UNIVERSIDAD
DE GRANADA

SOLVING REAL-WORLD FINANCE PROBLEMS BY MEANS OF DATA MINING ALGORITHMS USING HIGH-PERFORMANCE COMPUTING PLATFORMS

Thesis submitted by

SALAH AL-DEEN TAHA SAFI

To obtain the International Ph.D. degree as part of the

PROGRAMA DE DOCTORADO EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Supervisors

PEDRO ÁNGEL CASTILLO VALDIVIESO
HOSSAM FARIS

Feb 15, 2023

Editor: Universidad de Granada. Tesis Doctorales
Autor: Salah Aldeen Safi
ISBN: 978-84-1117-876-1
URI: https://hdl.handle.net/10481/82206

DECLARATION OF SUPERVISION

The Professors of *Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada*, **Dr. Pedro Ángel Castillo Valdivieso** and **Dr. Hossam Faris**,

GUARANTEE:

that the Doctoral Thesis contained in the present report, entitled:

"***Solving real-world finance problems by means of data mining algorithms using high-performance computing platforms***"

has been done by the Ph.D. candidate **Mr. Salah Al-Deen Taha Safi** under our direction and supervision and, as far as our knowledge reaches, the rights of the cited authors have been respected when using their results or publications.

Granada, Feb 15, 2023.

The supervisors:

Sgd. Pedro Ángel Castillo Valdivieso

Sgd. Hossam Faris

DECLARATION OF AUTHORSHIP

I declare that I have done this Doctoral Thesis under the direction of my supervisors. Therefore, I assume the authorship of the contents except those mentions of other authors' works, which are duly referenced.

I also declare that the work is unpublished and is not plagiarism, in whole or in part, of any other research carried out by other persons.

Likewise, I assert that the results and data exposed in this document have not been falsified and that any error that may exist has not been introduced intentionally.

Granada, Feb 15, 2023

Sgd. Salah Al-Deen Taha Safi

To my beloved wife Duaa, whose unwavering love and support made this achievement possible.

ACKNOWLEDGEMENTS

I express my deepest gratitude to Dr. Pedro Ángel Castillo Valdivieso and Dr. Hussam Faris, for their invaluable guidance, support, and encouragement throughout this journey. Their expertise in my field of research and their constructive feedback on my work has been essential in completing this thesis. I would like to extend my sincerest thanks and gratitude to my wife, Duaa, for her unwavering love and support throughout the entire process of completing this thesis. Her encouragement, understanding, and patience have inspired and motivated me. My children Taha and Ghaith have also been a source of joy and inspiration, and I am grateful to them for bringing light and love into my life.

ACRONYMS

A

ABC	Artificial Bee Colony
AdaBoost	Adaptive Boosting
ADASYN-I	Adaptive Synthetic Sampling
ADOMS	Adjusting the Direction Of the synthetic Minority class examples
AHC	Agglomerative Hierarchical Clustering
ANN	Artificial Neural Network
ANNs	Artificial Neural Networks
AUC	Area Under the Curve

B

BPNN	Back-Propagation Neural Network

C

CART	Classification And Regression Trees
CGIs	Corporate Governance Indicators
CSO	Competitive Swarm Optimization

E

ENNs	Evolutionary Neural Networks

F

FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FRs	Financial Ratios

G

GA	Genetic Algorithm
GMBoost	Geometric Mean-Based Boosting algorithm

H

HVDM	Heterogeneous Value Distance Metric

I

IR	Imbalance Ratio

K

KNN	K-Nearest Neighbors

L

LDA	Linear Discriminant Analysis

M

MDA	Multiple Discriminate Analysis
MHOANN	Metaheuristic Optimized Artificial Neural Network
MHOANNs	Metaheuristic Optimized Artificial Neural Networks
MLP	Multi Layer Perceptron
MSE	Mean Square Error

N

NNR	Neural Network Regression

P

PSO	Particle Swarm Optimization

R

RBF	Radial Basis Function
ROS-I	Random Over-Sampling

S

SDA	Stepwise Discriminant Analysis
SMOTE	Synthetic Minority Oversampling Technique
SMOTE-ENN	Synthetic Minority Oversampling Technique and Edited Nearest Neighbor
SMOTE-TL	Synthetic Minority Oversampling Technique and Tomek's Links
SOA	Selective Oversampling Approach
SOFM	Self-Organizing Feature Maps
SPIDER	Selective Preprocessing of Imbalanced Data
SPIDER-II	Selective Preprocessing of Imbalanced Data II
SVM	Support Vector Machine

T

TN True Negative
TP True Positive

X

XGBoost Extreme Gradient Boosting

ABSTRACT

Without question, we are now living in the era of data. It may surprise some people to learn that humans have been utilizing data to inform decisions since the dawn of time. Every day, we use data to judge seemingly uncomplicated things like what to dress depending on the current weather and how to go to work based on traffic reports. As a result of ongoing technological advancements, a vast quantity of data is currently being generated, gathered, stored, and analyzed. Furthermore, technology has also advanced over the years to provide us with the means and tools we need to collect, store, display, comprehend, and apply data to develop valuable forecasts that will aid in resolving real-world problems. Additionally, machine Learning, a field that has grown so fast recently, relies on computers to analyze and understand the data given to them to predict results.

The financial distress forecast problem is essential in the financial sector because it has consequences on banks, companies, and organizations and is the primary subject of this thesis. Poor financial distress projections may result in significant financial losses. Thus, major attempts have been made to create prediction models to aid in improving such activities by assisting decision-makers in foreseeing incidents prior to they happen and preventing the company from going bankrupt.

From a machine learning perspective, financial distress forecasting is viewed as a binary classification issue, where the data is usually highly imbalanced, meaning that the vast majority of companies are solvent, while only a tiny number are insolvent, making it a challenging task. As a result, various algorithms and techniques have been created in the past years to classify imbalanced datasets. Three main techniques for learning from imbalanced data may be recognized: data-level techniques, also known as external methods that modify the distributions of the instances and maybe exclude problematic samples. Techniques that modify existing learning algorithms, sometimes called internal techniques, to mine data with skewed distributions and lessen their bias towards majority instances. Furthermore, hybrid strategies combine the advantages of the two earlier techniques.

In this thesis, we tackled external methods and internal methods separately. In the case of external methods: We attempt to improve the financial distress prediction models' ability to forecast failure by addressing the uneven distribution issue. We specifically concentrate on implementing and contrasting eleven advanced resampling techniques to preprocess the data to lower their imbalance ratio. Following the

data balance, we create the decision trees to forecast financial distress using the C4.5 classifier. For this study, a real dataset that was gathered from the Spanish market was used. Due to the dataset's extremely imbalanced distribution, where insolvent cases make up only 2% of the entire sample, it is thought to be exceedingly tricky. We observed a substantial improvement regarding the evaluated evaluation measurements, hence a decrease in the misclassification of positive occurrences, which is thought to be the most significant risk factor.
In contrast, regarding internal methods, artificial neural networks based on metaheuristic optimization have shown impressive results in various applications, including classification problems. More thought has yet to be devoted to using a metaheuristic optimization-based artificial neural network with a cost-sensitive fitness function to address the challenge of predicting a financial crisis. This thesis proposes a novel ENS_PSONN$_{cost}$ and ENS_CSONN$_{cost}$; metaheuristic optimization-based artificial neural networks that utilize the particle swarm optimizer and competitive swarm optimizer with a cost-sensitive fitness function, using five of these as the foundation for a majority-voting ensemble learning approach. In order to prevent dataset bias, three extremely imbalanced datasets of Polish, Taiwanese, and Spanish enterprises were considered. The g-mean (geometric mean of sensitivity and specificity) measure and the f1-score (harmonic mean of precision and sensitivity) measure demonstrated considerable improvement in the findings while retaining sufficient accuracy.

RESUMEN

Actualmente vivimos en la era de los datos. Puede que a algunos les sorprenda saber que el ser humano lleva utilizando datos para tomar decisiones desde la noche de los tiempos. Todos los días utilizamos datos para juzgar cosas aparentemente sencillas, como qué ropa elegir en función del tiempo que hace o cómo ir al trabajo en función del tráfico.
Gracias a los continuos avances tecnológicos, actualmente se genera, recopila, almacena y analiza una enorme cantidad de datos. Además, la tecnología también ha avanzado a lo largo de los años para proporcionarnos los medios y las herramientas que necesitamos para recopilar, almacenar, mostrar, comprender y aplicar datos para desarrollar predicciones que ayuden a resolver problemas del mundo real. Además, el aprendizaje automático, un campo que ha crecido tan rápidament e en los últimos tiempos, se basa en el análisis de datos computacionalmente para predecir resultados.
El problema de la predicción de la quiebra financiera es esencial en el sector empresarial porque tiene consecuencias en bancos, empresas y organizaciones, siendo el tema principal de esta tesis. Una mala previsión de las dificultades financieras puede acarrear importantes pérdidas económicas. Por ello, se han hecho grandes intentos de crear modelos de predicción que ayuden a mejorar los resultados, ayudando a los responsables de la toma de decisiones a prever incidentes antes de que ocurran y evitando que la empresa entre en quiebra.
Desde el punto de vista del aprendizaje automático, la predicción de dificultades financieras se considera un problema de clasificación binaria, en el que el conjunto de datos suele estar muy desequilibrado, lo que significa que la gran mayoría de las empresas son solventes, mientras que sólo un número ínfimo son insolventes. Esto lo convierte en un problema muy difícil. Por ello, en las últimas décadas se han creado diversas técnicas y algoritmos para clasificar conjuntos de datos desbalanceados. Se pueden reconocer tres técnicas principales para el aprendizaje a partir de datos desbalanceados: Técnicas a nivel de datos, también conocidas como métodos externos que modifican las distribuciones de las instancias y pueden excluir muestras problemáticas; técnicas que modifican los algoritmos de aprendizaje existentes, a veces denominadas técnicas internas, para extraer datos con distribuciones sesgadas y disminuir su sesgo hacia las instancias mayoritarias; y por último, estrategias híbridas combinan las ventajas de las dos técnicas anteriores.

En esta tesis abordamos los métodos externos y los métodos internos por separado. En el caso de los métodos externos se busca mejorar la capacidad de los modelos de predicción de quiebra empresarial para predecir la quiebra abordando el problema de la distribución desigual. En concreto, nos centramos en aplicar y contrastar once técnicas avanzadas de remuestreo para preprocesar los datos con el fin de reducir su ratio de desbalanceo. Tras el balanceo de los datos, creamos los árboles de decisión para predecir la quiebra empresarial utilizando el clasificador C4.5. Para este estudio se utilizó un conjunto de datos reales procedentes del mercado español. Debido a la distribución extremadamente desbalanceada del conjunto de datos, en el que los casos insolventes representan sólo el 2% de toda la muestra, se considera que es un problema muy complicado. De los resultados obtenidos observamos una mejora sustancial con respecto a las medidas de evaluación, que significa una disminución del error en clasificación de ocurrencias positivas, lo que supone el factor de riesgo más significativo.

Por otra parte, en el caso de los métodos internos, las redes neuronales artificiales basadas en la optimización metaheurística han mostrado notables resultados en diversas aplicaciones, incluidos los problemas de clasificación. Aún no se ha reflexionado más sobre el empleo de una función de adecuación sensible a los costes en las redes neuronales artificiales basadas en la optimización metaheurística para abordar el reto de predecir la quiebra empresarial.

