Polish companies to evaluate the proposed method. The dataset of Spanish companies was considered very challenging, owing to its highly imbalanced distribution in which insolvency cases only formed 2% of the whole sample. In the datasets of Taiwanese companies and Polish companies, insolvency cases formed approximately 3% and 2% of the samples, respectively.

When applying the cost sensitivity fitness function, we noticed a significant improvement in the number of true positive (TP) predictions but an increase in the number of false positive (FP) predictions. To overcome this problem, we used majority voting ensemble learning to maintain the high TP prediction rate and reduce the number of FP predictions. This work proposes a framework for solving financial distress prediction problems for extremely imbalanced datasets. The framework uses a cost sensitivity fitness function to reduce the number of FN predictions. Moreover, it relies on ensemble learning to compensate for the faults of individual learners and reduce the number of FP predictions. All of the steps in the framework are internal and do not affect the data; hence, it can be a helpful tool in financial distress prediction. To the best of our knowledge, our work is the first to combine a cost-sensitive MHOANN with majority voting ensemble learning for financial distress prediction. Another contribution of this work is the comparison of a PSO and CSO as optimization techniques for the MHOANN.

The remainder of this paper is organized as follows. In the following section, we review the related works. Then, in Section 3, we explain the optimization algorithms that were used in our study. In Section 4, we describe the considered datasets. Section 5 describes the proposed method and in Section 6, we describe the evaluation metrics that were used. The experiments that were conducted and the obtained results are explained in Section 7. Finally, the conclusions and future work are discussed in Section 8.

2. Related Works

In the literature, much research has been conducted on examining the problem of imbalanced datasets using a variety of methods and approaches in different combinations. For example, a modified version of a support vector machine (SVM) that was based on density weight was proposed in [18] to tackle the binary class imbalance classification problem. Experimental analyses were performed on certain intriguing imbalanced artificial and real-world datasets and their performances were measured using the metrics of the area under the curve and the geometric mean. The results were compared to those from an SVM, a least squares SVM, a fuzzy SVM, an improved fuzzy least squares SVM, a fuzzy SVM that was based on affinity and class probability, and an entropy-based fuzzy least squares SVM. The similar or better generalization results indicated the efficacy and applicability of the proposed algorithms. Deep learning (DL) methods have also been considered to overcome the class imbalance challenge. In [19], the authors presented a novel comparison between three different DL methods: a deep belief network (DBN), long-short term memory (LSTM), and a multilayer perceptron model (MLP). They also compared five ensemble classifiers financial distress prediction: XGBoost, SVM, K-nearest neighbor (KNN), and AdaBoost. A new selective oversampling approach (SOA) that uses an outlier identification technique to separate the most representative samples from the minority classes and then uses these samples for synthetic oversampling was proposed in [20]. Their experiments demonstrated that the suggested method outperformed two state-of-the-art oversampling strategies: synthetic minority oversampling and adaptive synthetic sampling.

Moreover, using cost-sensitive learning to solve the imbalanced classification problem has also been very popular in the literature. Robust cost-sensitive classifiers have been constructed by changing the objective functions of well-known algorithms, including logistic regression, decision trees, extreme gradient boosting, and random forests, which can then be then utilized to predict medical diagnoses effectively, as proposed in [21]. Furthermore, the cost-sensitive approaches outperformed the standard algorithms, according to the findings of those experiments. In another study, the authors used decision trees as a boosting method to improve business failure prediction performance. A weighted objective function, weighted cross-entropy, was incorporated into the boosted tree architecture to overcome the class imbalance issue in the business failure datasets, making the weighted XGBoost a cost-sensitive business failure prediction model [22].

Furthermore, using evolutionary algorithms to train artificial neural networks (ANNs) has been very popular since the 1980s. The use of the genetic algorithm (GA) to train an

ANN for image classification was discussed in [23]. Additionally, using metaheuristic algorithms to train ANNs to manage the disadvantages of gradient-based methods, particularly backpropagation techniques, has also been extensively researched. During the early 2000s, numerous studies focused on the use of metaheuristic algorithms in neural network training for binary classification tasks, such as financial distress prediction. Metaheuristic approaches were proven to perform better than gradient-based algorithms in [24]. The effects of fitness functions on MHOANN learning when dealing with imbalanced datasets was also discussed in [25]. A PSO algorithm was used to optimize the weights and biases in a neural network architecture to predict bankruptcy among Indian firms in [26]. An artificial neural network that was trained by a metaheuristic artificial bee colony (ABC) algorithm was proposed in [27]. The model was used for corporate bankruptcy prediction and then the proposed method was compared to the multiple discriminant analysis (MDA) model and an ANN that was trained by the most common learning algorithm (backpropagation (BPNN)). Their experimental results showed that the ABC algorithm could be used as an optimization algorithm for artificial neural networks to predict potential corporate bankruptcy. In another study, the authors conducted a comprehensive benchmark of 15 population-based optimization algorithms that were used to train ANNs. Their obtained experimental results using a challenging set of eight classification problems showed that the PSO yielded the best performance out of the other population-based metaheuristic algorithms [28].

On the other hand, ensemble classifiers have been effectively employed in credit scoring and the forecasting of company insolvency in recent years. For example, a cost-sensitive neural network ensemble for credit scoring was proposed in [29]. The suggested method outperformed the benchmark individual and ensemble methods, as evidenced by the comparative results. In another study, an ensemble classifier-based scoring model for the early prediction of the risk of bankruptcy among Polish businesses was proposed in [30]. Their results proved that using ensemble classifiers could be very powerful for foreseeing bankruptcy. Additionally, an ensemble classifier for classifying binary, non-stationary, and imbalanced data streams in which the Hellinger distance was used to prune the ensemble was implemented in [31]. The Hellinger distance weighted ensemble approach was thoroughly tested using many imbalanced data streams and the results proved the usefulness of the method.

MHOANN, cost-sensitive learning, and ensemble learning have shown promising results for classification problems. However, little attention has been paid to the effects of combining the cost sensitivity fitness function within an MHOANN with ensemble learning for financial distress prediction.

3. Background

Optimization algorithms are methods that are used to update the weights and biases in an ANN to overcome the disadvantages of conventional training algorithms. This work utilized state-of-the-art PSO and CSO (a recent variant of a PSO) metaheuristic algorithms as optimization techniques for our ANN.

3.1. Particle Swarm Optimization (PSO)

This population-based optimization technique was inspired by the movement of flocks of birds and schools of fish. It uses social interactions to find the best solutions. The swarm is randomly initialized with a population of solutions that are called particles (or agents). The search for the optimal solution is repeated in iterations, during which these particles move around the search space according to a mathematical formula that governs the position and velocity of the particles. The motion of each particle is affected by the best solution that has been achieved so far by that particular particle and is guided to the known best positions within the search space, which are adjusted when better positions are discovered by other particles in the swarm. Hence, the swarm moves toward the optimal solution [15].

In this study, the velocity was modeled mathematically, as stated in Equation (1), where $v_{id}(t)$ is the velocity of the particle i in dimension $d = 1, \dots, n_p$ at time step t , w is the inertia weight, r_1 and r_2 are random values $\in [0, 1]$ from a uniform distribution, c_1 and c_2 are positive acceleration constants, $p_{id}(t)$ is the best position that the particle i has visited since the first time step with d dimensions at time t , and $g_d(t)$ is the best global particle position. The position was also modeled mathematically, as stated in Equation (2), where $x_{id}(t)$ is the position of the particle and $v_{id}(t + 1)$ is the velocity of the particle i in dimension d at time step $t + 1$ [32].

$$v_{id}(t + 1) = w \cdot v_{id} + r_1 c_1 \cdot [p_{id}(t) - x_{id}(t)] + r_2 c_2 \cdot [g_d(t) - x_{id}(t)] \tag{1}$$

$$x_{id}(t + 1) = x_{id}(t) + v_{id}(t + 1) \tag{2}$$

3.2. Competitive Swarm Optimizer (CSO)

This is a method that is based on a PSO but is significantly different. In a CSO, neither the particle’s personal best position nor the global best position (or the neighborhood best positions) is used to update the particles. Instead, a pairwise competition mechanism is implemented in which the losing particle learns from the winner and updates its location. Despite its algorithmic simplicity, CSOs outperform the latest metaheuristic algorithms in terms of overall performance [6].

In our CSO, we had $P(t)$, which comprised a swarm of m particles, where m is the size of the swarm and t is the index of the generation. Each particle represented a candidate solution for the optimization problem. The CSO compared two particles that were randomly picked from $P(t)$ in each generation until all particles had competed in at least one competition, providing that the swarm size was an even number. The comparison was made by calculating the fitness of each particle. The particle with the better fitness was considered the winner and was passed directly to the next generation $P(t + 1)$, while the particle that lost the competition was passed to the next generation after learning from the winner. The velocity of the losing particle was updated using Equation (3), where, $x_{w,i}(t)$ is the position of the winning particle in the i -th round of competition in generation t , $x_{l,i}(t)$ is the position of the losing particle in the i -th round of competition in generation t , $v_{w,i}(t)$ is the velocity of the winning particle in the i -th round of competition in generation t , $v_{l,i}(t)$ is the velocity of the losing particle in the i -th round of competition in generation t , $i = 1, 2, \dots, m/2$, m is the population size, $r_1(i, t)$, $r_2(i, t)$ and $r_3(i, t) \in [0, 1]$ are three vectors that were randomly generated after the i -th competition and learning process in generation t , $\bar{x}(t)$ is the mean position value of all particles (which can be regarded as the center of the swarm in generation t), and φ is the parameter that controlled the influences or effects of $\bar{x}(t)$. Then, the position of the losing particle was updated using the newly calculated velocity, according to Equation (4) [6].

$$v_{l,i}(t + 1) = r_1(i, t)v_{l,i}(t) + r_2(i, t)(x_{w,i}(t) - x_{l,i}(t)) + \varphi r_3(i, t)(\bar{x}(t) - x_{l,i}(t)) \tag{3}$$

$$x_{l,i}(t + 1) = x_{l,i}(t) + v_{l,i}(t + 1) \tag{4}$$

4. The Considered Datasets

As previously indicated, three different datasets were selected to verify the effectiveness of the proposed method. While the independent variables and the number of independent variables varied per dataset, forecasting the financial distress of companies was treated as a classification problem in this work and the effectiveness of the proposed method was validated separately for each dataset. The following is a brief description of each dataset.

4.1. Dataset of Spanish Companies

This dataset was for Spanish companies, from which we considered several financial and non-financial features. We considered the dependent variable of bankruptcy as the class for each record or sample and we aimed to classify the instances according to class. The dependent variable was insolvency, which corresponded to the existence of continued losses over three years [33].

This dataset was extracted from the Infotel database (which was bought from <http://infotel.es>, accessed on 1 May 2017). As a result, we had data from 470 businesses that were gathered over six years (from 1998 to 2003). There were 2860 samples in all, with 62 corresponding to insolvent companies, meaning that insolvency cases only formed 2% of the whole sample.

Initially, each row of the dataset had 37 independent variables and 1 dependent variable (bankruptcy). A prior effort by the authors in [33] changed this list by removing unnecessary variables (i.e., those without significance, for instance, internal database firm codes), resulting in 33 independent variables. So, every record in the dataset that was used in this work had 33 features, which comprised a mix of financial indicators and non-financial indicators. Each feature had either a qualitative (categorical) or a quantitative (numerical) value. Table 1 shows the independent variables after removing the unnecessary variables, as well as their type and description. The size of the firm, the kind of company, provincial code (i.e., where the company is situated), and the auditor's judgments were among the non-financial data that had categorical value. Usually, the size of the firm is a number, but in this dataset, it was either small, medium or large, based on the size of the company. Moreover, in this work, we used all 33 features without applying feature selection because, as pointed out by [34], adding a feature selection step would not improve the results.

Table 1. The independent variables of the dataset of Spanish companies (financial and non-financial).

Financial Variables	Description	Type
Debt Structure	Long-term Liabilities/Current Liabilities	Real
Debt Amount	Interest Amount/Total Liabilities	Real
Debt-Paying Ability	Operating Cash Flow/Total Liabilities	Real
Debt Ratio	Total Assets/Total Liabilities	Real
Working Capital	Working Capital/Total Assets	Real
Warranty	Financial Warranties	Real
Operating Income Margin	Operating Income/Net Sales	Real
Returns on Operating Assets	Operating Income/Average Operating Assets	Real
Returns on Equity	Net Income/Average Total Equity	Real
Returns on Assets	Net Income/Average Total Assets	Real
Stock Turnover	Cost of Sales/Average Inventory	Real
Asset Turnover	Net Sales/Average Total Assets	Real
Receivables Turnover	Net Sales/Average Receivables	Real
Asset Rotation	Asset Allocation Decisions	Real
Financial Solvency	Current Assets/Current Liabilities	Real
Acid Test	(Cash Equivalents + Marketable Securities + Net Receivables)/Current Liabilities	Real
Non-Financial Variables	Description	Type
Year	Corresponding to the sample	Integer
Size	Small, medium or large	Categorical
Number of Employees		Integer
Age of Company		Integer
Type of Company	Public company, limited liability company or other	Categorical

Table 1. Cont.