En esta tesis se proponen dos nuevos modelos predictivos complejos, llamados ENS_PSONN$_{cost}$ y ENS_CSONN$_{cost}$. Específicamente se trata de redes neuronales artificiales basadas en la optimización metaheurística que utilizan el optimizador de enjambre de partículas (particle swarm optimizer, PSO) y el optimizador de enjambre competitivo (competitive swarm optimizer, CSO) con una función de adecuación sensible al coste. Cada conjunto de predictores (”ensemble”) está compuesto por cinco modelos en un paradigma de aprendizaje de votación mayoritaria.

Para evitar sesgos en los conjuntos de datos se consideraron tres conjuntos de datos extremadamente desbalanceados de empresas españolas, taiwanesas y polacas. Usando las medidas g-mean (media geométrica de la sensibilidad y la especificidad) y f1-score (media armónica de la precisión y la sensibilidad) se ha conseguido una mejora considerable en los resultados obtenidos.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

PRELIMINARY & BACKGROUND

1

INTRODUCTION

CONTENTS

Machine learning has been extensively used in solving the problem of forecasting corporate financial distress since it significantly influences the future of business, which is an actual example of real-world problem-solving. As a result, this Ph.D. thesis investigates the challenge of forecasting the financial distress of organizations and suggests fresh ideas to enhance the accuracy and general efficiency of machine learning algorithms in tackling it. Notably, the primary barrier to addressing this issue is the rarity of financial distress in firms, which presents a significant hurdle for the classifiers in determining the actual financial situation of the firms (i.e., insolvent or solvent) and ensuring a high level of validity accordingly.

In the literature, the terms bankruptcy and insolvency are commonly used synonymously [1]. In a legal-financial process known as bankruptcy, a person or business declares that it cannot meet its financial commitments. This legal stance has the result that part of the obligations will be paid off by selling off the debtor's assets, while the remaining debts will be disregarded [2].

Therefore, predicting financial distress (i.e., insolvency or bankruptcy) is a crucial application in the financial sector that helps make intelligent business decisions [3]. A more comprehensive picture of the business's health will be provided by successfully forecasting this problem, which will also help decision-makers foresee events before they occur.

1.1 CONTEXT AND MOTIVATION

Because of its repercussions on banks, businesses, and a wide range of stakeholders, including employees, customers, vendors, and ultimately whole countries, the importance of enterprises' financial distress prediction problem is apparent in today's world. Due to poor judgment and analysis, there may be significant financial losses.

Giving a specific approach to deal with financial data is, therefore, a challenging task because there is typically a high degree of imbalance in the distribution of financial data; that is, the number of companies in financial distress is much lower than that of successful ones, and the occurrence of the financial distress of companies is rare compared to solvency in the real world. In other words, there is a significant bias in the firm's financial data, which shows the urgent need to apply resampling methodologies to address the issue since the asymmetric distribution of the data has a detrimental impact on the accuracy and overall performance of classifiers in predicting the financial collapse of firms. A classifier is a piece of software or programming that represents an algorithm's processing of data observations or samples to identify the class of new ones [4].

Datasets about financial distress present a challenge because of their extreme imbalance. The dataset is unbalanced if samples from one class are much more numerous than samples from the other classes. Classifiers may have excellent accuracy for the majority class but inferior accuracy for the minority class due to the stronger majority class's impact on traditional training criteria. Reducing the error rate or the percentage of incorrect class label predictions is the main objective of most novel classification algorithms [5].

Given the importance of financial distress prediction and the imbalanced nature of the data, this doctorate thesis aims to make relevant contributions to improving machine learning algorithms in predicting companies' financial distress by first evaluating the effect of oversampling techniques. Second, develop a new cost-sensitive metaheuristic-optimized artificial neural network and then assess the impact of ensemble learning on the newly developed framework.

1.2 OBJECTIVES

The thesis aims to make valuable contributions to enhancing machine learning algorithms' overall prediction capability in predicting companies' financial distress, considering the critical factors outlined in the previous Sect. 1.1; the specific key goals are listed in further detail below.

1.2.1 *Evaluate advanced oversampling methods*

The external technique is the one that is most typically employed to address the imbalanced aspect of financial distress prediction datasets. Boosting the minority class number, known as *Oversampling*, or lowering the majority class ones, referred to as *Undersampling*, modifies the distribution of the class labels [6].

In this Ph.D. thesis, we experimentally assess and contrast eleven advanced oversampling techniques for predicting financial distress. For this study, an actual dataset that was gathered from the Spanish market was used.

1.2.2 *Develop a new cost-sensitive metaheuristic-optimized artificial neural network*

Cost-sensitive learning and metaheuristic-optimized artificial neural networks have demonstrated promising outcomes for classification problems. Despite this, the impacts of integrating the cost-sensitive fitness function with a metaheuristic-optimized artificial neural network on financial distress prediction have received relatively little attention.

A subset of Artificial Neural Networks (ANNs) known as Metaheuristic Optimized Artificial Neural Networks (MHOANNs) uses metaheuristic optimization techniques to choose its weights and biases [7]. Swarm-based algorithms have been designed as a robust collection of optimization methods, and they were motivated by social animals' group dynamics. A group of particles that navigate through the parameter space, creating paths based on their own and their peers' optimal performances, is how Particle Swarm Optimization (PSO) describes the gathering of possible solutions to the optimization problem, referred to

as the particle model[8]. In contrast, Competitive Swarm Optimization (CSO) is a more contemporary form of PSO that incorporates a competition mechanism that involves two particles, where the looser particle learns from the winning particle and adjusts its position accordingly [9].

We suggest employing a cost-sensitive MHOANN to do this in order to enhance the predictions of the minor class in the financial distress dataset. The cost-sensitive component is utilized to enhance the minority class prediction. PSO and CSO were the two optimization methods that were selected. PSO and CSO were chosen as the optimization strategies in this thesis because, in contrast to other metaheuristic approaches algorithms, PSO only needs a few parameters and hence requires fewer iterations [10]. Moreover, CSO is a more modern form of PSO intended as a solution for large-scale optimization issues due to the fact that each iteration only updates 50% of the particles [11].

1.2.3 *Apply ensemble learning on the new cost-sensitive Metaheuristic-optimized artificial neural network*

Combining multiple learning algorithms to achieve a higher level of predicted performance than what could be achieved by any single learning algorithm is what ensemble methods in statistics and machine learning are all about. Even while a machine learning ensemble merely consists of a tangible, limited set of distinct models, it often permits a significantly more adaptable framework within those alternatives. The fundamental tenet of ensemble learning is that by combining many models, the shortcomings of one model will almost certainly be offset by those of others [12].

The primary idea behind ensemble learning is that when integrating several models, the deficiencies of a single model can be made up for by the remaining models, producing one powerful learner from several weak ones. to accomplish this, the ensemble's weak learners should be diverse and accurate [12]. This thesis evaluated: if the Metaheuristic-optimized artificial neural network is accurate and diverse enough to be applied with an ensemble learning paradigm.

1.3 THESIS STRUCTURE

- **Chapter 1. Introduction**: In this chapter, we presented this thesis's context, motivation, and objectives.
- **Chapter 2. Background**: This chapter provides a comprehensive description and explanation of the algorithms used in this thesis.
- **Chapter 3. Literature Review**: In this chapter, we review the state-of-the-art machine learning methods used in the literature to handle imbalanced classification problems in general and financial distress prediction in specific.
- **Chapter 4. Methodology**: This chapter details the datasets, machine learning methods, experiments' setup, results, and analysis.
- **Chapter 5. Conclusions and future works**: This chapter summarizes the conclusions reached based on the findings and contributions of this Ph.D. thesis. Additionally, the suggested upcoming work is also revealed.

2

BACKGROUND

CONTENTS

This chapter comprehensively describes and explains the methods and algorithms used in this Ph.D. thesis.

2.1 IMBALANCED DATASETS CLASSIFICATION METHODS

Learning from imbalanced data is still a significant study area after more than two decades of advancement. This issue has developed well beyond its original notion as an issue of unbalanced distributions in binary tasks since we have obtained a greater understanding of the nature of unbalanced learning because of the development of machine learning and data mining and the emergence of the era of big data while confronting new obstacles [13].

According to traditional machine learning techniques, the count of occurrences in the classes under consideration is often comparable. However, the distribution of samples is occasionally distorted since individuals from particular groups occur much more frequently in actual situations. There will be a problem since learning algorithms will be skewed in favor of the majority group. From the perspective of data mining, the minority class is often the one that is more important since, despite its rarity, it may contain essential and pertinent information [13].

Classifying unbalanced data usually challenges these conventional classification methods, even though machine learning techniques have been widely employed to create classification models to support administrative and commercial decision-making. Generally, there are five reasons for this [14]:

SVM: Support Vector Machine

- Balanced training sets work well with standard classifiers like logistic regression, SVM, and decision trees. When faced with unbalanced circumstances, these models frequently produce subpar classification results, i.e., they cover the majority class well while distorting the minority ones.

- Even though the forecast model attains a high overall accuracy, the training process is inclined towards the more prevalent class when global performance measures like prediction accuracy drive it.

- The learning model could consider rare minority examples as noise. On the other hand, noise could be mistakenly classified as minority occurrences since both patterns are uncommon in the data space.

- Minority instances typically overlap with other areas where the initial probability of both classes is nearly the same, despite the

fact that unbalanced sample distributions are not necessarily challenging to learn (such as in situations where the classes can be differentiated).

- Additionally, difficulties with unbalanced learning, such as tiny disjuncts, an insufficiency of concentration, and limited sample sizes with high feature complexity, frequently result in learning models failing to discover unusual patterns.

Over the last few decades, a number of machine-learning techniques have been created to cope with imbalanced data categorization. These tactics primarily rely on ensemble approaches, cost-sensitive learning, and sampling procedures. A simple solution to this problem is the resampling strategy, which involves either raising the number of records in the minority class or lowering the number in the majority class. Two resampling techniques that are frequently employed include oversampling and undersampling [15].

Sampling-based approaches offer the data-level solution by evenly distributing the samples among the classes. The two basic sampling techniques, undersampling and oversampling, involve taking fewer samples from the majority class or including more samples from the minority class [16]. In this Ph.D. thesis, we will study the oversampling technique by evaluating eleven oversampling methods.

On the other hand, cost-sensitive classification works at the algorithm-level. It aims to reduce misclassification in the minority class by considering that the costs caused by different kinds of misclassifications are not considered equal [7]. This thesis used an MHOANN as our classifier based on PSO and CSO optimizers with a cost-sensitive fitness function. Then, utilizing ensemble learning with homogenous majority voting, we enhanced the capabilities of our model.

2.2 OVERSAMPLING METHODS

Methods of oversampling replicate instances in the minority class or create new instances using examples from the minority class in order to to reduce the imbalanced distribution of a given dataset. The following is a succinct explanation of the oversampling techniques evaluated in this Ph.D. thesis.

2.2.1 *Random Over-Sampling (ROS-I)*

ROS-I is a simple over-sampling technique that is comparable to more advanced over-sampling methods. By randomly replicating existing data points, ROS-I is a non-heuristic method for balancing an unbalanced data set by raising the proportion of minority class members. Furthermore, the average increase in the mean number of induced rules is often the least compared to other oversampling methods. Moreover, it yields valuable results and is computationally less costly than alternative approaches. Though straightforward and practical, this strategy is highly susceptible to overfitting because it produces identical duplicates of the minority class instances [17].

2.2.2 *Synthetic Minority Oversampling Technique (SMOTE)*

The technique most frequently used for producing new examples is SMOTE. It is preferable to synthesize fresh instances from the minority class as opposed to using examples from the same class repeatedly. SMOTE increases the diversity of the training data by incorporating additional instances, which helps to even out any irregular concentrations of data points. It does this by randomly selecting the k closest instances from the minority class. The process then calculates the gap between the minority class samples and its closest neighbor, scales it by a random factor within the range of 0 and 1, and then incorporates it into the feature vector. It generates synthetic samples on the line connecting any or all of the k minority class nearest neighbors.

The approach is effective as it creates authentic synthetic samples from the minority class that are close in feature space to existing minority class instances. However, the synthetic samples are generated independently of the majority class, which could lead to confusing examples if the two classes have a high degree of overlap which has been highlighted in [18].

2.2.3 *Synthetic Minority Oversampling Technique and Tomek's Links (SMOTE-TL)*

SMOTE-TL is a sophisticated oversampling method designed to address the overfitting issue and eliminate noisy examples on the incorrect area

of the boundary of the decision. It cleans the data using the "Tomek links" technique. In the beginning, SMOTE is used to replicate minor class instances. After that, Tomek links are found and deleted for instances from both the minor and the major classes. The following is a definition of Tomek links: If two samples are from separate classes and are the closest to one another, they are said to have a Tomek link [17].

2.2.4 *Synthetic Minority Oversampling Technique and Edited Nearest Neighbor (SMOTE-ENN)*

SMOTE-ENN employs SMOTE to generate synthetic examples, and then it will clean the data by reducing the noisy samples to improve the classifiers' generalization capacity. The cleanup procedure is described as follows: If two or more of the three closest neighbors, for an instance E, are from a minority class and E belongs to the majority class, then E is deleted and contrariwise applied to instances from the majority class [17].

2.2.5 *Borderline SMOTE*

Most classification methods attempt to distinguish the boundaries for every class during the training phase. In general, borderline cases are usually instances that are frequently incorrectly classified. Consequently, this technique concentrates on these instances. This technique operates in the following manner: for each instance E belongs to the minority class, it locates its k nearest neighbors; if they all belong to the majority class, the sample is treated as noise and is not taken into account. The instance is regarded as safe and also disregarded if there are fewer instances of the majority class in the k nearest neighbors than there are of the minority class. The rest, which have more neighbors who are members of the majority class, are deemed dangerous, and these are the examples that are artificially repeated [19].

2.2.6 *Safe-Level SMOTE*

Solely seeks to produce artificial instances in safe areas. It operates as follows: It calculates the safe level ratio s for minor instance p, described as the number of minority class instances within the k nearest neighbor.

After that, it divides the safe level by the safe level of the nearest neighbors n to get the safe level ratio for the instance p. There will be five different scenarios:

- When s is ∞, and p's safe level equals 0, p and n are both treated as noise and disregarded.
- When s is ∞, and p's safe level does not equal 0, n is considered noise, and a synthetic sample is created away from it.
- When s equals 1, a synthetic sample is generated between p and n.
- When the safe level is more than 1, p is considered safer than n. Hence a synthetic sample is generated near p.
- On the contrary, when s is less than 1, the synthetic sample will be created close to n as it is considered safer than p [20].

2.2.7 *Adaptive Synthetic Sampling (ADASYN-I)*

ADASYN-I employs weights to assess unpredictable minority class occurrences. This method calculates the density distribution by the k nearest neighbors for each instance in the minority class identified. The number of cases in the k nearest neighbors that correspond to the majority class is then divided by k to determine the density distribution. This assessment of weight distribution for various minority classes is considered to calculate the required number of synthetic examples, which lessens prejudice. It moves the classification decision boundary to the challenging samples [21].

2.2.8 *Adjusting the Direction Of the synthetic Minority class examples (ADOMS)*

It uses the principal component analysis approach, concentrating on the dataset's fluctuations and regularities. Altering the orientation of the artificially generated minority class samples generates synthetic samples that are well-aligned with the true distribution of the dataset. The first principal component of the feature space that captures the most significant proportion of the variance in the local data distribution

will be utilized to generate synthetic examples. This axis is a linear combination of all features with the most significant variation concerning all other linear combinations. It effectively reduces the experimental classifier's classification performance in class imbalance scenarios and aids in lowering problems brought on by freshly created synthetic samples for the minority class [22].

2.2.9 *Selective Preprocessing of Imbalanced Data (SPIDER)*

A method for selectively preparing unbalanced data, it divides instances into two categories—safe and noisy—based on their inherent characteristics. A built-in classifier will classify safe samples correctly, but noisy examples have a high possibility of misclassification, necessitating preprocessing. By combining HVDM with NNR, the sample is classified as either safe or noisy [23].

HVDM: Heterogeneous Value Distance Metric
NNR: Neural Network Regression

2.2.10 *SPIDER-II*

This technique, which distinguishes between safe, borderline, and noisy samples, makes the same argument as SPIDER that the presence of marginal and noisy examples hinders the learning process for algorithms. The SPIDER-II method utilizes a two-step preprocessing technique for examples that are a part of the minor and major classes, in contrast to the SPIDER method, which processes the minority and majority examples simultaneously, which could result in extreme changes in certain areas of the examples that are a part of the majority class [24].

2.2.11 *Agglomerative Hierarchical Clustering (AHC)*

AHC is a technique employed in data mining for grouping similar examples together. The technique starts by treating each sample as an individual cluster, then successively calculates the closest distance between two clusters, combines them, and continues this process for all data points. This process will be continued until the desired number of clusters is reached. The resulting clusters are used to create synthetic data points to address imbalances in the original dataset. This technique helps to improve the sensitivity of various classification algorithms.

Specifically, it works by removing the initially combined clusters from the dataset, adding the new cluster, and repeating the procedure [25].

2.3 COST-SENSITIVE LEARNING

By imposing higher penalties for improperly classifying examples from the minority class instances relative to majority class instances, cost-sensitive learning may be implemented at both the algorithmic level and the data level (such as resampling and feature selection) [26].

The costs are sometimes expressed as cost matrices, where C_{ij} stands for the misclassification cost of allocating instances to class i when they should be in class j. Cost matrices can be established for a given domain using expert judgment. In scenarios involving data streams, they can fluctuate for each record or vary in a dynamic unbalanced condition [27].

Cost-sensitive learning is less computationally demanding than resampling techniques, making it potentially more appropriate for massive data streams. However, compared to resampling procedures, this technique is far less used. There may be two possible causes, one of which is that setting values in the cost matrix is challenging. Most of the time, an expert cannot estimate the cost of misclassification since it cannot be determined from the data. However, another approach may be used to solve this problem. It involves setting the majority class misclassification cost to 1 and the penalty minority class value to the IR [28]. Resampling is a popular option for researchers with limited domain knowledge. The second cause is that resampling techniques are significantly more straightforward to use in single and ensemble models than cost-sensitive learning, which frequently necessitates changing the learning algorithm [14].

IR: Imbalance Ratio

2.4 METAHEURISTIC OPTIMIZED ARTIFICIAL NEURAL NETWORKS (MHOANNS)

ANNs are machine learning algorithms influenced by human brain function and structure. They are made of interconnected nodes known as neurons arranged in layers and are trained to perform various tasks by adjusting the strengths of the connections (or weights) between the nodes. ANNs have been widely used in various applications, including

pattern recognition, data classification, and function approximation [29].

One of the challenges in training ANNs is finding the optimal set of weights that will give the best performance on a given task. This problem is often referred to as the training or optimization problem. It can be challenging due to a large number of weights that need to be adjusted, the complex and non-linear nature of the relationships between the weights and the performance of the ANN, and the presence of local minima in the error surface [30].

A subset of ANNs known as MHOANNs uses metaheuristic optimization techniques to choose its weights and biases [7]. Swarm-based algorithms have been created as a powerful collection of optimization techniques, and they were inspired by the group behavior of social animals. A swarm of particles that can move through the parameter space and create trajectories motivated by their own and their neighbors' best performances is how PSO describes the collection of potential solutions to the optimization problem [8], which is referred to as the particle model. A mechanism that involves competition between pairs is used in CSO, a more contemporary form of PSO, where the losing particle learns from the winning particle and adjusts its position [9].

In addition to these traditional metaheuristics, more recent studies have been dedicated to creating hybrid methods that combine metaheuristics with other optimization techniques, such as gradient-based methods and particle swarm optimization [31, 32]. These hybrid algorithms are particularly effective at finding reasonable solutions to the optimization problem in a relatively short time while still being able to avoid getting stuck in local minima and cover a large area of the search space.

Overall, metaheuristics-based optimization algorithms have proven to be a valuable tool for training ANNs and have been widely used in various applications. As research in this area continues to evolve, we will likely see the development of even more effective and efficient metaheuristics-based optimization algorithms for ANNs in the future.

2.4.1 *Metaheuristic optimization algorithms*

There are numerous essential applications for optimization, and the methods used to do it are diverse and have a wide variety of practical uses. Modern metaheuristics are gaining popularity among these

optimization techniques, giving rise to a new subfield known as metaheuristic optimization.

Metaheuristic optimization algorithms are a set of algorithms that are used to solve complex optimization problems. These algorithms are designed to be flexible and adaptable and can be applied to various optimization problems in various fields, including engineering, computer science, and finance [33]. Metaheuristic algorithms are often used when traditional optimization methods, such as gradient descent or linear programming, are insufficient to find a problem's optimal solution. They are beneficial for solving problems that are large, complex, or noisy or that involve multiple objectives or constraints.

One of the main advantages of metaheuristic algorithms is their ability to find reasonable, near-optimal solutions quickly. They are also relatively easy to implement and can be applied to problems with little or no prior knowledge of the problem domain [34].

Most metaheuristic algorithms, including ant colony optimization, simulated annealing, particle swarm optimization, and cuckoo search, are derived from natural phenomena. Since the advent of swarm intelligence algorithms such as PSO in the 1990s, over a dozen new metaheuristic algorithms have been created. These algorithms have been applied in various fields, such as optimization, data mining, machine intelligence, and many other domains. Numerous books and countless research articles have been published [33]. In this Ph.D. thesis, we used PSO and CSO as the optimization methods for the MHOANN.

However, metaheuristic algorithms have some limitations, and they may only sometimes find the optimal global solution to a problem and require more computational resources than traditional optimization methods. Overall, metaheuristic optimization algorithms are a valuable tool for solving complex optimization problems and can be an effective alternative to traditional optimization methods in many cases [35].

2.4.1.1 *Particle Swarm Optimization (PSO)*

PSO is an optimization method that works incredibly well for issues where the parameter's best solution is a point in a multidimensional space. The social interaction and communication found in fish schools and flocks of birds served as inspiration (real-valued optimization). The characteristic of the particles being described by a position and a velocity, as well as having the ability to change their place in the

search space, is translated from inspiration from natural analogs, such as schooling or flocking [8].

A collection of potential solutions, referred to as particles or agents, are first generated at random by the algorithm. A mathematical formula that regulates the position and velocity of the particles directs how they move throughout the search space over a series of iterations that make up the optimization process. The most advantageous solution that each particle has discovered affects the direction that particle is moving in. They are directed toward the search space's known optimal places, which are updated when other swarm particles find new, more advantageous sites. As a result, the entire swarm gradually moves in the direction of the ideal solution [36].