Non-Financial Variables	Description	Type
Linked to Group?	Is the company part of a holding company?	Binary
Number of Partners		Integer
Provincial Code	Postal code for the location of the company	Categorical
Number of Changes of Location		Integer
Delay	Has the company submitted its annual accounts on time?	Binary
Historic Number of Judicial Incidences	Number of judicial instances since the company was created	Integer
Number of Judicial Incidences Last Year	Number of judicial incidences in the last year	Integer
Historic Amount of Money Spent on Judicial Incidences	How much money has the company spent on judicial incidences since it was created?	Real
Amount of Money Spent on Judicial Incidences Last Year	How much money has the company spent on judicial incidences in the last year?	Real
Historic Number of Serious Incidences (e.g., strikes, accidents, etc.)		Integer
Audited?	Has the company been audited?	Binary
Auditor's Judgments	Favorable, exceptional or unfavorable	Categorical

4.2. Dataset of Taiwanese Companies

This dataset was compiled from 10 years (1999–2009) of records from the Taiwan Economic Journal and comprised 6819 entries in total, with 6599 records relating to non-bankrupt firms (97%) and the remainder representing bankrupt firms (220 records), meaning that bankruptcy cases formed approximately 3% of the whole sample. The dataset had 95 financial characteristics. However, the firms in this dataset were chosen based on two criteria: the company's information had to be accessible for three years (so a decision on its financial state could occur) and the size of the firm had to measure up to a sufficient number of firms for comparison. The judgments concerning each firm's financial standing were mostly based on the trading regulations of the stock exchange in Taiwan. Additional information can be found in [35].

4.3. Dataset of Polish Companies

This dataset contained information about the likelihood of a Polish company becoming bankrupt. The information was gathered from the Developing Markets Information Service (EMIS), which is a global collection of information on emerging markets. The insolvent firms were studied from 2007 to 2012, while the enterprises that were still running were assessed from 2007 to 2013. This dataset was also extremely imbalanced, with the number of insolvent companies (203) forming around 2% of the whole sample, which contained around 10,000 instances. The dataset had 64 numerical financial characteristics with no category values. More information about this dataset can be found in [36] and the dataset itself can be downloaded from the Kaggle ML community website (<https://www.kaggle.com/competitions/companies-bankruptcy-forecast/data>, accessed on 28 June 2022).

5. The Proposed Method

This section presents the proposed method for classifying insolvent companies using an MHOANN with a PSO and a CSO as the optimization algorithms and a cost sensitiv-

ity fitness function within a homogeneous majority voting ensemble learning paradigm (ENS_PSONN_{cost} and ENS_CSONN_{cost}). The system architecture of the MHOANN with the embedded cost sensitivity fitness function is illustrated in Figure 1. Furthermore, the proposed architecture for the MHOANN in the majority voting ensemble learning paradigm is shown in Figure 2.

First, we discuss the ANN as a classifier and then we explain how the optimizers (PSO and CSO) were used to set the weights and biases of the ANN, the use of fitness functions to obtain the best solutions, and finally, how all of that fit within a majority voting ensemble learning paradigm. An illustration of the proposed method is presented in Figure 3.

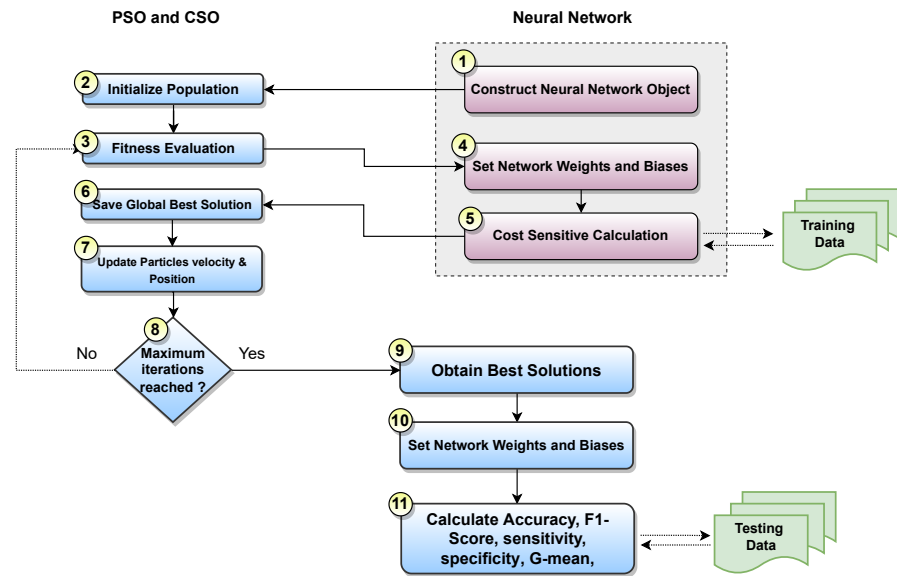


Figure 1. The cost sensitivity fitness function that was embedded in the metaheuristic optimization-based neural network architecture. Here, the metaheuristic optimizer (PSO or CSO) generated the NN weights and biases. After the optimizer found a solution, the solution was used to set the weights and biases for the NN and then the constructed NN was used to generate the predictions. After that, the costs were calculated by the cost sensitivity fitness function and the best solution was saved. These steps were repeated up to the maximum number of iterations and then the saved best solution was used to set up the NN weights and biases. Then, the trained NN was used to classify the instances in the testing dataset and all of the evaluation metrics were calculated and reported.

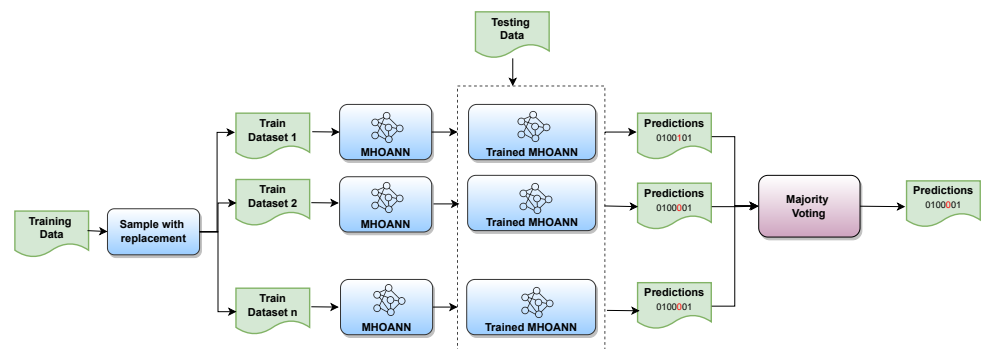


Figure 2. The MHOANN with a PSO or a CSO as the optimization technique and the cost sensitivity fitness function in the homogeneous majority voting ensemble learning paradigm architecture. Here, the training dataset was processed using sampling with replacements to generate n training datasets and then each dataset was used to train the MHOANN. Each trained MHOANN was then used to generate predictions using the same testing dataset and majority voting was used to generate the final predictions.

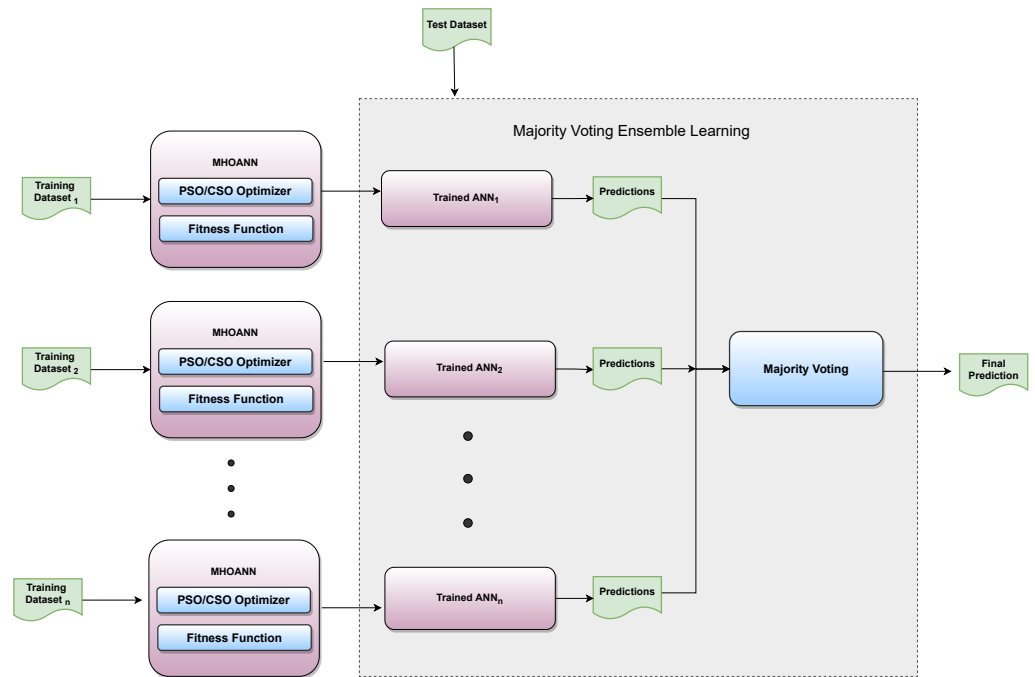


Figure 3. A component diagram of ENS_PSONN_{cost} and ENS_CSONN_{cost}. Here, the main blocks of our framework can be seen. Each inducer was an MHOANN with a PSO or a CSO as the optimizer for the NN, with an embedded custom fitness function that was cost-sensitive. In the second block, the output of each inducer was combined with the output of the other inducers to generate with the final predictions, based on the majority voting method.

5.1. ANN Classifier

Artificial neural networks (ANNs) [13,37–39] are one of the main tools that are used to solve classification problems and are brain-inspired systems that are intended to simulate the way that humans learn. The learning process of an ANN is very difficult, owing to its nonlinear nature and the unknown optimal set weights and biases of the neural network. The efficiency of an ANN is significantly affected by its learning process. An architectural diagram of a standard ANN is shown in Figure 4.

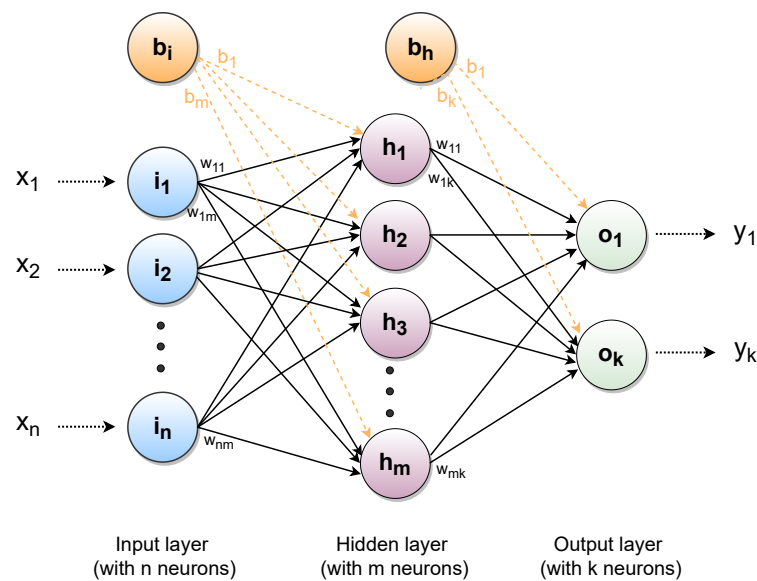


Figure 4. The standard artificial neural network architecture.

5.2. The Optimizer

Optimization algorithms are methods that are used to update the weights and biases in an ANN to overcome the disadvantages of conventional training algorithms. In this work, we utilized state-of-the-art PSO and CSO metaheuristic algorithms as the optimization techniques for our ANN.

In this work, we constructed a neural network model using two sets of weights, $(w_{11} - w_{nm})$ and $(w_{11} - w_{mk})$, and two sets of biases, $(\beta_1 - \beta_m)$ and $(\beta_1 - \beta_k)$, where n is the total number of input features, m represents the number of hidden neurons, k is the number of output neurons, w represents the weights between the input and hidden layers (and the weights between the hidden and output layers), and β represents the biases of the hidden and output layers. Every particle in the swarm population corresponded to one vector. The total length of the solution vector (l_s) could be calculated using Equation (5). An illustration of a solution vector (particle) is shown in Figure 5. In the binary classification, as we had a single neuron in the output layer, k was equal to 1 and the total length of a solution vector (l_{s_binary}) could be simplified, as in Equation (6).

$$l_s = (n * m) + (m * k) + m + k \tag{5}$$

$$l_{s_binary} = (n * m) + (2 * m) + 1 \tag{6}$$

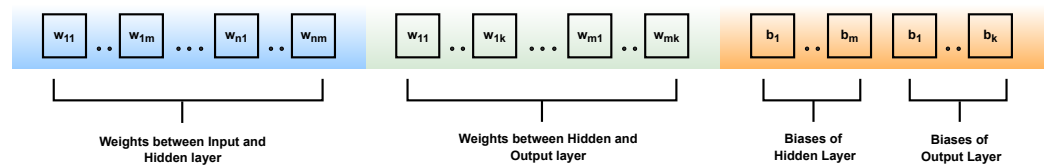


Figure 5. A representation of solution vectors (particles).

5.3. Fitness Functions

In evolutionary computing, the population evolves to increase its fitness, which is the selected fitness function [40]. In this work, we used the mean squared error (MSE) and accuracy as the benchmark fitness functions for the proposed cost sensitivity fitness function. In these cases, the fitness was compared using the following functions.

5.3.1. Mean Squared Error (MSE)

MSE is considered to be one of the most common fitness functions that are used in MHOANNs and ENNs [41,42]. The value is the mean of the summation of the differences between the predictions and the ground truths, as described in Equation (7), where $i = 1, 2, \dots, n$, n is the number of samples, y_i is the actual or ground truth value, and \hat{y}_i is the prediction.

$$cost_{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{7}$$

5.3.2. Accuracy

Accuracy is the number of correctly predicted data points out of all the data points. In this case, the value was simply the accuracy subtracted from 1 (see Equation (8), where TP is the number of true positives, TN is the number of the true negatives, FP is the number of false positives, and FN is the number of false negatives).

$$cost_{accuracy} = 1 - \left(\frac{TP + TN}{TP + TN + FP + FN} \right) \tag{8}$$

5.3.3. Cost Sensitivity

We took misclassification costs into consideration using a cost matrix. Similar to a confusion matrix, a cost matrix is an $n \times n$ matrix (where n is the number of classes) and

each element within the cost matrix represents the weight of the misclassification costs of the corresponding element in the confusion matrix.