The velocity of every particle is computed in every iteration as stated by equation 2.1, where the particle P_i's velocity in dimension $d = 1, ..., n$ at time step t is denoted by $v_{id}(t)$, w defined as inertia keeps track of the prior flow direction to stop the particle from abruptly changing direction, r_1, and r_2 are two randomly chosen numbers $\in [0, 1]$ within a uniform distribution. Positive constants called acceleration constants are c_1 and c_2, $p_{id}(t)$ is the ith particle best position known for this particle recorded since the first iteration in dimensions d at t time. The best global particle location is donated by p_g.

After that, each particle's position is updated in accordance with equation 2.2, where $x_{id}(t)$ represents the particle's position and $v_{id}(t+1)$ represents the velocity of the ith particle in dimension d at time step $t+1$ [36]. Figure 2.1 shows the pseudocode for the standard PSO algorithm.

$$\begin{aligned} v_{id}(t+1) = w \,.\, v_{id}(t) \\ + r_1 c_1 \,.\, [p_{id}(t) - x_{id}(t)] \\ + r_2 c_2 \,.\, [p_g - x_{id}(t)] \end{aligned} \tag{2.1}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{2.2}$$

2.4.1.2 *Competitive Swarm Optimizer (CSO)*

It is a PSO-based technique, but it differs significantly from PSO. In CSO, the particles are not updated using either the particle's best

Initialize population
for $t = 1$: maximum generation
  for $i = 1$: population size
    if $f\left(x_{i,d}(t)\right) < f\left(p_i(t)\right)$ **then** $p_i(t) = x_{i,d}(t)$
      $f\left(p_g(t)\right) = \min_i\left(f\left(p_i(t)\right)\right)$
    end
    for d =1 : dimension
      $v_{i,d}(t+1) = wv_{i,d}(t) + c_1 r_1\left(p_i - x_{i,d}(t)\right) + c_2 r_2\left(p_g - x_{i,d}(t)\right)$
      $x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1)$
      if $v_{i,d}(t+1) > v_{\max}$ **then** $v_{i,d}(t+1) = v_{\max}$
      else if $v_{i,d}(t+1) < v_{\min}$ **then** $v_{i,d}(t+1) = v_{\min}$
      end
      if $x_{i,d}(t+1) > x_{\max}$ **then** $x_{i,d}(t+1) = x_{\max}$
      else if $x_{i,d}(t+1) < x_{\min}$ **then** $x_{i,d}(t+1) = x_{\min}$
      end
    end
  end
end

Figure 2.1: Pseudocode of the standard PSO algorithm [37].

position or the global best position. Rather, a pair-by-pair competition process is used, in which the losing particle changes its position by learning from the winning particle. Despite its straightforward approach, the CSO exceeds the most recent metaheuristic algorithms in terms of overall performance [9]. A diagram that explains the CSO mechanism is shown in Figure 2.2.

Each particle in CSO stands for a potential answer to the optimization problem. Let a swarm of m particles is represented by $P(t)$ where m is the number of the particles, and t represents the index of iteration or generation. Until all particles have participated in at least one competition, two particles are chosen randomly from $P(t)$ in each generation, assuming that m is an even number.

In each generation of CSO, two particles are picked at random from the current population $P(t)$ and compared by calculating their fitness. If the number of particles in the swarm is even, this process continues until all particles have competed at least once. The winning particle is carried over to the following generation, $P(t+1)$, and is the one with the highest fitness. In contrast, the losing particle is carried over after it updates its position based on the winner's knowledge.

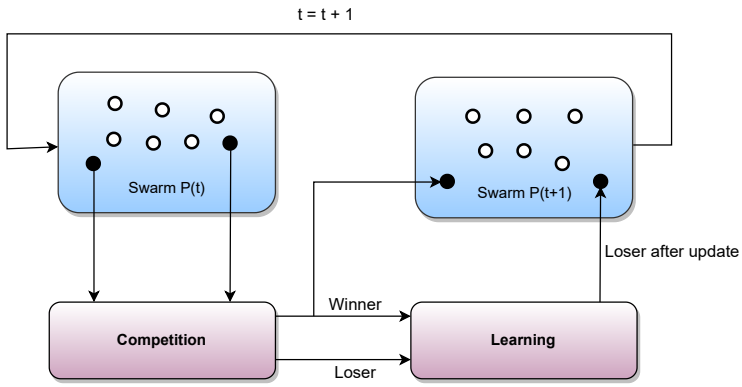

Figure 2.2: A diagram that explains the general idea of CSO. Particles from the current swarm are pairwise and randomly chosen during each generation to compete. The winner is immediately carried over to the swarm of the subsequent generation. At the same time, the one with the low fitness value is improved by gaining knowledge from the winner after each competition.

Learning from the winner is simply updating the losing particle's position according to equation 2.3, where $x_{l,i}(t+1)$ is the losing particle's position in the next generation, $x_{l,i}(t)$ is the current losing particle's position of and $v_{l,i}(t+1)$ is the updated velocity after learning from the winner particle.

Equation 2.4 explains updating the velocity of the losing particle, where $x_{w,i}(t)$ is the winner particle's position in the i-th round of competition in iteration t, $x_{l,i}(t)$ is the loser particle's position in the i-th round of competition in generation t, $v_{w,i}(t)$ is the winner particle's velocity in the i-th round of competition in generation t, $v_{l,i}(t)$ is the loser particle's velocity in the i-th round of competition in generation t, $i \in 1, 2, ..., m/2$, m is the number of particles or the population size, $r_1(i,t)$, $r_2(i,t)$ and $r_3(i,t) \in [0,1]$ are three randomly generated vectors at the i-th competition and learning process in generation t, $\bar{x}(t)$ is the mean position value of all particles, which is considered the center of the swarm in generation t, and φ is a parameter used to control the effect or influence of $\bar{x}(t)$ [9].

$$x_{l,i}(t+1) = x_{l,i}(t) + v_{l,i}(t+1) \tag{2.3}$$

$$
\begin{aligned}
v_{l,i}(t+1) = {} & r_1(i,t)v_{l,i}(t) \\
& + r_2(i,t)(x_{w,i}(t) - x_{l,i}(t)) \\
& + \varphi\, r_3(i,t)(\bar{x}(t) - x_{l,i}(t))
\end{aligned}
\tag{2.4}
$$

2.5 ENSEMBLE LEARNING

By training several models and combining their predictions, ensemble learning is frequently regarded as the most sophisticated solution to various machine learning problems. It is a general term for techniques that combine different inducers to get a conclusion and is widely applied in supervised machine learning scenarios. A model (such as a classification or a regression model) is built using an inducer approach, also known as a base learner, which accepts a collection of labeled instances as input. Recent samples without labels may be predicted using the constructed model. As ensemble inducers, a variety of machine learning approaches may be applied (e.g., neural network, random forest, Logistic regression). The underlying tenet of ensemble learning is the ability to compensate for the shortcomings of one model by the rest of the models in the ensemble, enhancing the ensemble's overall performance in terms of prediction [38].

There are several types of ensemble learning methods, including bagging, boosting, and bootstrapping. Bagging entails training several models on various subsets of the training data and averaging their forecasts to arrive at the final forecast. Boosting entails successively training several models, each of which tries to fix the mistakes caused by the one before it. On the other hand, Bootstrapping entails using various subsets of the training data to train different models drawn with replacement and averaging their predictions to make the final prediction [39].

Members of an ensemble might be of the same type or different kinds, and they could or might have been trained using a different training dataset. It is considered homogenous if each learner in an ensemble belongs to the same kind. For instance, a "neural network ensemble" consists solely of neural networks [40].

Combining the results of all base learners can be done in the classification case by three different methods of majority voting: (1) unanimous voting, where all classifiers agree on the prediction; (2) simple majority, where at least one greater than 50% of the number of models predicted

the same label; (3) plurality voting, the forecast with the most votes wins [41]. Figure 2.3 shows a graphical abstraction of an ensemble diagram that includes five different decision trees in a majority voting scheme.

Ensemble learning has been demonstrated to be particularly useful in enhancing the performance of machine learning models in various tasks, including classification, regression, and clustering. For example, in the area of classification, ensemble learning is effective in improving the performance of decision trees, neural networks, and support vector machines [39]. In the area of regression, ensemble learning has been shown to improve the performance of linear and nonlinear regression models. Moreover, ensemble learning proved to be effective in enhancing the performance of clustering methods, k-means, and hierarchical clustering, for example [42].

For several reasons, ensemble approaches frequently enhance prediction performance. Some of the reasons can be summarized as follows: Avoiding overfitting, when there is limited data, a learning algorithm is more likely to develop a variety of hypotheses that accurately predict well for the training data while producing inaccurate predictions for unobserved cases. Averaging different hypotheses improves overall prediction performance since it reduces the possibility of choosing a wrong hypothesis. By combining many learners, ensemble techniques lessen the likelihood that the overall model will get trapped in local minima. The perfect hypothesis could exist outside of the scope of any specific model. Moreover, combining several models can improve the fit to the data space, which broadens the search space [43, 44].

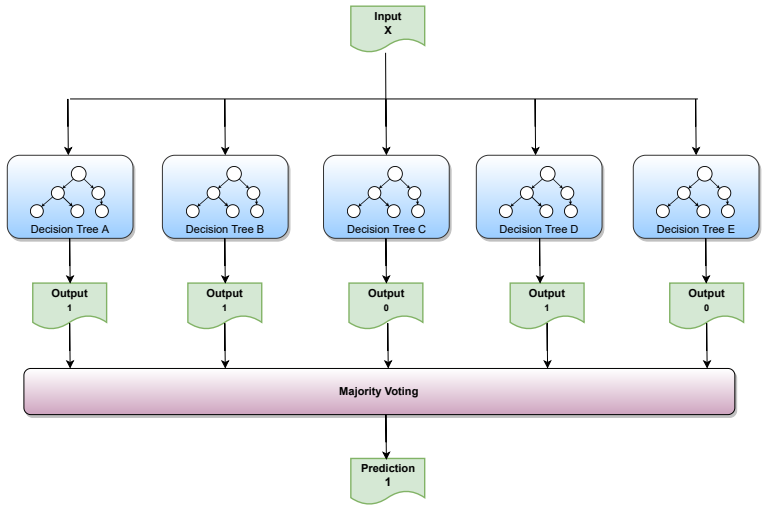

Figure 2.3: An ensemble of five classification trees: For the input x, each tree offers its classification, and instance x should be classified as "1" using majority voting, as three of the five trees voted for it.

Part II

CASE STUDY & DISCUSSION

3

LITERATURE REVIEW

CONTENTS

This chapter presents a literature review of various machine learning methods used to tackle the imbalance classification problem and companies' financial distress prediction.

Many scholars focus on the crucial problem of financial distress prediction since an incorrect assessment of a company's financial health might result in significant losses [45]. Predicting financial distress is difficult since the information is typically heavily skewed, meaning that the number of insolvent enterprises is far lower than the number of solvent ones leading to an imbalanced binary classification problem [46].

3.1 OVERSAMPLING METHODS

The oversampling technique is prevalent in tackling the class imbalance problem in binary classification and, indeed, in financial distress prediction. For instance, García *et al.* [47] compared the effectiveness and performance of several undersampling and oversampling procedures used prior to categorizing various unbalanced credit datasets. Then they utilized four categorization techniques that are frequently used to forecast credit risk (KNN, MLP, RBF, and SVM). Five real-world credit datasets with different balance ratios have been considered to assess the resampling procedures and classification. The findings of this study demonstrate that the (SMOTE-ENN algorithm and SVM classifier) produce the best outcomes.

KNN: K-Nearest Neighbors
MLP: Multi Layer Perceptron
RBF: Radial Basis Function

Later, Jose *et al.* [48] proposed using resampling with filtering criteria to neglect borderline and noisy samples. In [49], Liang *et al.* investigated how the Financial Ratios (FRs) and Corporate Governance Indicators (CGIs) together affected the performance of the classifiers in forecasting the financial health of Taiwanese enterprises. A stratified sampling approach was used to choose a balanced subset and resolve the issue of uneven data distribution. The chosen subset included 239 records for firms that had gone bankrupt and another 239 records for still viable enterprises. Consequently, five classifiers—namely, SVM, KNN, CART, MLP, and Naïve Bayes—were compared. The performance of the classifiers was then enhanced by merging the FRs and CGIs with Stepwise Discriminant Analysis (SDA) feature selection technique and SVM, yielding the best outcomes.

CART: Classification And Regression Trees

ADASYN-I and SMOTE-ENN were used by Le *et al.* [50] to address the issue of categorizing very unbalanced datasets. They used several classifiers, including SVM, MLP, Random Forest, and Decision Trees. The authors' highly unbalanced dataset, which includes 120048 solvent and 307 bankrupt enterprises, was obtained from a Korean financial company. The findings of this study demonstrated that oversampling strategies might enhance bankruptcy prediction performance. The most remarkable findings for the Area Under the Curve (AUC) measurement came from SMOTE-ENN utilized as the random forest preprocessing step. Additionally, a novel sampling method to tackle the imbalance problem was introduced by González *et al.* [51], in which sampling is done inside monotonic chains.

Then, using a highly imbalanced dataset, Islam *et al.* [52] examined 13 classification algorithms to forecast bankruptcy. Data was prepossessed using SMOTE, and the evaluation metrics showed an increase in the

performance of the classification algorithms. In a recent study, a new method called Selective Oversampling Approach (SOA) was proposed by Gnip *et al.* [53] that distinguishes the most representative samples from minority groups using an outlier detection approach. Then, artificial oversampling is done with these samples. The results of the experiments showed that this method outperforms synthetic minority oversampling and adaptive synthetic sampling.

3.2 COST-SENSITIVE LEARNING

It is also quite common in the literature to use cost-sensitive learning to resolve the unbalanced classification problem. For example, the issue of estimating the likelihood of bankruptcy from an unbalanced dataset was addressed by Kim *et al.* [54]. They suggested an AdaBoost variation called the Geometric Mean-Based Boosting algorithm (GMBoost). The researchers took into account data gathered from a Korean commercial bank. In order to compare the generated findings with and without utilizing SMOTE, they divided the dataset into five subgroups. In contrast to AdaBoost, GMBoost produced the most significant results regarding prediction and learning capabilities.

ADABOOST: ADAPTIVE BOOSTING

Recently, Mienye *et al.* proposed in [55] that reliable cost-sensitive classifiers could be created when objective functions of algorithms like random forest, logistic regression, Extreme Gradient Boosting (XGBoost), and decision tree are updated. Then, accurate medical diagnoses are predicted using these algorithms. The research findings demonstrate that cost-sensitive methods outperform traditional algorithms.

Moreover, to address the binary classification of unbalanced class datasets. Hazarika *et al.* [56] conducted experimental analyses on several unique imbalanced synthetic and real-world datasets. Their effectiveness is evaluated using the geometric mean and the area under the curve. The outcomes are contrasted with SVM, fuzzy SVM, enhanced fuzzy least squares SVM, fuzzy SVM using affinity and class probabilities, and fuzzy least squares SVM using entropy. Similar or superior generalization outcomes demonstrate the effectiveness and applicability of the suggested methods.

3.3 ANN

Many researchers have used ANNs to predict financial failures. For example, a study by Pompe and Feelders [57] showed that ANNs performed better than Linear Discriminant Analysis (LDA) and classification trees for this case. Additionally, Baek and Cho [58] developed a training technique to enhance the effectiveness of an auto-associative neural network in predicting the financial collapse of organizations. They used a sample dataset that included bankrupt and solvent companies to train the network, which improved the network's performance compared to when it was trained on a dataset of only solvent companies.

Later, Bose and Pal [59] conducted research evaluating several financial forecasting techniques. They found that neural networks have a higher accuracy rate than SVMs.

After that, the effectiveness of various strategies in foretelling the financial disasters of Taiwanese publicly traded industrial companies was thoroughly addressed by Lin [60]. Multiple Discriminate Analysis (MDA) and ANNs, Probit, and Logit were the techniques that the study examined. The Taiwan Economic Journal dataset of public industrial businesses in Taiwan from 1998 to 2005 was used to train the classifiers. ANN, Probit, and Logit models produced outstanding findings in relation to prediction accuracy. The rest of the classifiers evaluated in that research outperformed Probit, demonstrating the most significant and consistent performance.

In a more recent study, Mansouri *et al.* [61] evaluated the effectiveness of three-layer ANNs and Logit in foretelling the collapse of firms financially registered on the Stock Exchange of Tehran one, two, and three years in advance. In order to anticipate a company's financial failure forecasted one, two, and three years ahead, the team stated that ANN surpassed Logit concerning accuracy.

3.4 MHOANN

Since the 1980s, training ANNs with evolutionary algorithms has gained much popularity. The Genetic Algorithm (GA) was employed by Montana *et al.* [62] to train an ANN for image categorization. How to train an ANN utilizing metaheuristic algorithms has also been widely

studied to overcome the drawbacks of gradient-based approaches, particularly back-propagation techniques.

The metaheuristic techniques employed in ANN training for binary classification tasks like bankruptcy prediction were the subject of a plethora of studies in the early 2000s. Mendes *et al.* [63] used the PSO method to train an ANN, and the findings showed that PSO has a substantial capacity to handle search spaces with many local minima.

Moreover, according to Ansari *et al.* [64], metaheuristic methods outperform gradient-based algorithms in terms of performance. Also, Al-Badarneh *et al.* [65] explore the impact of fitness functions in MHOANN while working on unbalanced data. In another research, Mousavi *et al.* [66] thoroughly compared 15 population-based optimization techniques when applied to training ANNs. According to the findings of the experiments on eight classification tasks, PSO outperforms alternative population-based metaheuristic algorithms according to the benchmark.

Recently, a metaheuristic Artificial Bee Colony (ABC) algorithm-trained artificial neural network was suggested by Ali *et al.* [67]. The experimental results demonstrate that ABC is suitable to be an optimization method for ANNs for bankruptcy prediction. The model was then tested for its ability to predict corporate bankruptcy and compared to an ANN trained using the widely used back-propagation (BPNN) and the MDA model.

BPNN: Back-Propagation Neural Network

3.5 ENSEMBLE LEARNING

In recent years, ensemble learning has been successfully used in credit scoring and financial distress prediction. Alfaro *et al.* [68] evaluated the performance of ensemble classifiers against a neural network classifier and found that utilizing AdaBoost resulted in a 30% reduction in test error as opposed to using a single neural network. Additionally, Sun *et al.* [69] considered combining several statistical and machine learning predictors, namely: SVM, MDA, decision tree, Logit, and ANN, using weighted majority voting, and it was noted that the ensemble outperformed utilizing a single classifier alone in terms of performance.

After that, Krawczyk *et al.* [70] implemented an ensemble of Decision Trees using cost-sensitive learning to address the issue of imbalanced classification, proving that the suggested method is a beneficial

method for the classification of unbalanced datasets and frequently outperforms current state-of-the-art ensembles.

Additionally, Ziba *et al.* [71] recommended using XGBoost, a novel method for bankruptcy prediction. It is employed in the training of a group of decision trees. They also provide a unique approach called synthetic features to generate higher-order statistics in data. The financial data for Polish businesses were collected between 2000 and 2012 for those still in business and between 2007 and 2013 for those who had filed for bankruptcy. The authors' methodology yielded superior results compared to the appropriate listed methodologies, including J48, random forest, SVM, and AdaBoost.

Recently, Pisula and Tomasz [72] built a scoring model based on ensemble classifiers for the early prediction of bankruptcy risk for Polish enterprises and showed the effectiveness of utilizing ensemble classifiers to forecast bankruptcy.

3.6 HYBRID METHODS

Numerous studies have been done to study the imbalance classification problem in the literature utilizing a range of tools and approaches in various combinations. Three approach combinations have been put out to forecast the financial distress of Korean enterprises by Hu *et al.* [73]. The suggested models were Self-Organizing Feature Maps (SOFM), decision trees, and MDA + ANNs. According to the authors, hybrid neural network models delivered on their promise of accuracy in forecasting a company's financial health.

After that, Min *et al.* [74] dedicated the GA to enhancing the capability of SVM when forecasting the financial situation of Korean enterprises. The primary purposes of the genetic algorithms were to enhance the SVM technique parameters and the feature subset. The novel approach outperformed stand-alone SVM and Logit in terms of performance. A modified form of SVM was also used by Tang *et al.* [75] to address the problem of class imbalance. The SVM was modified using various "rebalance" strategies, including oversampling, undersampling, and cost-sensitive methods. On the other hand, To forecast the financial collapse of dotcom enterprises, Chandra *et al.* [76] integrated several potent classifiers, including CART, SVM, Logit, MLP, and random forest.

Additionally, ensemble approaches and cost-sensitive methods were merged by Ghatasheh *et al.* [77] to forecast firms' financial situation. The effectiveness of three cost-sensitive techniques—cost-sensitive learning, cost-sensitive classification, and MetaCost—was evaluated in combination with various ensemble classifiers. This combination surpassed the other ensemble and cost-sensitive techniques analyzed in that study regarding type I error, type II error, and accuracy.

Moreover, Faris *et al.* [6] suggested a combined strategy to forecast organizations' financial state, which was based primarily on many steps, including data normalization, oversampling, feature selection, and then classification. They examined the effectiveness of utilizing basic and ensemble classifiers in their investigation. Compared to other basic and ensemble classifiers, they discovered that combining AdaBoost, SMOTE, and ensemble techniques using reduced error pruning tree showed good outcomes.

Most recently, Yotsawat *et al.* [78] employed a cost-sensitive Neural Network Ensemble for credit scoring, and the comparison findings show that the recommended technique surpassed the benchmark individual and ensemble models. Moreover, By employing tree-ensemble as a boosting approach, Zou *et al.* [79] improved the performance of the firm failure prediction. The weighted cross-entropy goal function is added to the boosted tree design to address the class imbalance problem in financial distress datasets. As a result, the weighted XGBoost may be used to forecast company failure while considering the cost.

4

METHODOLOGY

CONTENTS

This chapter presents in detail the datasets used and the two approaches (internal & external) carried out to tackle the problem of predicting company financial distress.

4.1 DATASETS

Three distinct datasets were chosen to test the suggested strategies' viability in forecasting the financial distress of enterprises. Despite the fact that the independent and independent factors changed amongst the datasets, this study nevertheless regarded this task as a classification problem. In the case of external methods (oversampling), only the Spanish companies dataset was used. On the other hand, for the internal technique, the proposed methods were evaluated individually for every dataset to accurately evaluate the suggested internal technique's significance. A summary of each dataset is provided below:

4.1.1 *Spanish companies' dataset*

This dataset for Spanish enterprises contains many financial and non-financial factors that we consider. In this case, we attempted to categorize the occurrences per class by using the label for each sample to be the dependent variable, *Bankruptcy*. Noting that the dependent variable bankruptcy is defined as three years of continuous losses [80].