We let A be the confusion matrix and C be the cost matrix. We multiplied each element in the confusion matrix by its corresponding weight in the cost matrix to obtain matrix A' , which was our newly updated confusion matrix. We then calculated the accuracy using the updated confusion matrix. We subtracted the resulting values from 1 to obtain the final costs. The steps that were followed to calculate the costs of the cost sensitivity fitness function are illustrated in Equation (9).

$$A = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \quad (9a)$$

$$C = \begin{bmatrix} W_{TP} & W_{FP} \\ W_{FN} & W_{TN} \end{bmatrix} \quad (9b)$$

$$A' = \begin{bmatrix} W_{TP} \times TP & W_{FP} \times FP \\ W_{FN} \times FN & W_{TN} \times TN \end{bmatrix} = \begin{bmatrix} TP' & FP' \\ FN' & TN' \end{bmatrix} \quad (9c)$$

$$CostSensitiveAccuracy = \frac{TP' + TN'}{TP' + TN' + FP' + FN'} \quad (9d)$$

$$cost_{cost_sensitive} = 1 - CostSensitiveAccuracy \quad (9e)$$

5.4. Majority Voting Ensemble Learning

Ensemble learning refers to methods for making predictions that combine several inducers. It is often used in supervised machine learning applications. An inducer, also known as a basic learner or weak learner, is a machine learning algorithm that takes a set of labeled examples as its input and produces a model. The model can then be used to make predictions for new unlabeled samples. Any type of machine learning approach can be employed as an ensemble inducer (e.g., decision trees, neural networks, linear regression models, etc.). The predictions of these models are then integrated to generate a final prediction. The core concept of ensemble learning is that by combining multiple models, the faults of an individual inducers can be compensated by the other inducers, which creates a strong learner out of several weak learners [43].

Ensemble members can be of the same or various types and they may or may not be trained using the same training dataset [44]. When all individual learners in an ensemble are of the same type, the ensemble is said to be homogeneous. For example, a “neural network ensemble” contains only neural networks [45].

In the case of classification, the combination of the results from all of the base learners can be accomplished using majority voting, which has three types: (1) unanimous voting, in which all of the classifiers agree on the prediction; (2) simple majority, in which more than half of the classifiers predict the same class; (3) plurality voting, in which the prediction receives the most votes, regardless of whether the total number of votes exceeds 50% of the classifiers [46].

In this work, we trained homogeneous ensemble learning using the MHOANN with the cost sensitivity fitness function as the ensemble members and the training dataset after applying sampling with replacements. Subsequently, majority (plurality) voting was implemented to generate the final predictions using the testing dataset.

6. Evaluation Measurements

The obvious challenge when dealing with the binary classification of an imbalanced dataset is that the training model is biased toward the majority class, resulting in a high accuracy for the majority class but the model failing to predict instances from the minority classes. In this work, we used the following metrics: accuracy, which was calculated using the confusion matrix defined in Equation (10), where TP represents the number of true positives, TN represents the number of true negatives, FP represents the number of false positives, and FN represents the number of false negatives [47]; g-mean, which was the geometric mean of the sensitivity and specificity, as defined in Equation (13); F1 score, which

was the harmonic mean of the precision and sensitivity, as defined in Equation (14), where β is the real positive factor, which was chosen such that the sensitivity was considered to be β times more important than the precision. In this work, we used $\beta = 1$, which allocated the same weighting to both the sensitivity and precision.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (10)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (11)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} \quad (12)$$

$$\text{g-mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \quad (13)$$

$$f1\text{-score} = \frac{(1 + \beta^2) \cdot \text{sensitivity} \cdot \text{precision}}{\text{sensitivity} + \beta \cdot \text{precision}} \quad (14)$$

where

$$\beta \geq 0$$

7. Experiments and Results

This section provides the experimental setups, benchmarks, and steps that were used throughout the experiments, along with the results that were obtained and their analysis.

7.1. Environmental and Experimental Setups

The experiments were executed using a laptop with 16 GB of RAM and eight cores of 2.3-GHz CPUs. We used Evolopy-NN [48] to implement the ANN, which was powered by a PSO or a CSO as the optimization technique with the cost sensitivity fitness function. Evolopy-NN is an open-source nature-inspired optimization framework for training neural networks using evolutionary and metaheuristic algorithms, which was built with Python 3.7. Both datasets were split into a training dataset (66%) and a testing dataset (34%) [49,50]. We used stratified sampling to maintain the ratio between the minor and major classes in the resulting datasets. So, after the sampling, the minor classes formed 2% of the training and testing datasets for the Spanish companies. Similarly, the minor classes formed 3% and 2% of the training and testing datasets for the Taiwanese companies and the Polish companies, respectively.

Each experiment was executed 10 different times for 100 iterations, in which the population size was set to 50. During the ensemble learning, we used five weak learners and majority voting to generate the final predictions.

As described in Section 5, we proposed the use of two optimization algorithms, a PSO and CSO, and three fitness functions, MSE, accuracy, and cost sensitivity. In this experiment, we constructed six variations of the MHOANN, as follows:

1. The ANN with a PSO and MSE as the fitness function;
2. The ANN with a PSO and accuracy as the fitness function;
3. The ANN with a PSO and cost sensitivity as the fitness function (ENS_PSONN_{cost});
4. The ANN with a CSO and MSE as the fitness function;
5. The ANN with a CSO and accuracy as the fitness function;
6. The ANN with a CSO and cost sensitivity as the fitness function (ENS_CSONN_{cost}).

7.2. Effects of Fitness Function

We extended the MHOANN to add in the costs of misclassified instances during model training by implementing a cost sensitivity fitness function, which was based on the confusion matrix that was described in Section 5. For the problem in question, we tried to

avoid FN predictions, i.e., the model predicts that a company is financially stable while it is actually in financial distress. Hence, we assigned a weighted cost to FN predictions. Determining the proper weight for the FN predictions depended on the dataset and the algorithm that were being used. We accomplished this by experimenting with different weights while monitoring the metrics to determine the best weight to use. Since the datasets in this work were relatively small in size, we were able to experiment using the whole datasets; however, in real applications with large datasets, we recommend using a sample of the dataset to find the best weight to use in order to reduce the computational costs. We considered the weight that yielded the highest g-mean score for the subsequent experiments. Table 2 shows the results for the dataset of Spanish companies with the PSO, Table 3 shows the results for the dataset of Spanish companies with the CSO, Table 4 shows the results for the dataset of Taiwanese companies with the PSO, Table 5 shows the results for the dataset of Taiwanese companies with the CSO, Table 6 shows the results for the dataset of Polish companies with the PSO, and Table 7 shows the results for the dataset of Polish companies with the CSO.

Table 2. The effects of the weight of false negative predictions on all metrics using the PSO (for the dataset of Spanish companies). The best result for each metric is marked in boldface.

FN Weight	Accuracy	Sensitivity	Specificity	F1 Score	G-Mean
1	0.978	0.048	0.999	0.088	0.219
25	0.913	0.476	0.922	0.190	0.662
50	0.818	0.810	0.818	0.160	0.814
75	0.766	0.810	0.765	0.131	0.787
100	0.749	0.952	0.745	0.141	0.842
125	0.807	0.810	0.807	0.154	0.808
150	0.713	0.810	0.711	0.108	0.759
175	0.723	0.857	0.720	0.117	0.786
200	0.724	0.857	0.721	0.117	0.786

Table 3. The effects of the weight of false negative predictions on all metrics using the CSO (for the dataset of Spanish companies). The best result for each metric is marked in boldface.

FN Weight	Accuracy	Sensitivity	Specificity	F1 Score	G-Mean
1		0.048	0.987	0.057	0.237
25	0.909	0.610	0.916	0.225	0.748
50	0.856	0.724	0.859	0.180	0.789
75	0.768	0.819	0.767	0.134	0.793
100	0.731	0.781	0.729	0.115	0.755
125	0.687	0.857	0.683	0.106	0.765
150	0.725	0.800	0.724	0.114	0.761
175	0.684	0.848	0.680	0.104	0.759
200	0.667	0.857	0.663	0.101	0.754

Table 4. The effects of the weight of false negative predictions on all metrics using the PSO (for the dataset of Taiwanese companies). The best result for each metric is marked in boldface.

FN Weight	Accuracy	Sensitivity	Specificity	F1 Score	G-Mean
1	0.967	0.064	0.997	0.110	0.244
25	0.881	0.779	0.884	0.299	0.829
50	0.824	0.848	0.823	0.242	0.834
75	0.828	0.832	0.828	0.241	0.830
100	0.766	0.880	0.762	0.198	0.819
125	0.749	0.909	0.744	0.193	0.822
150	0.759	0.872	0.755	0.192	0.810
175	0.773	0.827	0.771	0.208	0.790
200	0.774	0.827	0.771	0.208	0.790

Table 5. The effects of the weight of false negative predictions on all metrics using the CSO (for the dataset of Taiwanese companies). The best result for each metric is marked in boldface.

FN Weight	Accuracy	Sensitivity	Specificity	F1 Score	G-Mean
1	0.968	0.049	0.998	0.087	0.207
25	0.861	0.773	0.864	0.265	0.817
50	0.808	0.888	0.806	0.237	0.845
75	0.776	0.867	0.773	0.201	0.818
100	0.763	0.880	0.759	0.197	0.817
125	0.755	0.880	0.751	0.197	0.811
150	0.632	0.942	0.622	0.143	0.765
175	0.720	0.898	0.714	0.172	0.800
200	0.702	0.907	0.696	0.171	0.792

Table 6. The effects of the weight of false negative predictions on all metrics using the PSO (for the dataset of Polish companies). The best result for each metric is marked in boldface.

FN Weight	Accuracy	Sensitivity	Specificity	F1 Score	G-Mean
1	0.967	0.014	0.987	0.018	0.118
25	0.887	0.377	0.897	0.119	0.582
50	0.824	0.464	0.832	0.097	0.621
75	0.740	0.522	0.745	0.076	0.624
100	0.755	0.652	0.757	0.098	0.703
125	0.727	0.71	0.728	0.096	0.719
150	0.737	0.826	0.735	0.113	0.779
175	0.792	0.899	0.789	0.149	0.842
200	0.706	0.826	0.704	0.103	0.763
225	0.653	0.754	0.651	0.081	0.701

Table 7. The effects of the weight of false negative predictions on all metrics using the CSO (for the dataset of Polish companies). The best result for each metric is marked in boldface.

FN Weight	Accuracy	Sensitivity	Specificity	F1 Score	G-Mean
1	0.967	0.014	0.986	0.017	0.117
25	0.709	0.457	0.714	0.061	0.571
50	0.703	0.478	0.708	0.062	0.582
75	0.639	0.623	0.640	0.066	0.631
100	0.609	0.768	0.605	0.074	0.682
125	0.620	0.812	0.616	0.080	0.707
150	0.725	0.841	0.723	0.111	0.780
175	0.789	0.913	0.787	0.150	0.848
200	0.705	0.841	0.703	0.104	0.769
225	0.652	0.768	0.650	0.082	0.707

From these experiments, we observed that the best weight for FN predictions when using the PSO for the dataset of Spanish companies was 100, as shown in Figure 6. The corresponding result was 75 when using the CSO for the same dataset, as shown in Figure 7. On the other hand, we noticed that the best weight for FN predictions when using the PSO for the dataset of Taiwanese companies was 50, as shown in Figure 8. The result was the same when the CSO was used for the same dataset, as shown in Figure 9. We also noticed that the best weight for FN predictions when using the PSO for the dataset of Polish companies was 175, as shown in Figure 10. The result was the same when using the CSO for the same dataset, as shown in Figure 11. After determining the best FN weight for each particular optimization algorithm and dataset, we trained the MHOANN using the cost sensitivity fitness function, fed it with the corresponding FN weight, and then used the trained model to classify the instances in the testing dataset. We observed that in order to obtain reasonable g-mean scores, the weight of the FN predictions needed to be

considerably high, from 50 to 175. This could be explained by the extreme imbalance of the data in the considered datasets.

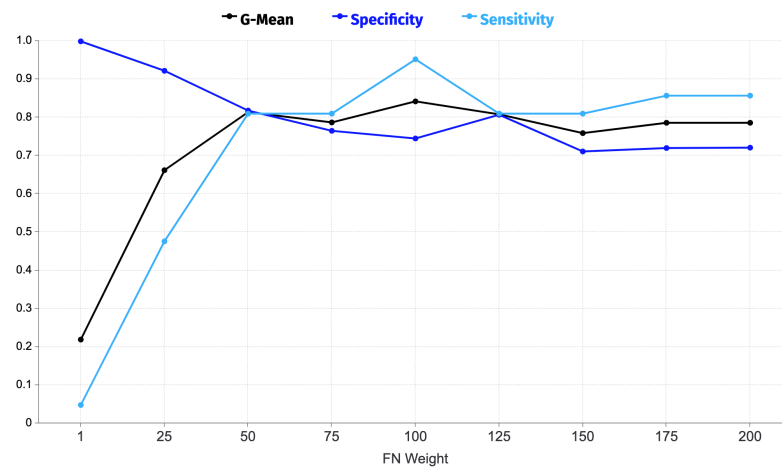


Figure 6. The effects of FN prediction weight on g-mean, specificity, and sensitivity when using the cost-sensitive MHOANN with the PSO for the dataset of Spanish companies.

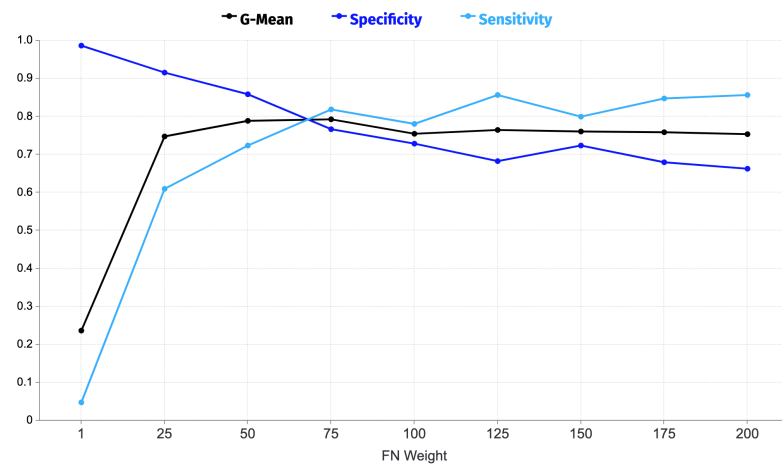


Figure 7. The effects of FN prediction weight on g-mean, specificity, and sensitivity when using the cost-sensitive MHOANN with the CSO for the dataset of Spanish companies.

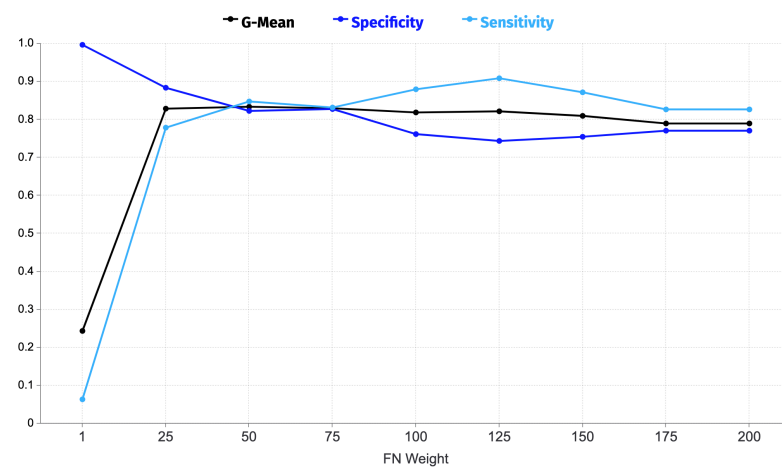


Figure 8. The effects of FN prediction weight on g-mean, specificity, and sensitivity when using the cost-sensitive MHOANN with the PSO for the dataset of Taiwanese companies.

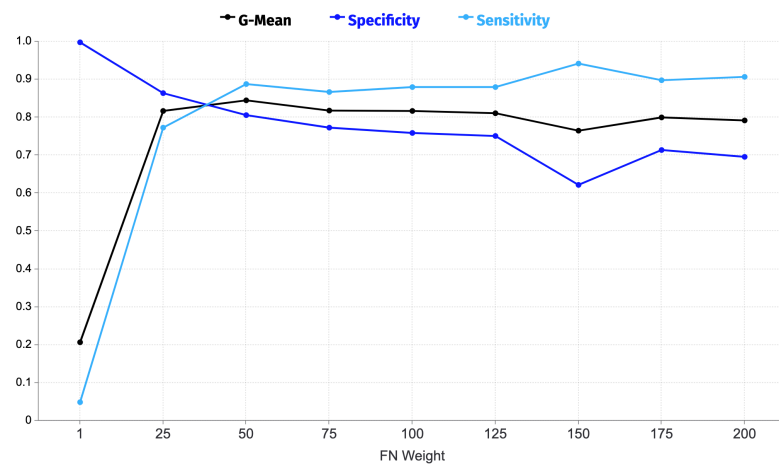


Figure 9. The effects of FN prediction weight on g-mean, specificity, and sensitivity when using the cost-sensitive MHOANN with the CSO for the dataset of Taiwanese companies.

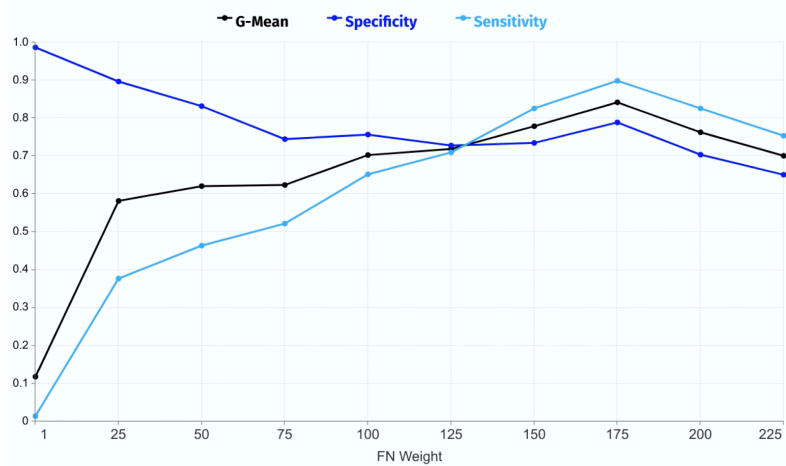


Figure 10. The effects of FN prediction weight on g-mean, specificity, and sensitivity when using the cost-sensitive MHOANN with the PSO for the dataset of Polish companies.

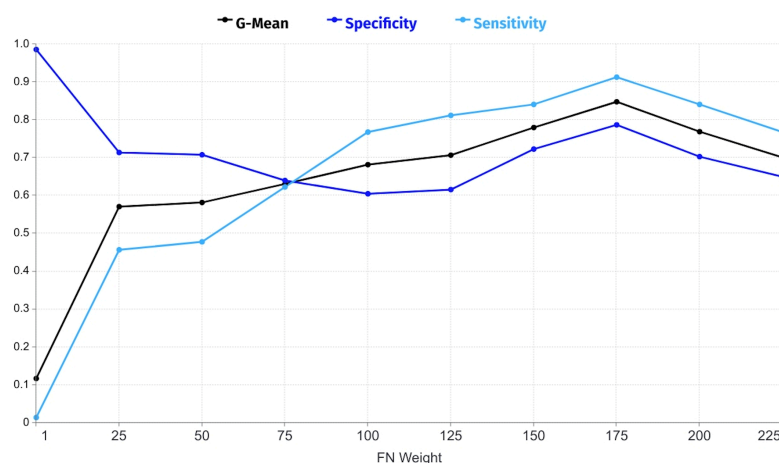


Figure 11. The effects of FN prediction weight on g-mean, specificity, and sensitivity when using the cost-sensitive MHOANN with the CSO for the dataset of Polish companies.

To assess the effects of the cost sensitivity fitness function, we based our results on a benchmark. In the benchmark, we used each optimizer (PSO and CSO) with two different fitness functions, MSE and accuracy, and then trained the ANN using both datasets to observe the evaluation metrics without cost-sensitive learning. For each dataset, we

executed four experiments: the ANN with the PSO and MSE as the fitness function, the ANN with the PSO and accuracy as the fitness function, the ANN with the CSO and MSE as the fitness function, and the ANN with the CSO and accuracy as the fitness function. The averages and standard deviations were calculated, along with the best scores for each metric.

Table 8 shows the results for all of the fitness functions that were applied to the dataset of Spanish companies. In Table 9, the results from all of the fitness functions that were applied to the dataset of Taiwanese companies are illustrated. In Table 10, the results from all of the fitness functions that were applied to the dataset of Polish companies are shown. The cost-sensitive MHOANN showed major improvements when predicting the minority classes, which had a major positive impact on the g-mean and F1 score metrics and a negative impact on the accuracy.

Table 8. The results of the evaluation metrics for all of the fitness functions that were applied to the dataset of Spanish companies per optimization algorithm. The best average result for each metric is marked in boldface.

Fitness Function	Optimizer	Accuracy			G-Mean			F1 Score		
		Avg.	Best	Std.	Avg.	Best	Std.	Avg.	Best	Std.
MSE	PSO	0.978	0.980	0.002	0.211	0.309	0.126	0.104	0.174	0.071
Accuracy	PSO	0.979	0.979	0.001	0.131	0.218	0.120	0.054	0.091	0.049
Cost Sensitivity	PSO	0.749	0.750	0.001	0.842	0.843	0.001	0.141	0.142	0.001
MSE	CSO	0.980	0.981	0.001	0.211	0.309	0.126	0.104	0.174	0.071
Accuracy	CSO	0.980	0.981	0.000	0.062	0.308	0.138	0.032	0.160	0.072
Cost Sensitivity	CSO	0.768	0.771	0.001	0.793	0.801	0.001	0.134	0.150	0.001

Table 9. The results of the evaluation metrics for all of the fitness functions that were applied to the dataset of Taiwanese companies per optimization algorithm. The best average result for each metric is marked in boldface.

Fitness Function	Optimizer	Accuracy			G-Mean			F1 Score		
		Avg.	Best	Std.	Avg.	Best	Std.	Avg.	Best	Std.
MSE	PSO	0.968	0.970	0.001	0.332	0.415	0.069	0.186	0.257	0.061
Accuracy	PSO	0.967	0.969	0.001	0.244	0.365	0.074	0.110	0.220	0.049
Cost Sensitivity	PSO	0.824	0.830	0.001	0.834	0.835	0.001	0.242	0.243	0.001
MSE	CSO	0.967	0.969	0.001	0.290	0.346	0.079	0.147	0.198	0.065
Accuracy	CSO	0.968	0.969	0.001	0.207	0.305	0.095	0.087	0.163	0.070
Cost Sensitivity	CSO	0.808	0.810	0.002	0.845	0.846	0.001	0.237	0.239	0.002

Table 10. The results of the evaluation metrics for all of the fitness functions that were applied to the dataset of Polish companies per optimization algorithm. The best average result for each metric is marked in boldface.

Fitness Function	Optimizer	Accuracy			G-Mean			F1 Score		
		Avg.	Best	Std.	Avg.	Best	Std.	Avg.	Best	Std.
MSE	PSO	0.970	0.971	0.001	0.118	0.118	0.000	0.019	0.020	0.001
Accuracy	PSO	0.967	0.969	0.001	0.118	0.118	0.000	0.018	0.019	0.001
Cost Sensitivity	PSO	0.792	0.792	0.001	0.842	0.849	0.007	0.149	0.151	0.002
MSE	CSO	0.970	0.971	0.001	0.118	0.118	0.000	0.020	0.020	0.000
Accuracy	CSO	0.967	0.969	0.001	0.117	0.118	0.001	0.017	0.019	0.001
Cost Sensitivity	CSO	0.790	0.794	0.003	0.848	0.850	0.001	0.150	0.152	0.002

Using the dataset of Spanish companies, when comparing the ANN with the PSO and the cost sensitivity fitness function to the same classifier with MSE as the fitness function, we noticed a major improvement in the g-mean from 0.211 to 0.842, an improvement in the F1 score from 0.104 to 0.141, and a drop in the accuracy from 0.978 to 0.749. When comparing the ANN with the PSO and the cost sensitivity fitness function to the same classifier with accuracy as the fitness function, we observed similar results: a major increase in the g-mean from 0.131 to 0.842, an improvement in the F1 score from 0.054 to 0.141, and a drop in the accuracy from 0.979 to 0.749. Similarly, when comparing the ANN with the CSO and the cost sensitivity fitness function to the same classifier with MSE as the fitness function, we noticed a major increase in the g-mean from 0.211 to 0.793, an improvement in the F1 score from 0.104 to 0.134, and a drop in the accuracy from 0.980 to 0.768. When comparing the ANN with the CSO and the cost sensitivity fitness function to the same classifier with accuracy as the fitness function, we also observed a major increase in the g-mean from 0.062 to 0.793, an improvement in the F1 score from 0.032 to 0.134, and a drop in the accuracy from 0.980 to 0.768.

We also observed similar results while using the dataset of Taiwanese companies. When comparing the ANN with the PSO and the cost sensitivity fitness function to the same classifier with MSE as the fitness function, we noticed a major increase in the g-mean from 0.332 to 0.834, an improvement in the F1 score from 0.186 to 0.242, and a drop in the accuracy from 0.968 to 0.824. When comparing the ANN with the PSO and the cost sensitivity fitness function to the same classifier with accuracy as the fitness function, we also noticed a major increase in the g-mean from 0.244 to 0.834, an increase in the F1 score from 0.110 to 0.242, and a drop in the accuracy from 0.967 to 0.824. When comparing the ANN with the CSO and the cost sensitivity fitness function to the same classifier with MSE as the fitness function, the increase in the g-mean was from 0.290 to 0.845, the increase in the F1 score was from 0.147 to 0.237, and the drop in the accuracy was from 0.967 to 0.808. Likewise, when comparing the ANN with the CSO and the cost sensitivity fitness function to the same classifier with accuracy as the fitness function, the increase in g-mean was from 0.207 to 0.845, the increase in the F1 score was from 0.087 to 0.237, and the drop in the accuracy was from 0.968 to 0.808.