The dataset was collected from the Infotel website[1]. Over the course of six years, the data was collected from 470 firms (from 1998 to 2003). There are 2860 samples in total, and 62 of them relate to bankrupt businesses. Therefore just 2% of the sample is made up of insolvent companies.

The dataset initially consisted of rows, with each row containing 37 independent variables and one dependent variable labeled Bankruptcy. This set of factors was altered in a previous attempt in [80] by eliminating unnecessary ones (those without relevance, for example, internal database business code), leading to 33 independent variables. Therefore, each record in the dataset utilized for this study comprises 33 characteristics, a mixture of non-financial and financial information. Both qualitative and quantitative (numerical) values can be assigned to a feature.

Table 4.1 presents the independent variables, together with a description of their nature, after any extraneous variables have been removed. The non-financial information that has a category value includes the firm's size, type, and provincial code (where it is located), and it also

1 Bought from http://infotel.es

includes the auditor's opinion. In contrast to the norm, where the size of the firm is usually a number, the size of the firm in this dataset is either small, medium, or big. In addition, we employed all 33 features in this study without using feature selection since, as noted by the authors in [6], incorporating a feature selection phase did not enhance the outcomes.

Table 4.1: The Spanish companies' dataset's independent variables, including financial and non-financial factors.

Financial Variables	**Description**	**Type**
Debt Structure	Long-Term Liabilities / Current Liabilities	Number
Debt Cost	Interest Cost / Total Liabilities	Number
Debt Paying Availability	Operating Cash Flow / Total Liabilities	Number
Debt Ratio	Total Assets / Total Liabilities	Number
Working Capital	Working Capital / Total Assets	Number
Warranty	Financial Warrant	Number
Operating Income Margin	Operating Income / Net Sales	Number
Return on Operating Assets	Operating Income / Average Operating Assets	Number
Return on Equity	Net Income / Average Total Equity	Number
Return on Assets	Net Income / Average Total Assets	Number
Stock Turnover	Cost of Sales / Average Inventory	Number
Asset Turnover	Net Sales / Average Total Assets	Number
Receivable Turnover	Net Sales / Average Receivables	Number
Asset Rotation	Asset allocation decisions	Number
Financial Solvency	Current Assets / Current Liabilities	Number
Acid Test	(Cash Equivalent + Marketable Securities + Net receivables) / Current Liabilities	Number
Non-financial Variables	**Description**	**Type**
Year	Corresponding to the sample	Integer
Size	Small | Medium | Large	Categorical
Number of employees		Integer
Age of the company		Integer
Type of company	Public | Limited Liability | Others	Categorical
Linked to a group	If the firm is part of a group holding	Binary
Number of partners		Integer
Province code	Location of the firm	Categorical
Number of changes of location		Integer
Delay	If the company has submitted its annual accounts on time	Binary
Historic number of judicial incidences	Since the company was established	Integer
Number of judicial incidences	Last year	Integer
Historic amount of money spent on judicial incidences	Since the firm was established	Real
Amount of money spent on judicial incidences	Last year	Real
Historic number of serious incidences	strikes, accidents...	Integer
Audited	If the firm has been audited	Binary
Auditor's opinion	Favorable | Exceptions | Unfavorable	Categorical

4.1.2 *Taiwanese Companies' Dataset*

The Taiwan Economic Journal was used to generate this dataset, which totals 6819 entries for the ten-year period (1999–2009). Of these, 6599

records (or 97% of the total) are related to non-bankrupt enterprises, while the remaining 220 records (or 3% of the total sample) are related to bankrupt firms.

This dataset comprises 95 economic features. The companies in this dataset were selected based on two specific conditions: availability of three years of financial information and comparable size to a significant group of companies. The evaluation of a company's financial health is primarily established on the restrictions of the Taiwan Stock Exchange. For more details, refer to [49].

4.1.3 *Polish companies' dataset*

This dataset contains data regarding the possibility of a Polish firm filing for bankruptcy. The data on developing markets was collected from the Emerging Markets Information Service, a worldwide database. While the still-operational businesses were evaluated from 2007 to 2013, the bankrupt companies were investigated from 2007 to 2012.

This dataset is highly imbalanced, with only 203 instances of insolvent companies, representing approximately 2% of the entire sample of around 10000 records. The dataset comprises 64 financial features, all of which are numerical. More information regarding this dataset is available in [71], which is available for download from the Kaggle ML community website. [2]

4.2 ASSESSMENT OF ADVANCED OVERSAMPLING TECHNIQUES

This section presents the study discussed in the published paper "Empirical evaluation of advanced oversampling methods for improving bankruptcy prediction [81]". We empirically evaluated and compared 11 oversampling methods for bankruptcy forecasting in this approach. The oversampling methods are ROS-I, SMOTE, SMOTE-TL, SMOTE-ENN, Borderline SMOTE, Safe-Level SMOTE, ADASYN-I, ADOMS, SPIDER, SPIDER-II, and AHC. A brief description of each oversampling method can be found in section 2.2. We only used the Spanish companies' dataset in this approach.

2 https://www.kaggle.com/competitions/companies-bankruptcy-forecast/data

Table 4.2: The confusion matrix

	Predict positive	Predict negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

4.2.1 *The classifier*

All experiments make use of the C4.5 Decision Tree as the classification algorithm. The C4.5 classifier is an extension of the ID3 algorithm. It has improvements for managing absent values, continuous attribute value distribution, and the option to select the best attribute selection metric [82]. Decision trees are generally preferred for this application because they provide comprehensible and straightforward models for decision-makers to explain.

4.2.2 *Evaluation measurements*

We used the 2X2 confusion matrix displayed in Table 4.2 to evaluate the classifier's performance along with the oversampling techniques.

Four performance measures are calculated: False Negative Rate (FNR) is expressed as a proportion and is defined in equation 4.1, False Positive Rate (FPR) is expressed as a proportion and is defined in equation 4.2, the average geometric mean is defined in equation 4.3 and accuracy defined in equation 4.4. Where TN and TP donate properly classified negative and positive cases, respectively, and positive and negative instances incorrectly classified are denoted as FP and FN, respectively [83].

$$FNR = \frac{FN}{TP + FN} \tag{4.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.2}$$

$$TPrate = \frac{TP}{FN + TP} \tag{4.3a}$$

$$TNrate = \frac{TN}{FP + TN} \tag{4.3b}$$

$$g - mean = \sqrt{TPrate \times TNrate} \tag{4.3c}$$

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.4}$$

4.2.3 *Experiments and results*

We generated separate datasets for training and testing using stratified sampling and 5-fold cross-validation, which involves dividing the training dataset into different equivalent (or nearly equivalent) segments, using one segment as the testing dataset and the remaining segments as the training dataset. In other words, at every cross-validation stage, every partition will serve as a test set. The average accuracy of each test partition is calculated as the last step in this method. Additionally, stratified sampling is employed to maintain and replicate the proportion of both classes in the train and validation segments as closely as feasible.

The process starts by normalizing the bankruptcy data and then using oversampling methods to balance the class distribution. The final step is to use the C4.5 classifier to classify the processed dataset.

We investigated several k values, ranging from 3 to 19 with an increment of 2, for the techniques that use the k closest neighbors method to provide additional minor class samples.

Without using oversampling techniques, the C4.5 classifier was evaluated as the experiment's first phase. Table 4.3 presents the findings of this evaluation. We notice that the classifier performs poorly in terms of FNR error and g-mean due to the severe imbalance in the data distribution.

Table 4.4 displays the evaluation measurements (FNR, FPR, Accuracy, and g-mean) outcomes for all oversampling methods. Only the results of the best k per method are displayed. Our major attention went to FNR since it is extra significant in a financial distress forecasting scenario, even though FPR increased when different oversampling approaches were applied [84].

Table 4.3: The prediction results obtained without applying any oversampling techniques.

Algorithm	**FNR**	**FPR**	**Accuracy**	**G-Mean**
*C*4.5	77.40%	0.46%	98.78%	0.46

Table 4.4: The evaluation results of the various oversampling techniques when used in combination with the C4.5 classification algorithm. Boldface denotes the best outcome for each metric.

Oversampling Technique	**k Value**	**FNR**	**FPR**	**G-Mean**	**Accuracy**
ROS	N/A	54.84%	2.00%	0.65	96.85%
SMOTE	5	24.19%	6.79%	0.84	92.82%
SMOTE-TL	15	17.74%	11.55%	0.85	88.32%
SMOTE-ENN	13	**12.90**%	12.37%	**0.87**	87.61%
Borderline SMOTE	19	48.39%	2.86%	0.70	96.15%
Safe Level SMOTE	13	27.41%	20.02%	0.76	79.81%
ADASYN	5	30.65%	6.76%	0.80	94.78%
ADOMS	5	38.70%	4.76%	0.75	94.51%
SPIDER	3	53.23%	2.03%	0.67	96.85%
SPIDER-II	11	48.39%	**1.82**%	0.63	**97.17**%
AHC	N/A	48.38%	2.40%	0.70	96.53%

From the results, we noticed that SMOTE-ENN was the best oversampling method to be used with the C4.5 classifier, with the lowest FNR of 12.90% and the highest g-mean score of 0.87. Additionally, it is worth noting that SMOTE-based oversampling methods (SMOTE, SMOTE-TL, and SMOTE-ENN) produced the best results regarding g-mean and FNR. The observable improvements achieved by SMOTE-based techniques can be explained by the fact that these resampling methodologies create instances of minor classes based on their neighbors.

4.3 ENSEMBLE OF COST-SENSITIVE MHOANNS

This section presents the study discussed in the published paper "Cost-Sensitive Metaheuristic Optimization-Based Neural Network with Ensemble Learning for Financial Distress Prediction [85]". Which discussed the novel technique for insolvent firms prediction using a homogeneous majority-voting ensemble learning, in which the base learners are MHOANNs, optimized using PSO and CSO algorithms while applying a cost-sensitive fitness function (ENS_PSONN$_{cost}$ and ENS_CSONN$_{cost}$).

Figure 4.1 shows the architecture of a single base learner. In this architecture, the weights and biases of the neural network are generated using one of the metaheuristic optimizers (PSO or CSO). And then, the weights and biases will be used to construct the neural network that is used to generate the predictions. The cost-sensitive fitness function is used once the predictions have been computed and the best solution is stored. The maximum number of iterations will be reached by repeating these procedures. The optimal set of weights and biases will be utilized to create a neural network to categorize the samples in the testing dataset. Then, all metrics are computed and identified.

The whole framework design of an MHOANN in a majority-voting ensemble learning system is demonstrated in Figure 4.2. Here, sampling with replacement is used to generate n unique training datasets from the original training dataset. One of the resultant training datasets will be used to train every MHOANN. The output predictions are then supplied to the majority-voting component to compute the final predictions. Next, every trained MHOANN is utilized to create predictions on the same test dataset. The component diagram for the recommended approach is shown in Figure 4.3.