Moreover, we observed similar results while using the dataset of Polish companies. When comparing the ANN with the PSO and the cost sensitivity fitness function to the same classifier with MSE as the fitness function, we noticed a major increase in the g-mean from 0.118 to 0.842, an improvement in the F1 score from 0.019 to 0.149, and a drop in the accuracy from 0.970 to 0.790. When comparing the ANN with the PSO and the cost sensitivity fitness function to the same classifier with accuracy as the fitness function, we also noticed a similar increase in the g-mean from 0.118 to 0.842, an increase in the F1 score from 0.018 to 0.149, and a drop in the accuracy from 0.967 to 0.790. When comparing the ANN with the CSO and the cost sensitivity fitness function to the same classifier with MSE as the fitness function, the increase in the g-mean was from 0.118 to 0.848, the increase in the F1 score was from 0.020 to 0.150, and the drop in the accuracy was from 0.970 to 0.790. Likewise, when comparing the ANN with the CSO and the cost sensitivity fitness function to the same classifier with accuracy as the fitness function, the increase in the g-mean was from 0.117 to 0.848, the increase in the F1 score was from 0.017 to 0.150, and the drop in the accuracy was from 0.967 to 0.790.

We could see that by applying the weight of the FN predictions, the number of TP instances increased, which explained the improvements in the g-mean and F1 score values. However, it also caused an increase in the number of FP instances, which explained the decrease in the accuracy score. Next, we used majority voting ensemble learning to decrease the number of FP instances while maintaining the number of TP instances.

Additionally, since the PSO and CSO produced similar results, an interesting observation was that a light optimizer with a simple mechanism to update the particles within the search space, such as the CSO, could achieve similar results when used as an optimizer for an MHOANN.

Another observation was that, whereas the PSO and CSO produced similar results when using similar fitness functions, the CSO was better in terms of execution time. Using the same population size of (50) and the same number of iterations (100), CSO was 22.4% faster for the dataset of Spanish companies, 34.4% faster for the dataset of Taiwanese companies, and 48.3% faster for the dataset of Polish companies. Table 11 lists the actual execution times in seconds.

In this work, as discussed in Section 7.2, we noticed a direct relationship between the weight of the FN predictions and the set of metrics that were monitored. While we chose the weight that produced the best g-mean score, which meant a weight that produced a balance between sensitivity and specificity, a lower weight could produce a better specificity score and a higher weight could produce a better sensitivity score, depending on which metric the user focused on.

Table 11. A comparison of the PSO and CSO execution times.

Optimizer	Dataset	Execution Time (s)		
		Avg.	Best	Std.
PSO	Spanish	196.8	188.8	5.7
CSO	Spanish	152.8	151.0	2.3
PSO	Taiwanese	1260.0	1212.9	58.7
CSO	Taiwanese	826.6	798.3	19.7
PSO	Polish	1778.1	1732.1	50.3
CSO	Polish	918.7	820.0	59.4

7.3. Effects of the Ensemble Learning Framework

Whereas the cost-sensitive MHOANN performed better when predicting the minority classes and significantly reduced the number of FN instances, there was an increase in FP predictions as well. However, for these particular datasets, the minority classes were far more valuable and essential than the majority class. In other words, predicting that a company is solvent when it is actually in financial distress has considerably higher costs than predicting that a company is in financial distress when it is actually stable [51]; therefore, maintaining a high accuracy score in the classification model was crucial.

As described in Section 5, the key premise of ensemble learning is that by mixing many models, the flaws of one model can most likely be cancelled out by the other models. Hence, we used sampling with replacements to create five training sets per dataset and then trained the cost-sensitive MHOANN using each new training set. We then generated predictions using the existing testing dataset and used majority voting to obtain the final predictions.

Table 12 shows a comparison of the results from the cost-sensitive MHOANN using the dataset of Spanish companies and those from the cost-sensitive MHOANN within the majority voting ensemble learning system using the same dataset. In Table 13, a comparison of the results from the cost-sensitive MHOANN and those from the cost-sensitive MHOANN within the majority voting ensemble learning system using the dataset of Taiwanese companies is illustrated. Table 14 shows the same comparison for the dataset of Polish companies. By reviewing the results, we observed improvements in most of the evaluation metrics; specifically, we noticed an improvement in the accuracy of between 8.4% and 15.0%, an improvement in the g-mean score of between 4.2% and 12.6%, and a significant improvement in the F1 score of between 36.7% and 87.3%.

Table 12. A comparison between the results of the evaluation metrics from the cost-sensitive MHOANN and those from the cost-sensitive MHOANN within the majority voting ensemble learning system using the dataset of Spanish companies.

Algorithm	Optimizer	Accuracy	Sensitivity	Specificity	F1 Score	G-Mean
Cost-Sensitive	PSO	0.749	0.952	0.745	0.141	0.842
Ensemble Learning	PSO	0.851	0.905	0.850	0.207	0.877
Percentage Change		13.6%	−5.0%	14.1%	46.8%	4.2%
Cost-Sensitive	CSO	0.768	0.819	0.767	0.134	0.793
Ensemble Learning	CSO	0.883	0.905	0.882	0.251	0.893
Percentage Change		15.0%	10.5%	15.0%	87.3%	12.6%

Table 13. A comparison between the results of the evaluation metrics from the cost-sensitive MHOANN and those from the cost-sensitive MHOANN within the majority voting ensemble learning system using the dataset of Taiwanese companies.

Algorithm	Optimizer	Accuracy	Sensitivity	Specificity	F1 Score	G-Mean
Cost-Sensitive	PSO	0.824	0.848	0.823	0.242	0.834
Ensemble Learning	PSO	0.910	0.840	0.912	0.376	0.875
Percentage Change		10.4%	−1.0%	10.8%	55.4%	4.9%
Cost-Sensitive	CSO	0.808	0.888	0.806	0.237	0.845
Ensemble Learning	CSO	0.876	0.920	0.874	0.324	0.897
Percentage Change		8.4%	3.6%	8.4%	36.7%	6.2%

Table 14. A comparison between the results of the evaluation metrics from the cost-sensitive MHOANN and those from the cost-sensitive MHOANN within the majority voting ensemble learning system using the dataset of Polish companies.

Algorithm	Optimizer	Accuracy	Sensitivity	Specificity	F1 Score	G-Mean
Cost-Sensitive	PSO	0.792	0.899	0.789	0.149	0.842
Ensemble Learning	PSO	0.898	0.913	0.898	0.261	0.905
Percentage Change		13.4%	1.6%	13.8%	75.2%	7.5%
Cost-Sensitive	CSO	0.789	0.913	0.787	0.150	0.848
Ensemble Learning	CSO	0.888	0.928	0.887	0.269	0.907
Percentage Change		12.5%	1.6%	12.7%	79.3%	7.0%

The main idea of ensemble learning is to achieve a high prediction capability that at least exceeds the individual prediction capabilities of the techniques that make up the ensemble. To achieve this, the weak learners within the ensemble should be both accurate and diverse [52]. The improvements that were achieved for all metrics confirmed that the PSO and CSO were sufficient to optimize the ANN, which was both accurate and diverse and could be utilized within a homogeneous ensemble learning system.

7.4. Comparison to Other Approaches

In [34], the authors proposed a hybrid method that combined the synthetic minority oversampling technique with other ensemble methods. Additionally, the authors applied five different feature selection methods to determine the most dominant attributes of insolvency prediction using the same dataset of Spanish companies. First, the authors compared four oversampling methods and then applied the C4.5 decision tree classifier to determine the best method. SMOTE was subsequently selected since it produced the

best results, as suggested by the authors. Second, the authors compared several standard basic and ensemble classification algorithms as the baseline for the study. Table 15 shows the g-mean scores when using the standard classifiers in [34] compared to those when using the two methods that are proposed in this work. It can be seen that the proposed methods produced higher g-mean scores than all of the other classifiers in the related study. Third, the authors compared several basic and ensemble classification algorithms after applying oversampling using SMOTE to compare their performances and select the best performing classifier. The AB-Rep tree was subsequently selected as the best classifier. Finally, the authors applied different attribute selectors for feature selection and then applied oversampling using SMOTE and classification using the AB-Rep tree algorithm before comparing the results. Table 16 shows the best results, based on the g-mean scores in [34] and those of the two methods that are proposed in this work. It is clear that the proposed methods significantly improved the g-mean scores. According to these results, we noticed the benefits of applying cost-sensitive learning to our MHOANN, as well as the advantages that could be gained by using ensemble learning to improve financial distress prediction. Although the same dataset was used in this work and in [34], it is worth mentioning that there were some differences between the experiment setups: (1) in this work, we used a 66% to 34% split for the training and testing datasets, while the authors of [34] used a 10-fold cross-validation technique that meant that 90% of their data were used to train the model, but the approach that is proposed in this work still showed better results; (2) ten separate runs were performed in [34] for each combination, while we performed five separate runs per combination in this work.

Table 15. The results for the g-mean scores of the standard classifiers that were used in the related work compared to those of the two methods that are proposed in this work using the dataset of Spanish companies. The best g-mean result per classification approach is marked in boldface.

	Classifier	G-Mean
Basic Classifiers	k-NN [34]	0.367
	MLP [34]	0.427
	Naive Bayes [34]	0.402
	Random Tree [34]	0.602
	J48 [34]	0.583
	Rep tree [34]	0.336
Ensembles	Bag-J48/(10) [34]	0.488
	AB-J48(20) [34]	0.609
	Dec-J48/(10) [34]	0.549
	RF-J48(80) [34]	0.509
	Bag-Rep Tree/(80) [34]	0.315
	AB-Rep Tree (90) [34]	0.602
	Dec-Rep Tree/(10) [34]	0.414
	RF-Rep Tree (10) [34]	0.094
	Bag-Random Tree/(100) [34]	0.491
	AB-Random Tree/(10) [34]	0.574
	Dec-Random Tree/(20) [34]	0.532
	RtF-Random Tree/(30) [34]	0.518
RF/(50) [34]	0.464	
Proposed Methods	ENS_PSONN _{cost}	0.877
	ENS_CSONN _{cost}	0.893

Table 16. The best results for the g-mean scores of the hybrid methods that were used in the related work compared to those of the two methods that are proposed in this work using the dataset of Spanish companies. The best g-mean result per classification approach is marked in boldface

Classifier	Oversampling	Feature Selection	G-Mean
Random Tree [34]	No	No	0.602
AB-J48(20) [34]	No	No	0.609
Random Tree [34]	Yes	No	0.696
AB-Rep Tree/(90) [34]	Yes	No	0.730
AB-Rep Tree/(90) [34]	Yes	Yes	0.720
ENS_PSONN _{cost}	No	No	0.877
ENS_CSONN _{cost}	No	No	0.893

In another study that used the dataset of Taiwanese companies [35], the authors established that the integration of financial ratios (FRs) and corporate governance indicators (CGIs) could enhance the performance of the classifiers when forecasting the financial health of Taiwanese firms. Following this combination, five feature selection methodologies were evaluated to see whether they could lower data dimensionality. Consequently, the best results were achieved using an SVM with the stepwise discriminant analysis (SDA) feature selection method, along with the combination of FRs and CGIs (FC). The g-mean was not used as an evaluation metric in that study. Instead, type I and type II errors were used.

A type I error [53] is also known as the False Positive Rate (FPR). In binary classification tasks, the FPR quantifies the proportion of false positives among all of the positive samples. It is defined in Equation (15):

$$\text{Type I error} = \frac{FP}{TN + FP} = 1 - \text{Specificity} \tag{15}$$

A type II error [53] is also known as the False Negative Rate (FNR). In binary classification tasks, the FNR quantifies the proportion of false negatives among all of the negative samples. It is defined in Equation (16):

$$\text{Type II error} = \frac{FN}{TP + FN} = 1 - \text{Sensitivity} \tag{16}$$

Hence, the g-mean score could be extracted using Equation (17):

$$\text{g-mean} = \sqrt{(1 - \text{Type II error}) \times (1 - \text{Type I error})} \tag{17}$$

Table 17 shows the best results for the calculated g-mean scores using the type I and type II errors in [35] and the two methods that are proposed in this work. It can be seen that both of the proposed methods produced higher g-mean scores.

Table 17. The best results for the g-mean scores that were obtained in the related work compared to those of the two methods that are proposed in this work using the dataset of Taiwanese companies. The best g-mean result is marked in boldface.

Classifier	G-Mean
SVM+SDA+FC [35]	0.814
ENS_PSONN _{cost}	0.875
ENS_CSONN _{cost}	0.897

7.5. Analysis and Discussion

The results from our experiments indicated that for highly imbalanced datasets, the proposed method had a significant positive impact on the g-mean score (which measures the balance between the classification performances for both the majority and minority

classes) while maintaining an acceptable accuracy score. We found that the cost sensitivity fitness function helped to shift the bias away from the majority class and toward the minority classes and that ensemble learning could help to decrease the side effects of that bias shift.

In line with our hypothesis, applying a weight to the misclassified positive instances increased the number of TP predictions and decreased the number of FN predictions. However, as a side effect, the number of FP predictions increased and the number of TN predictions decreased. Since we were dealing with highly imbalanced datasets, the number of instances that belonged to the minor class (TP + FN) was much lower than the number of instances that belonged to the major class (FP + TN); so, the improvement in the sensitivity score was significant and the drop in the specificity score was not as drastic, which led to an overall improved g-mean score, as observed in the results from all experiments.

Moreover, when applying ensemble learning, we observed an overall improvement in all of the evaluation measurements that were used. This proved that the MHOANN was diverse and could be used in a homogeneous ensemble learning system. The ensemble learning created a stronger learner that approximately maintained the number FN predictions but decreased the number of FP predictions, resulting in a slightly better g-mean score and a significant improvement in the accuracy score.

In terms of performance, as previously mentioned, the CSO outperformed the PSO regarding execution time. In contrast to the PSO, only half of the population was updated in the CSO, which explained the faster execution times.