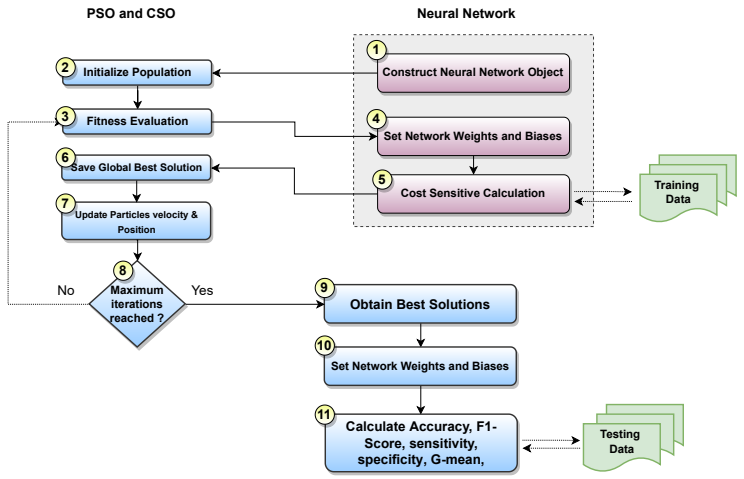

Figure 4.1: Diagram of the neural network architecture based on meta-heuristic optimization featuring a cost-sensitive fitness function.

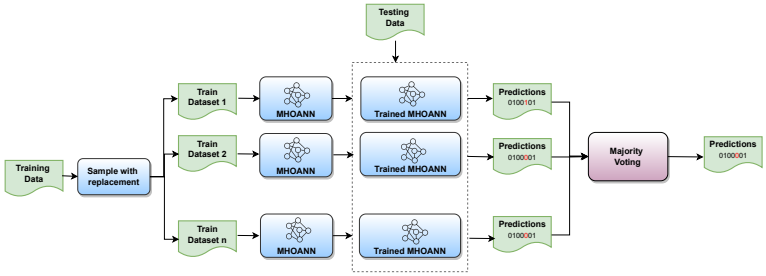

Figure 4.2: Diagram of a Homogeneous Ensemble Learning framework with majority voting where the base learners are MHOANN featuring a cost-sensitive fitness function.

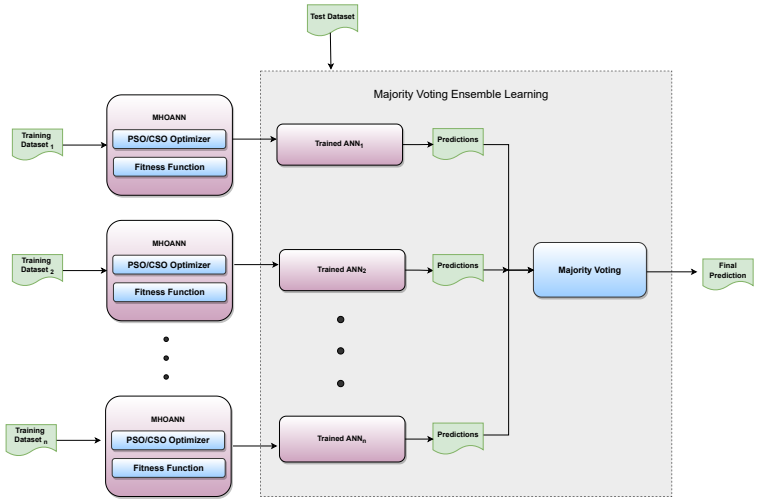

Figure 4.3: Component diagram of ENS_PSONN$_{cost}$ and ENS_CSONN$_{cost}$. This diagram illustrates the main components of our framework, including MHOANNs with PSO or CSO as the optimizers for the neural networks and a custom, cost-sensitive fitness function. The outputs of these MHOANNs are then combined using the majority voting method to produce the final prediction.

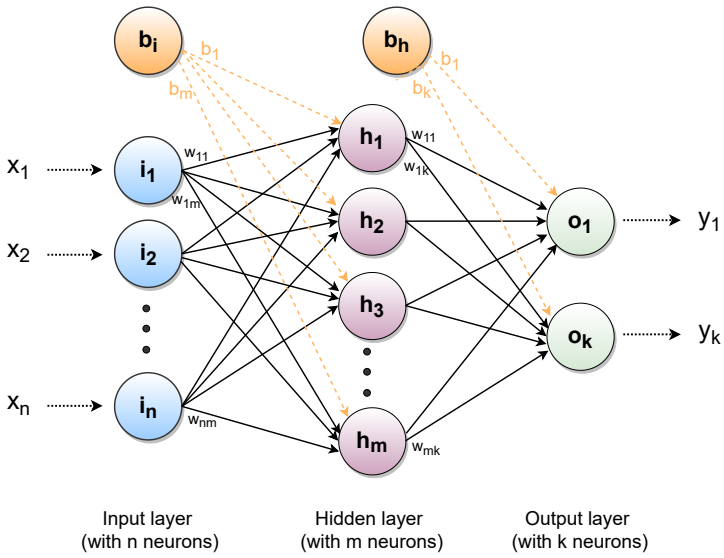

Figure 4.4: Typical design of the artificial neural network consists of its architectural elements..

4.3.1 *The framework*

In this section, we will discuss the suggested framework and explain how to apply the optimizers to compute the weights and biases of the neural networks. Describe the implementation of the different fitness functions and the suggested evaluation metrics. Also, how to apply all this in ensemble learning using majority voting.

4.3.1.1 *ANN Classifier*

ANNs [86–89] are among the most popular methods for solving classification problems, and they are algorithms modeled after the human brain designed to mimic how people learn. Due to the nonlinear structure of ANNs and their uncertain optimal set weights and biases, they have an extremely challenging learning process. An ANN's learning process significantly impacts how effective it is. Figure 4.4 displays a structural blueprint of a typical ANN.

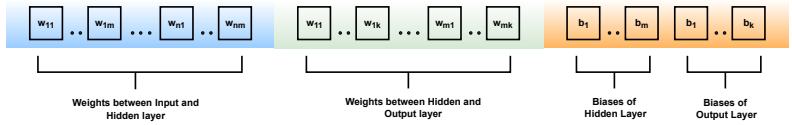

Figure 4.5: Solution illustration of particles by a vector.

4.3.1.2 *The optimizer*

In order to surmount the drawbacks of traditional training techniques, optimization algorithms were used to compute the weights and biases in the ANN. State-of-the-art PSO and CSO metaheuristic algorithms were used in this approach as the ANN optimization strategies.

In this approach, we have the input layer, one hidden layer, and the output layer. Let n be the input features count, m be the hidden neurons count, and k be the output neurons count. We have $(w_{11} - w_{nm})$ as the set of weights between the input layer and the hidden layer, $(w_{11} - w_{mk})$ as the set of weights between the hidden layer and the output layer, $(\beta_1 - \beta_m)$ as the set of biases of the hidden layer, and $(\beta_1 - \beta_k)$ is the set of biases of the output layer. Each particle in the swarm is represented by a vector. Figure 4.5, and each vector represents a solution. The length of the vector (l_s) is determined by equation 4.5. Since we have a binary classification task in this work, the output layer contains a single neuron, and the equation can be simplified by setting the value of k to 1, as shown in equation 4.6.

$$l_s = (n * m) + (m * k) + m + k \tag{4.5}$$

$$l_{s\_binary} = (n * m) + (2 * m) + 1 \tag{4.6}$$

4.3.1.3 *Fitness Functions*

According to the chosen fitness function in evolutionary computing, the population develops in an attempt to improve its fitness function score [90]. In this approach, the suggested cost-sensitive fitness function was tested against the Mean Square Error (MSE) and accuracy fitness functions. Below is a description of each fitness function:

- MSE: is a widely used fitness function in MHOANNs, and Evolutionary Neural Networks (ENNs) [91, 92]. It calculates the average difference between the predicted and actual values, as shown in equation 4.7. In the equation, i ranges from 1 to n, where n is the number of samples, y_i represents the actual value, and $\hat{y}_i$ is the predicted value.

$$cost_{MSE} = \frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.7}$$

- Accuracy: is the proportion of accurately anticipated cases to all instances. In this case, the cost is calculated as 1 minus the accuracy, as seen in equation 4.8. The equation uses four terms, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), to calculate the accuracy.

$$cost_{accuracy} = 1 - (\frac{TP + TN}{TP + TN + FP + FN}) \tag{4.8}$$

- Cost-Sensitive: A cost matrix is used to take into account the expenses of misclassifying. It is an n-by-n matrix, with n being the labels count, and it resembles a confusion matrix. Each element in this matrix corresponds to the cost of the misclassified element in the confusion matrix.

  To calculate the cost for a cost-sensitive fitness function, we define C as the cost matrix and A as the confusion matrix. The matching weight in the cost matrix is multiplied by each member in the confusion matrix to get the updated confusion matrix A'. The accuracy is then calculated using this updated confusion matrix. To obtain the final cost, we subtract this accuracy from 1. The steps for this calculation are shown in equation 4.9.

$$A = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \tag{4.9a}$$

$$C = \begin{bmatrix} W_{TP} & W_{FP} \\ W_{FN} & W_{TN} \end{bmatrix} \tag{4.9b}$$

$$A' = \begin{bmatrix} W_{TP} \times TP & W_{FP} \times FP \\ W_{FN} \times FN & W_{TN} \times TN \end{bmatrix} = \begin{bmatrix} TP' & FP' \\ FN' & TN' \end{bmatrix} \tag{4.9c}$$

$$CostSensitiveAccuracy = \frac{TP' + TN'}{TP' + TN' + FP' + FN'} \tag{4.9d}$$

$$cost_{cost\_sensitive} = 1 - CostSensitiveAccuracy \tag{4.9e}$$

4.3.1.4 *Majority Voting Ensemble Learning*

As described in Sect. 2.5 ensemble learning is a technique that uses multiple models, called inducers or basic learners, to make predictions. Each inducer is a machine learning algorithm trained on labeled examples to create a model that can generate predictions for new, unlabeled samples. The predictions of the multiple inducers are then combined to create a final prediction. The idea behind ensemble learning is that by using various models, the weaknesses of an individual inducer may be compensated for by other inducers, resulting in a more robust overall model [38].

In this technique, we use sampling with replacement to create five different datasets and then train a homogeneous ensemble learning model on the resulting datasets. Where MHOANNs with a cost-sensitive fitness function are the ensemble members. The plurality voting technique is then used to get the final predictions for the test dataset.

4.3.2 *Evaluation measurements*

An imbalanced dataset presents a clear problem for binary classification since the trained model favors the instance from the major class, producing high accuracy but an unsuccessful prediction of minority class samples. Hence, we will consider the following evaluation metrics in this approach to focus on both majority class and minority class prediction effectiveness: The accuracy is computed using the confusion matrix shown in equation 4.10. G-mean is the sensitivity and specificity's geometric mean, as determined by equation 4.13. The f1-score is the harmonic mean of the precision and sensitivity specified in equation 4.14, where *beta* is a real positive factor selected to make sensitivity *beta* times more significant than precision. In this study, we employed the formula $beta = 1$, which equally weights sensitivity and precision.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.10}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.11}$$

$$Specificity = \frac{TN}{FP + TN} \tag{4.12}$$

$$g - mean = \sqrt{Sensitivity \times Specificity} \tag{4.13}$$

$$f1 - score = \frac{(1 + \beta^2).sensitivity.precision}{sensitivity + \beta.precision} \tag{4.14}$$

where

$$\beta \geq 0$$

4.3.3 *Experiments and Results*

The experimental setup, benchmarks, and procedures are presented in this part, along with the experiment's outcomes and an analysis of them.