In Appendix A, we show the convergence (learning) curve graphs for sample runs using both optimizers (the PSO and CSO) for each fitness function and each dataset. We noticed that the fitness values were minimal in the cases of the MSE and accuracy fitness functions, which indicated that the model had a high accuracy (as confirmed by the previous results) but was biased toward the majority class and failed to predict the minority classes (as previously discussed). On the other hand, the fitness value was higher when using the cost sensitivity fitness function, which was expected because the number of FN predictions was multiplied by the allocated weight. Additionally, in all of our experiments, the fitness scores stabilized when approaching 100 iterations, which indicated that additional training would not significantly improve the model.

8. Conclusions and Future Work

This paper proposed the use of an MHOANN with a PSO or CSO as the optimization technique and a cost sensitivity fitness function within a majority voting ensemble learning system to handle the imbalanced distribution of financial distress datasets and maximize the prediction of positive instances. Experiments were conducted using datasets of Spanish companies, Taiwanese companies, and Polish companies. Then, we compared the results from the proposed approach to those that were obtained by applying the same MHOANN with a PSO or CSO but using MSE or accuracy as the fitness function.

The proposed method was able to provide better estimations for the financial distress prediction by avoiding biased results. The results showed that the cost sensitivity fitness function had an extremely positive overall effect on the accurate prediction of the minor classes in imbalanced datasets, with a significant improvement in the g-mean score and a moderately positive impact on the F1 score. Moreover, adapting the majority voting ensemble learning system improved the accuracy and the g-mean scores, along with a significant increase in the F1 scores. One primary limitation of this work was not having access to a domain expert to define the weights for the FN predictions, which is common in cost-sensitive learning [54]. It would be beneficial to obtain domain expert opinions and compared them to the proposed method to find the best weight for the FN instances.

In the future, we aim to investigate the application of the proposed method to other bankruptcy datasets. Additionally, we aim to use the same proposed approach for other imbalanced classification problems. Moreover, we aim to explore other methods for hyperparameter tuning, including finding the costs of misclassified instances, such as AutoML [55].

Author Contributions: Conceptualization, S.A.-D.S. and H.F.; methodology, S.A.-D.S.; software, S.A.-D.S.; validation, S.A.-D.S., H.F. and P.A.C.; formal analysis, H.F. and P.A.C.; investigation, S.A.-D.S.; resources, S.A.-D.S.; data curation, S.A.-D.S.; writing—original draft preparation, S.A.-D.S.; writing—review and editing, H.F. and P.A.C.; visualization, S.A.-D.S.; supervision, H.F. and P.A.C.; project administration, P.A.C. and H.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministerio Español de Ciencia e Innovación, under project number PID2020-115570GB-C22 (DemocratAI::UGR).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset of Spanish companies was bought from <http://infotel.es> (accessed on 1 May 2017), the dataset of Taiwanese companies was downloaded from <https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction> (accessed on 1 March 2020), and the dataset of Polish companies was downloaded from <https://www.kaggle.com/competitions/companies-bankruptcy-forecast/data> (accessed on 28 June 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Here, we present the figures that show the convergence (learning) curves for the sample runs using both optimizers (the PSO and CSO) for each fitness function and each dataset.

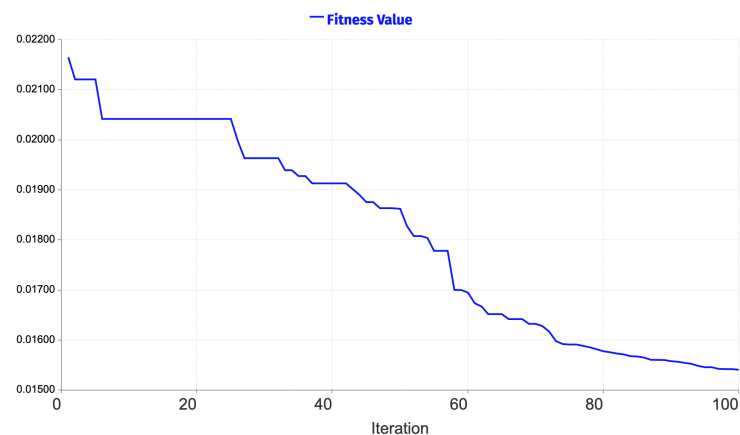


Figure A1. The training convergence curve when using the ANN with the PSO and the MSE fitness function for the dataset of Spanish companies.

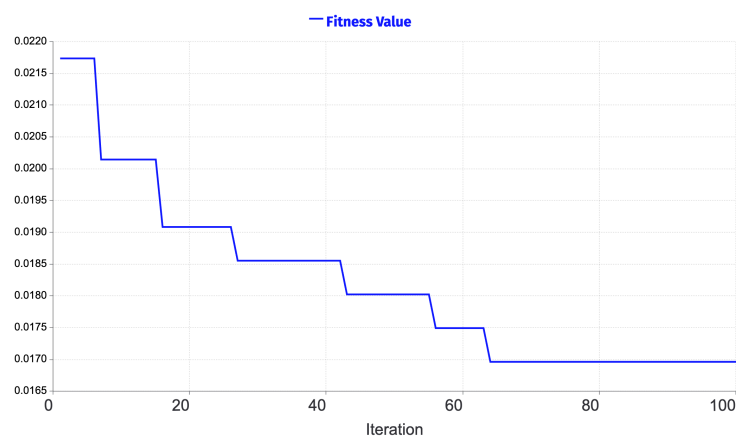


Figure A2. The training convergence curve when using the ANN with the PSO and the accuracy fitness function for the dataset of Spanish companies.

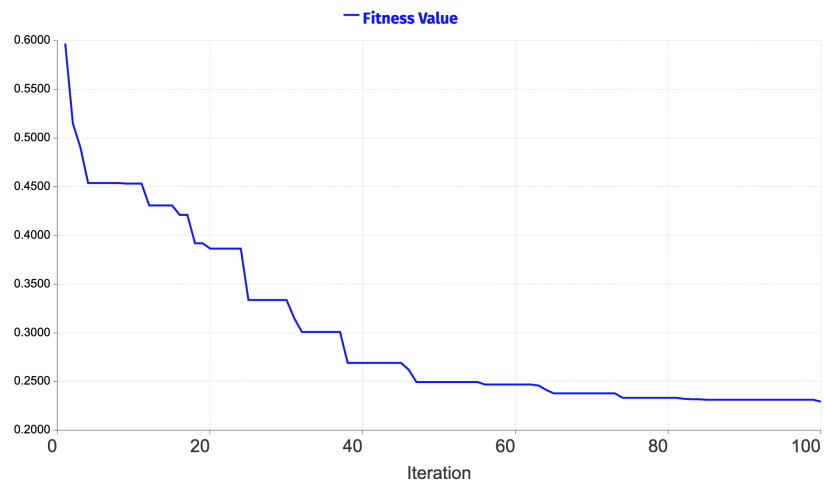


Figure A3. The training convergence curve when using the ANN with the PSO and the cost sensitivity fitness function for the dataset of Spanish companies.

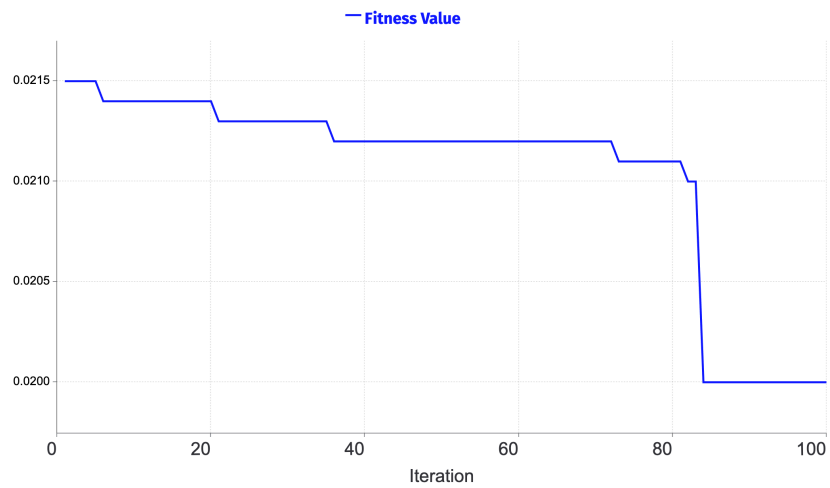


Figure A4. The training convergence curve when using the ANN with the CSO and the MSE fitness function for the dataset of Spanish companies.

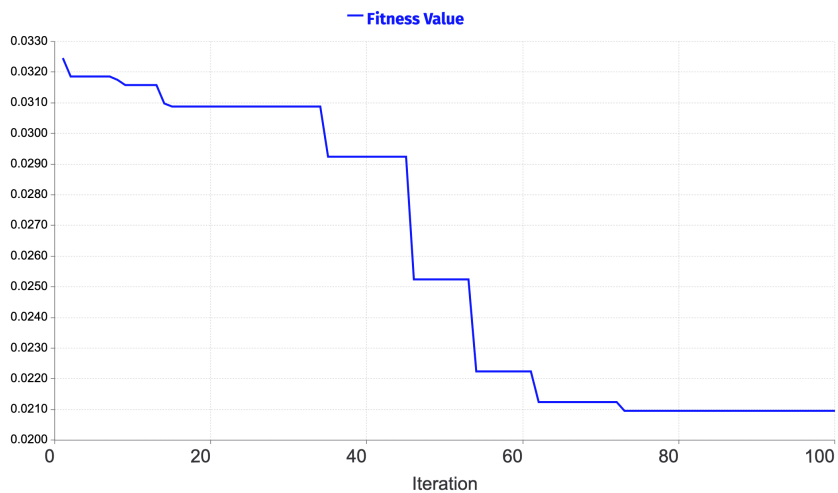


Figure A5. The training convergence curve when using the ANN with the CSO and the accuracy fitness function for the dataset of Spanish companies.

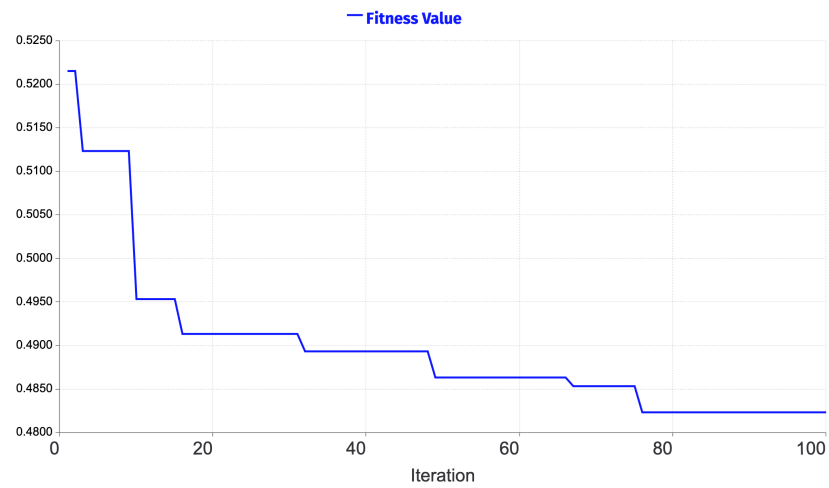


Figure A6. The training convergence curve when using the ANN with the CSO and the coast sensitivity fitness function for the dataset of Spanish companies.

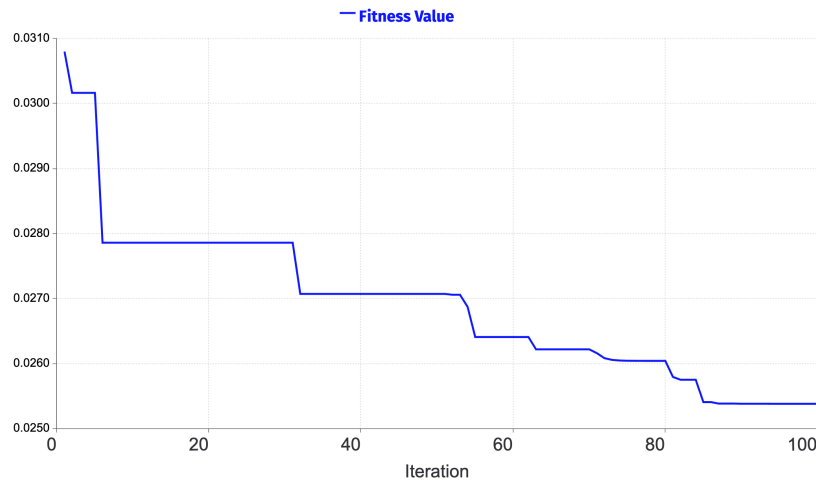


Figure A7. The training convergence curve when using the ANN with the PSO and the MSE fitness function for the dataset of Taiwanese companies.

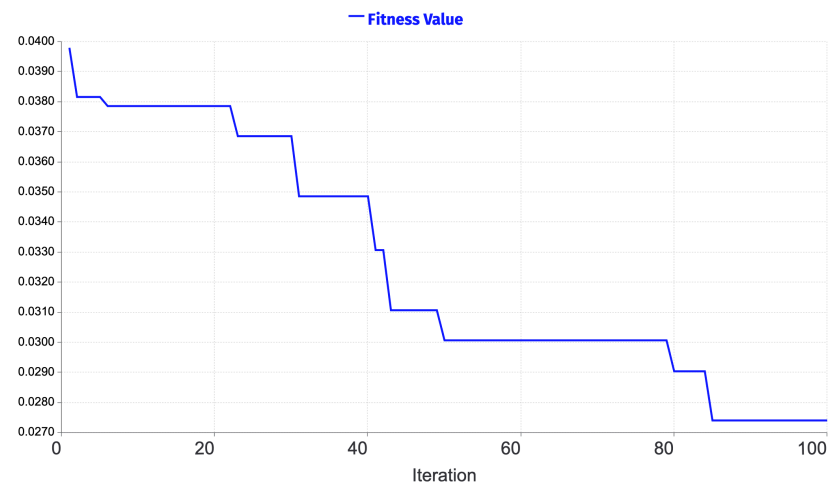


Figure A8. The training convergence curve when using the ANN with the PSO and the accuracy fitness function for the dataset of Taiwanese companies.

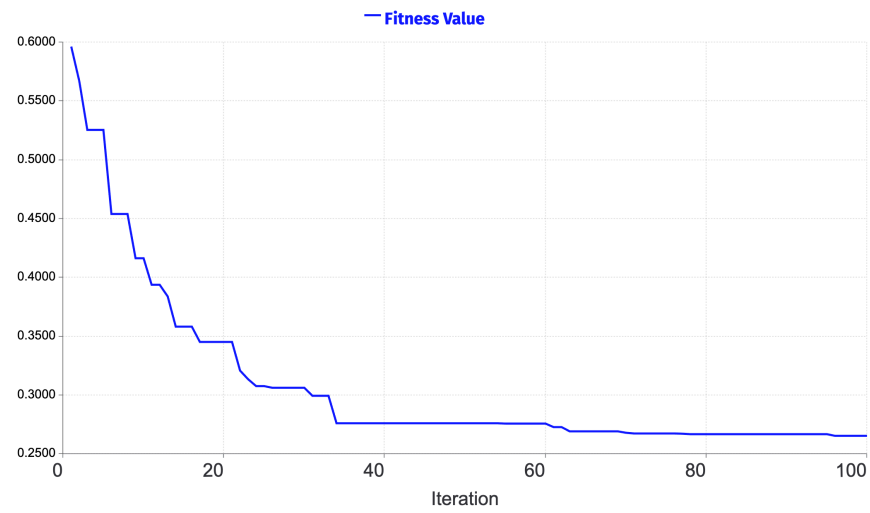


Figure A9. The training convergence curve when using the ANN with the PSO and the cost sensitivity fitness function for the dataset of Taiwanese companies.

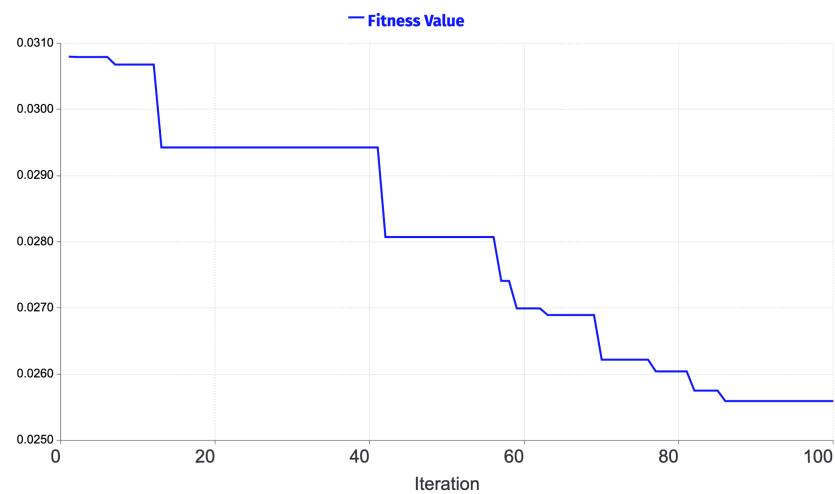


Figure A10. The training convergence curve when using the ANN with the CSO and the MSE fitness function for the dataset of Taiwanese companies.

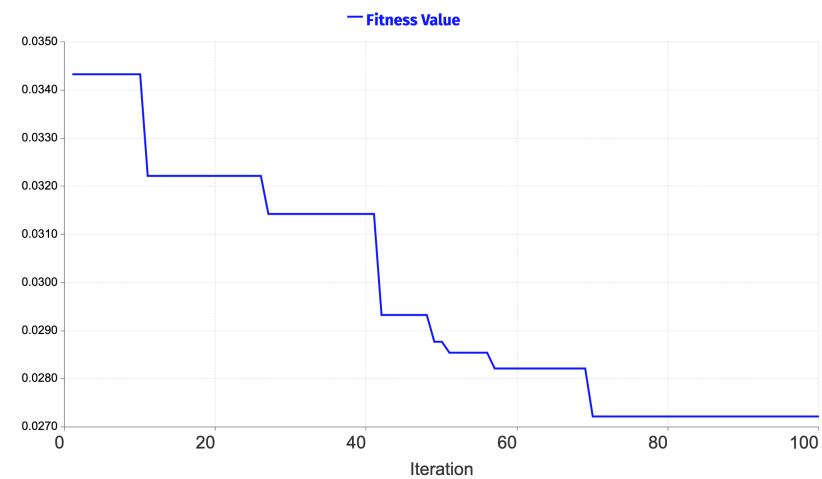


Figure A11. The training convergence curve when using the ANN with the CSO and the accuracy fitness function for the dataset of Taiwanese companies.

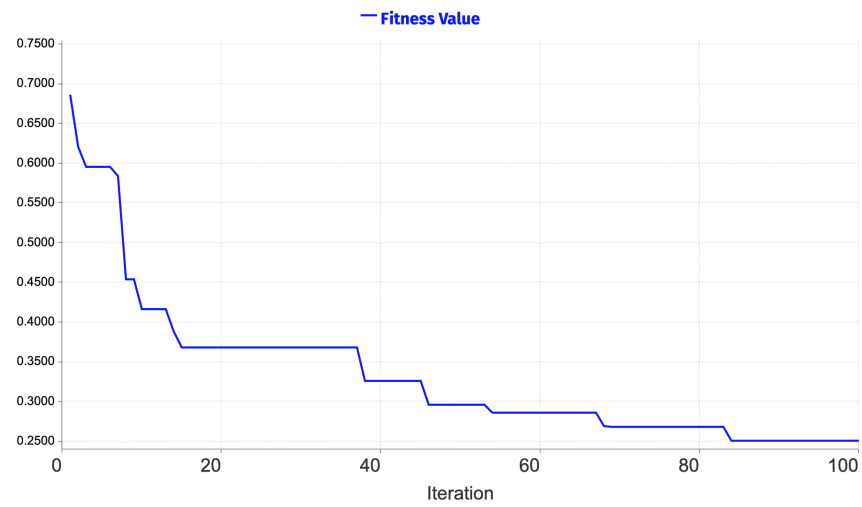


Figure A12. The training convergence curve when using the ANN with the CSO and the cost sensitivity fitness function for the dataset of Taiwanese companies.

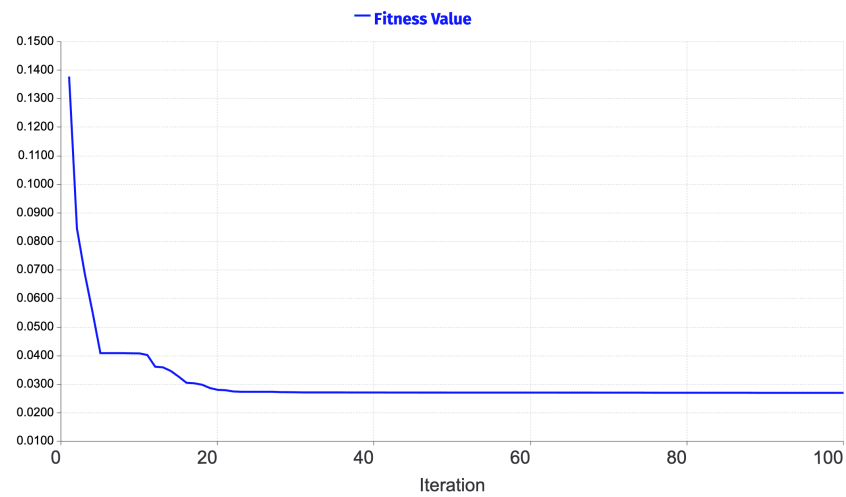


Figure A13. The training convergence curve when using the ANN with the PSO and the MSE fitness function for the dataset of Polish companies.

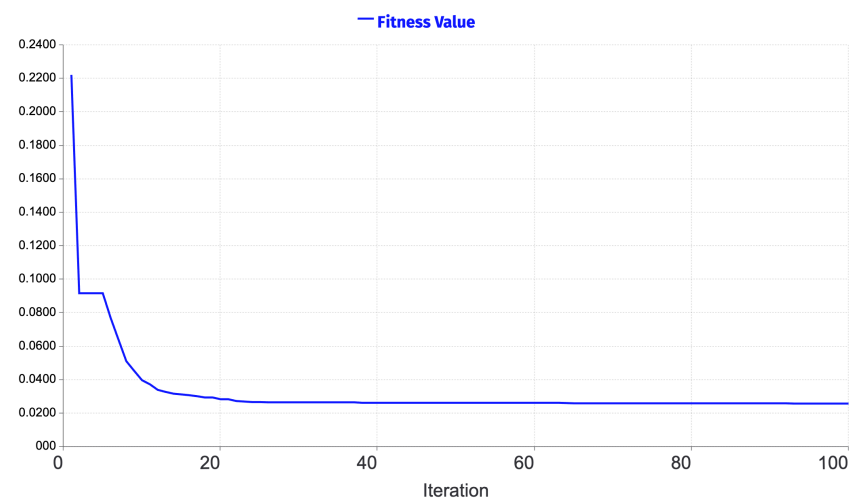


Figure A14. The training convergence curve when using the ANN with the PSO and the accuracy fitness function for the dataset of Polish companies.

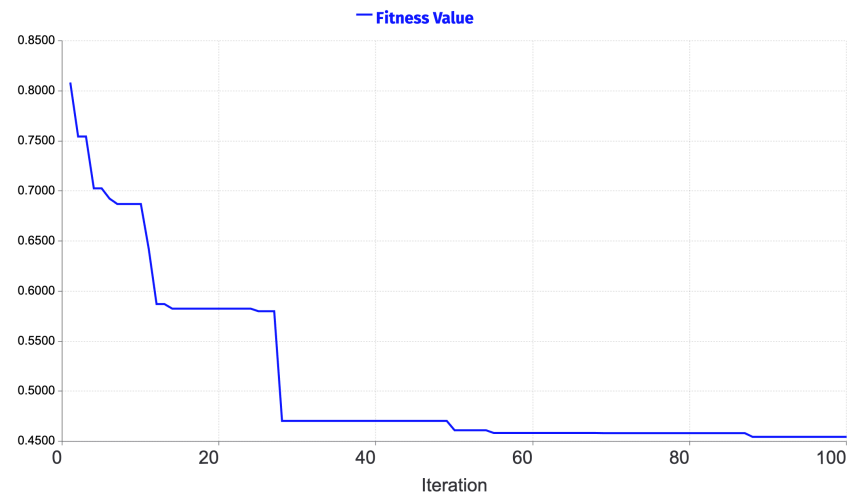


Figure A15. The training convergence curve when using the ANN with the PSO and the cost sensitivity fitness function for the dataset of Polish companies.

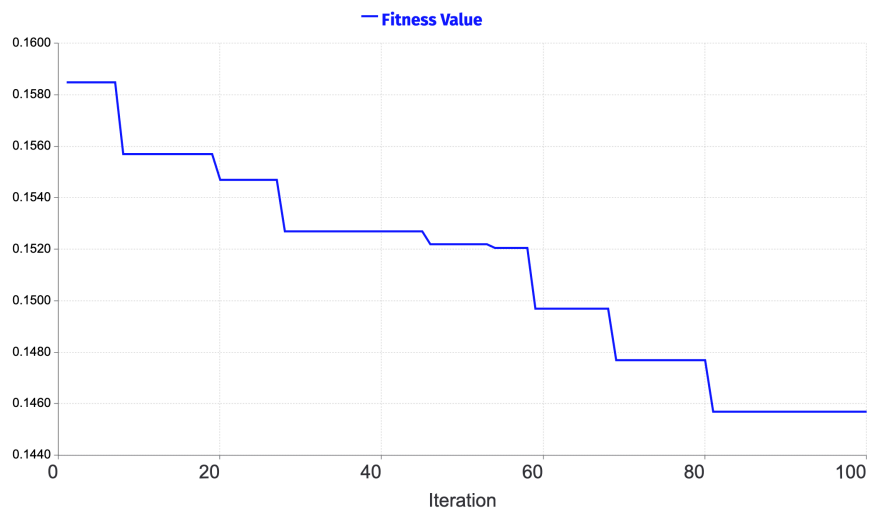


Figure A16. The training convergence curve when using the ANN with the CSO and the MSE fitness function for the dataset of Polish companies.

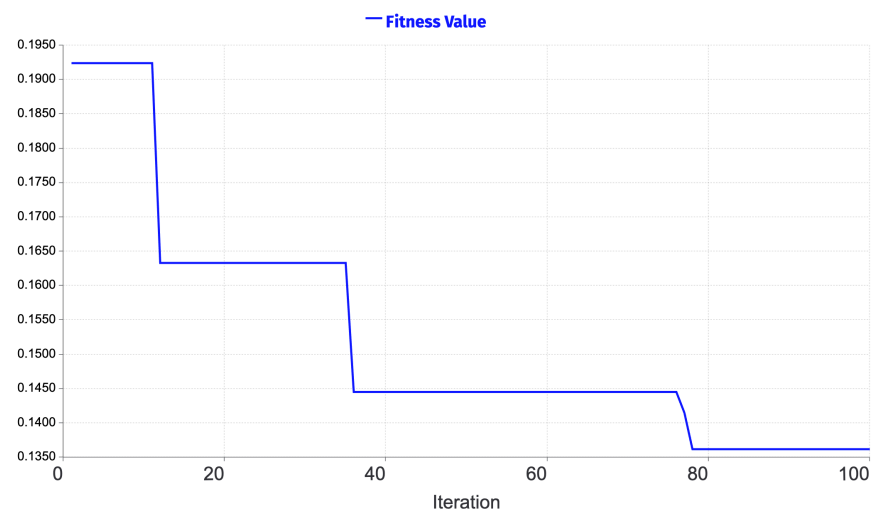


Figure A17. The training convergence curve when using the ANN with the CSO and the accuracy fitness function for the dataset of Polish companies.

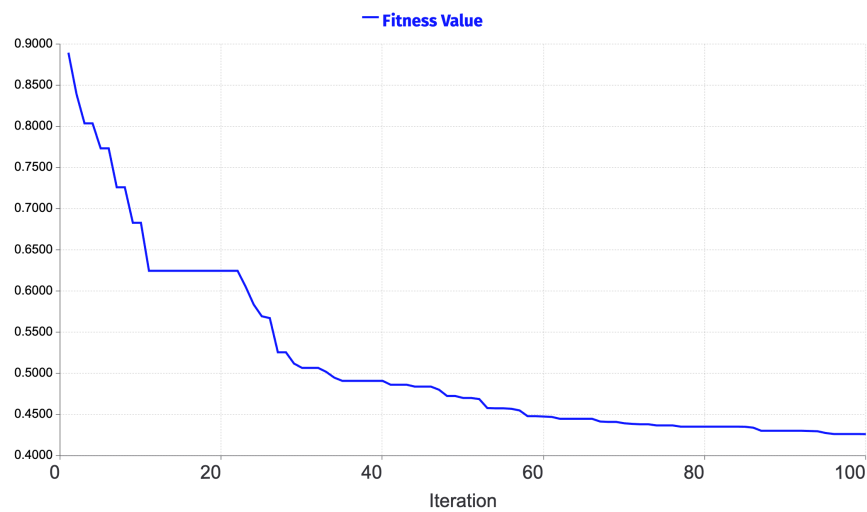


Figure A18. The training convergence curve when using the ANN with the CSO and the cost sensitivity fitness function for the dataset of Polish companies.

References

1. Bešlić Obradović, D.; Jakšić, D.; Bešlić Rupić, I.; Andrić, M. Insolvency prediction model of the company: The case of the Republic of Serbia. *Econ. Res.-Ekon. Istraž.* **2018**, *31*, 139–157. [\[CrossRef\]](#)
2. Altman, E.I.; Hotchkiss, E. *Corporate Financial Distress and Bankruptcy: Predict and Avoid Bankruptcy, Analyze and Invest in Distressed Debt*; John Wiley & Sons: Hoboken, NJ, USA, 2010; Volume 289.
3. Zhang, Y.; Liu, R.; Heidari, A.A.; Wang, X.; Chen, Y.; Wang, M.; Chen, H. Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis. *Neurocomputing* **2021**, *430*, 185–212. [\[CrossRef\]](#)
4. Ganganwar, V. An overview of classification algorithms for imbalanced datasets. *Int. J. Emerg. Technol. Adv. Eng.* **2012**, *2*, 42–47.
5. Khurma, R.A.; Aljarah, I.; Sharieh, A.; Mirjalili, S. Evolopy-fs: An open-source nature-inspired optimization framework in python for feature selection. In *Evolutionary Machine Learning Techniques*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 131–173.
6. Cheng, R.; Jin, Y. A competitive swarm optimizer for large scale optimization. *IEEE Trans. Cybern.* **2014**, *45*, 191–204. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Leung, F.H.F.; Lam, H.K.; Ling, S.H.; Tam, P.K.S. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Trans. Neural Netw.* **2003**, *14*, 79–88. [\[CrossRef\]](#)
8. Castillo, P.; Carpio, J.; Merelo, J.; Prieto, A.; Rivas, V.; Romero, G. Evolving multilayer perceptrons. *Neural Process. Lett.* **2000**, *12*, 115–128. [\[CrossRef\]](#)
9. Castillo, P.A.; Merelo, J.; Prieto, A.; Rivas, V.; Romero, G. G-Prop: Global optimization of multilayer perceptrons using GAs. *Neurocomputing* **2000**, *35*, 149–163. [\[CrossRef\]](#)
10. Castillo-Valdivieso, P.A.; Merelo, J.J.; Prieto, A.; Rojas, I.; Romero, G. Statistical analysis of the parameters of a neuro-genetic algorithm. *IEEE Trans. Neural Netw.* **2002**, *13*, 1374–1394. [\[CrossRef\]](#)
11. García-Pedrajas, N.; Hervás-Martínez, C.; Muñoz-Pérez, J. COVNET: A cooperative coevolutionary model for evolving artificial neural networks. *IEEE Trans. Neural Netw.* **2003**, *14*, 575–596. [\[CrossRef\]](#)
12. García-Pedrajas, N.; Ortiz-Boyer, D. A cooperative constructive method for neural networks for pattern recognition. *Pattern Recognit.* **2007**, *40*, 80–98. [\[CrossRef\]](#)
13. Yao, X. Evolving artificial neural networks. *Proc. IEEE* **1999**, *87*, 1423–1447.
14. Devikanniga, D.; Vetrivel, K.; Badrinath, N. Review of meta-heuristic optimization based artificial neural networks and its applications. *Proc. J. Phys. Conf. Ser.* **2019**, *1362*, 012074. [\[CrossRef\]](#)
15. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 153–165. [\[CrossRef\]](#)
16. Wang, Y.; Zhang, H.; Zhang, G. cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. *Swarm Evol. Comput.* **2019**, *49*, 114–123. [\[CrossRef\]](#)
17. Kaveh, A.; Mahdavi, V. A hybrid CBO-PSO algorithm for optimal design of truss structures with dynamic constraints. *Appl. Soft Comput.* **2015**, *34*, 260–273. [\[CrossRef\]](#)
18. Hazarika, B.B.; Gupta, D. Density-weighted support vector machines for binary class imbalance learning. *Neural Comput. Appl.* **2021**, *33*, 4243–4261. [\[CrossRef\]](#)
19. Aljawazneh, H.; Mora, A.; García-Sánchez, P.; Castillo-Valdivieso, P. Comparing the performance of deep learning methods to predict companies' financial failure. *IEEE Access* **2021**, *9*, 97010–97038. [\[CrossRef\]](#)
20. Gnip, P.; Vokorokos, L.; Drotár, P. Selective oversampling approach for strongly imbalanced data. *PeerJ Comput. Sci.* **2021**, *7*, e604. [\[CrossRef\]](#)

21. Mienye, I.D.; Sun, Y. Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. *Inform. Med. Unlocked* **2021**, *25*, 100690. [[CrossRef](#)]
22. Zou, Y.; Gao, C.; Gao, H. Business Failure Prediction Based on a Cost-Sensitive Extreme Gradient Boosting Machine. *IEEE Access* **2022**, *10*, 42623–42639. [[CrossRef](#)]
23. Montana, D.J.; Davis, L. Training feedforward neural networks using genetic algorithms. In Proceedings of the IJCAI, Detroit, MI, USA, 20–25 August 1989; Volume 89, pp. 762–767.
24. Ansari, A.; Ahmad, I.S.; Bakar, A.A.; Yaakub, M.R. A hybrid metaheuristic method in training artificial neural network for bankruptcy prediction. *IEEE Access* **2020**, *8*, 176640–176650. [[CrossRef](#)]
25. Al-Badarnah, I.; Habib, M.; Aljarah, I.; Faris, H. Neuro-evolutionary models for imbalanced classification problems. *J. King Saud Univ. Comput. Inf. Sci.* **2020**, *34* (Pt A), 2787–2797. [[CrossRef](#)]
26. Mahendru, K.; Garg, G.; Sharma, A.; Srivastava, R. Evolutionary Methods for Bankruptcy Prediction: A Study on Indian Firms. In *Soft Computing for Problem Solving*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 303–313.
27. Alibabae, G.; Khanmohammadi, M. The Study of the Predictive Power of Meta-heuristic Algorithms to Provide a Model for Bankruptcy prediction. *Int. J. Financ. Manag. Account.* **2022**, *7*, 33–51.
28. Mousavirad, S.J.; Schaefer, G.; Jalali, S.M.J.; Korovin, I. A benchmark of recent population-based metaheuristic algorithms for multi-layer neural network training. In Proceedings of the 2020 Genetic And Evolutionary Computation Conference Companion, Cancun, Mexico, 8–12 July 2020; pp. 1402–1408.
29. Yotsawat, W.; Wattuya, P.; Srivihok, A. A Novel Method for Credit Scoring Based on Cost-Sensitive Neural Network Ensemble. *IEEE Access* **2021**, *9*, 78521–78537. [[CrossRef](#)]
30. Pisula, T. An ensemble classifier-based scoring model for predicting bankruptcy of polish companies in the Podkarpackie Voivodeship. *J. Risk Financ. Manag.* **2020**, *13*, 37. [[CrossRef](#)]
31. Grzyb, J.; Klikowski, J.; Woźniak, M. Hellinger distance weighted ensemble for imbalanced data stream classification. *J. Comput. Sci.* **2021**, *51*, 101314. [[CrossRef](#)]
32. Eberhart, R.; Kennedy, J. A New Optimizer Using Particle Swarm Theory. In Proceedings of the MHS'95. Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
33. Román, I.; Gómez, M.; la Torre, J.; Merelo, J.; Mora, A. Predicting financial distress: Relationship between continued losses and legal bankruptcy. In Proceedings of the 27th Annual Congress European Accounting Association, Dublin, Ireland, 22–24 March 2006.
34. Faris, H.; Abukhurma, R.; Almanaseer, W.; Saadeh, M.; Mora, A.M.; Castillo, P.A.; Aljarah, I. Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: A case from the Spanish market. *Prog. Artif. Intell.* **2020**, *9*, 31–53. [[CrossRef](#)]
35. Liang, D.; Lu, C.C.; Tsai, C.F.; Shih, G.A. Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. *Eur. J. Oper. Res.* **2016**, *252*, 561–572. [[CrossRef](#)]
36. Zięba, M.; Tomczak, S.K.; Tomczak, J.M. Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Syst. Appl.* **2016**, *58*, 93–101. [[CrossRef](#)]
37. Armano, G.; Marchesi, M.; Murru, A. A hybrid genetic-neural architecture for stock indexes forecasting. *Inf. Sci.* **2005**, *170*, 3–33. [[CrossRef](#)]
38. Chen, Y.; Yang, B.; Dong, J.; Abraham, A. Time-series forecasting using flexible neural tree model. *Inf. Sci.* **2005**, *174*, 219–235. [[CrossRef](#)]
39. Yao, X.; Xu, Y. Recent advances in evolutionary computation. *J. Comput. Sci. Technol.* **2006**, *21*, 1–18. [[CrossRef](#)]
40. Bull, L. On model-based evolutionary computation. *Soft Comput.* **1999**, *3*, 76–82. [[CrossRef](#)]
41. Garro, B.A.; Vázquez, R.A. Designing artificial neural networks using particle swarm optimization algorithms. *Comput. Intell. Neurosci.* **2015**, *2015*, 369298. [[CrossRef](#)]
42. Gómez, J.C.; Hernández, F.; Coello, C.A.C.; Ronquillo, G.; Trejo, A. Flame classification through the use of an artificial neural network trained with a genetic algorithm. In Proceedings of the Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, 24–30 November 2013; pp. 172–184.
43. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1249. [[CrossRef](#)]
44. Zhang, C.; Ma, Y. *Ensemble Machine Learning: Methods and Applications*; Springer: Berlin/Heidelberg, Germany, 2012.
45. Zhou, Z.H. Ensemble learning. In *Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 181–210.
46. Polikar, R. Ensemble Learning. In *Ensemble Machine Learning: Methods and Applications*; Zhang, C., Ma, Y., Eds.; Springer: New York, NY, USA; Dordrecht, The Netherlands; Berlin/Heidelberg, Germany; London, UK, 2012.
47. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts Tech.*; Elsevier: Amsterdam, The Netherlands, 2011.
48. Faris, H.; Aljarah, I.; Al-Madi, N.; Mirjalili, S. Optimizing the learning process of feedforward neural networks using lightning search algorithm. *Int. J. Artif. Intell. Tools* **2016**, *25*, 1650033. [[CrossRef](#)]
49. Chen, Y.S. Building a Hybrid Prediction Model to Evaluation of Financial Distress Corporate. *Appl. Mech. Mater.* **2014**, *651–653*, 1543–1546. [[CrossRef](#)]
50. Yu, Q.; Miche, Y.; Séverin, E.; Lendasse, A. Bankruptcy prediction using extreme learning machine and financial expertise. *Neurocomputing* **2014**, *128*, 296–302. [[CrossRef](#)]
51. García, V.; Marqués, A.I.; Sánchez, J.S. An insight into the experimental design for credit risk and corporate bankruptcy prediction systems. *J. Intell. Inf. Syst.* **2015**, *44*, 159–189. [[CrossRef](#)]

52. Hosni, M.; Abnane, I.; Idri, A.; de Gea, J.M.C.; Alemán, J.L.F. Reviewing ensemble classification methods in breast cancer. *Comput. Methods Programs Biomed.* **2019**, *177*, 89–112. [[CrossRef](#)]
53. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.
54. Fernández, A.; García, S.; Galar, M.; Prati, R.C.; Krawczyk, B.; Herrera, F. *Learning From Imbalanced Data Sets*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 10.
55. Li, Y.; Wang, Z.; Xie, Y.; Ding, B.; Zeng, K.; Zhang, C. Automl: From methodology to application. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Online, 1–5 November 2021; pp. 4853–4856.