4.3.3.1 *Environment and Experiments Setup*

An eight-core, 2.3 GHz and 16 GB of RAM laptop was used to conduct the trials. We implemented the ANN powered by PSO and CSO as optimization methods with a cost-sensitive fitness function using Evolopy-NN [93]. Evolopy-NN, a Python 3.7-based open-source optimization framework inspired by nature, uses evolutionary and metaheuristic methods to train neural networks. All datasets were divided into a 34 percent testing dataset and a 66 percent training dataset, respectively [94, 95].

To keep the proportion of minor to major classes in the generated datasets, we employed stratified sampling. Therefore, following sampling, the minor class makes up 2% of both the training and test datasets for the Spanish enterprises. Similarly, the minor class makes up 3% of Taiwanese enterprises' training and test datasets. Additionally, the minor class makes up 2% of both the training and test datasets for Polish enterprises.

Each experiment was conducted five times for a total of 100 iterations, with the population size set to 50. Five weak learners were utilized in the ensemble learning process, and the final prediction was determined by a majority vote.

We suggest utilizing two optimization methods, PSO and CSO, as well as three fitness functions, MSE, accuracy, and cost-sensitive. We created six different versions of the MHOANN for this experiment, as follows:

1. MSE fitness function in a PSO-optimized ANN.
2. Accuracy fitness function in a PSO-optimized ANN.
3. Cost-sensitive fitness function in a PSO-optimized ANN.
4. MSE fitness function in a CSO-optimized ANN.
5. Accuracy fitness function in a CSO-optimized ANN.
6. Cost-sensitive fitness function in a CSO-optimized ANN.

4.3.3.2 *Effect of fitness function*

By constructing a cost-sensitive fitness function based on the confusion matrix as stated in Sub-Sect. 4.3.1.3, we enhanced the MHOANN to add a cost for a misclassified instance during model training. We are attempting to avoid FN for the issue we are addressing, which is when the model forecasts a firm will be financially sound even if it is in a financial crisis. As a result, we give the FN a weight cost. The dataset and technique employed determine the appropriate weight for the FN. To do this, we experimented with various weights while keeping an eye on the metrics to find the most effective weight to utilize. We were able to experiment with the entire datasets in this study since the datasets were relatively modest in size. However, in practical situations, when the dataset is larger, we advise utilizing a sample of the dataset to determine the appropriate weight to employ in order to minimize computing expenses.

For the following experiments, we took into account the weight that produced the greatest g-mean score. See Table 4.5 for the Spanish companies' dataset with the PSO optimizer, Table 4.6 for the Spanish companies' dataset with the CSO optimizer, Table 4.7 for the Taiwanese companies' dataset with the PSO optimizer, and Table 4.8 for the

Table 4.5: Impact of False Negative Weight on evaluation measures applying PSO-optimized neural network on the Spanish Companies' Dataset. Boldface denotes the best outcome for each metric.

FN Weight	Accuracy	Sensitivity	Specificity	F1-Score	G-Mean
1	**0.98**	0.05	**1.00**	0.09	0.22
25	0.91	0.48	0.92	0.19	0.66
50	0.82	0.81	0.82	0.16	0.81
75	0.77	0.81	0.77	0.13	0.79
100	0.75	**0.95**	0.75	0.14	**0.84**
125	0.81	0.81	0.81	**0.15**	0.81
150	0.71	0.81	0.71	0.11	0.76
175	0.72	0.86	0.72	0.12	0.79
200	0.72	0.86	0.72	0.12	0.79

Taiwanese companies' dataset with the CSO optimizer, Table 4.9 for the Polish companies' dataset with the PSO optimizer, and Table 4.10 for the Polish companies' dataset with the CSO optimizer.

These experiments led us to conclude that FN's ideal weight is 100 when applying PSO while using the Spanish companies' dataset, as illustrated in Figure 4.6. Additionally, 75 when the same dataset was analyzed using CSO, as shown in Figure 4.7. On the other hand, as illustrated in Figure 4.8, we found that the ideal weight of FN when applying PSO on the dataset of Taiwanese enterprises is 50. Moreover, the results are the same when CSO is applied to the same dataset, as in Figure 4.9. Additionally, as demonstrated in Figure 4.10, the optimal weight of FN while using PSO on the dataset of Polish enterprises is 175. Similar results were obtained when CSO was applied to the same dataset, as shown in Figure 4.11.

The MHOANN was trained using a cost-sensitive fitness function after applying the appropriate FN weight and applied to classify the testing sample. This process involved establishing the ideal FN weight for a specific optimization technique for each dataset. The severe data imbalance in the datasets under consideration may be used to explain why the cost of FN is so expensive, ranging from 50 up to 175.

We contrasted our findings with references to determine the impact of the cost-sensitive fitness function. To track the evaluation metrics without employing cost-sensitive learning, we trained the ANN on both datasets using each optimizer (PSO and CSO) and two different

Table 4.6: Impact of False Negative Weight on evaluation measures applying CSO-optimized neural network on the Spanish Companies' Dataset. Boldface denotes the best outcome for each metric.

FN Weight	Accuracy	Sensitivity	Specificity	F1-Score	G-Mean
1	**0.98**	0.05	**0.99**	0.06	0.24
25	0.91	0.61	0.92	**0.23**	0.75
50	0.86	0.72	0.86	0.18	0.79
75	0.77	0.82	0.77	0.13	**0.79**
100	0.73	0.78	0.73	0.12	0.76
125	0.69	**0.86**	0.68	0.11	0.77
150	0.73	0.80	0.72	0.11	0.76
175	0.68	0.85	0.68	0.10	0.76
200	0.67	**0.86**	0.66	0.10	0.75

Table 4.7: Impact of False Negative Weight on evaluation measures applying PSO-optimized neural network on the Taiwanese Companies' Dataset. Boldface denotes the best outcome for each metric.

FN Weight	Accuracy	Sensitivity	Specificity	F1-Score	G-Mean
1	**0.97**	0.06	**1.00**	0.11	0.24
25	0.88	0.78	0.88	**0.30**	0.83
50	0.82	**0.85**	0.82	0.24	**0.83**
75	0.83	0.83	0.83	0.24	0.83
100	0.77	0.88	0.76	0.20	0.82
125	0.75	0.91	0.74	0.19	0.82
150	0.76	0.87	0.76	0.19	0.81
175	0.77	0.83	0.77	0.21	0.79
200	0.77	0.83	0.77	0.21	0.79

Table 4.8: Impact of False Negative Weight on evaluation measures applying CSO-optimized neural network on the Taiwanese Companies' Dataset. Boldface denotes the best outcome for each metric.

FN Weight	Accuracy	Sensitivity	Specificity	F1-Score	G-Mean
1	**0.97**	0.05	**1.00**	0.09	0.21
25	0.86	0.77	0.86	**0.27**	0.82
50	0.81	0.89	0.81	0.24	**0.85**
75	0.78	0.87	0.77	0.20	0.82
100	0.76	0.88	0.76	0.20	0.82
125	0.76	0.88	0.75	0.20	0.81
150	0.63	**0.94**	0.62	0.14	0.77
175	0.72	0.90	0.71	0.17	0.80
200	0.70	0.91	0.70	0.17	0.79

Table 4.9: Impact of False Negative Weight on evaluation measures applying PSO-optimized neural network on the Polish Companies' Dataset. Boldface denotes the best outcome for each metric.

FN Weight	Accuracy	Sensitivity	Specificity	F1-Score	G-Mean
1	**0.97**	0.01	**0.99**	0.02	0.12
25	0.89	0.38	0.90	0.12	0.58
50	0.82	0.46	0.83	0.10	0.62
75	0.74	0.52	0.75	0.08	0.62
100	0.76	0.65	0.76	0.10	0.70
125	0.73	0.71	0.73	0.10	0.72
150	0.74	0.87	0.74	0.11	0.78
175	0.79	**0.90**	0.79	**0.15**	**0.84**
200	0.71	0.83	0.70	0.10	0.76
225	0.65	0.75	0.65	0.08	0.70

Table 4.10: Impact of False Negative Weight on evaluation measures applying CSO-optimized neural network on the Polish Companies' Dataset. Boldface denotes the best outcome for each metric.

FN Weight	Accuracy	Sensitivity	Specificity	F1-Score	G-Mean
1	**0.97**	0.01	**0.97**	0.02	0.12
25	0.71	0.46	0.71	0.06	0.57
50	0.70	0.48	0.71	0.06	0.58
75	0.64	0.62	0.64	0.07	0.63
100	0.61	0.77	0.61	0.07	0.68
125	0.62	0.81	0.62	0.08	0.71
150	0.73	0.84	0.72	0.11	0.78
175	0.79	**0.91**	0.79	**0.15**	**0.85**
200	0.71	0.84	0.70	0.10	0.77
225	0.65	0.77	0.65	0.08	0.71

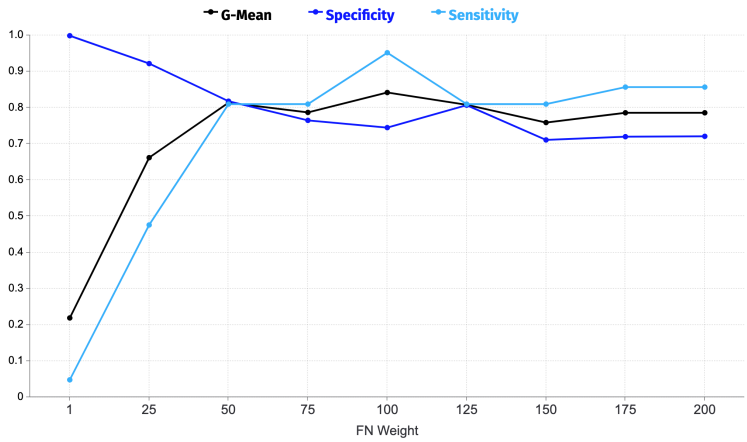

Figure 4.6: Sensitivity, Specificity, and G-Mean scores from employing Cost-Sensitive-PSO-Optimized-ANN on the Spanish firms' data for different False Negative weights.

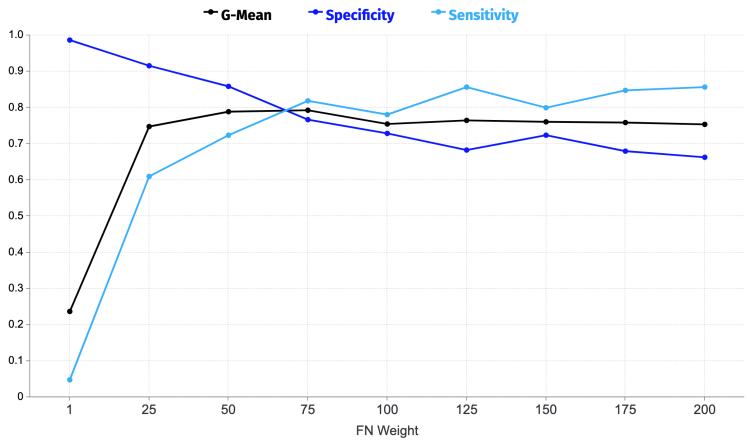

Figure 4.7: Sensitivity, Specificity, and G-Mean scores from employing Cost-Sensitive-CSO-Optimized-ANN on the Spanish firms' data for different False Negative weights.

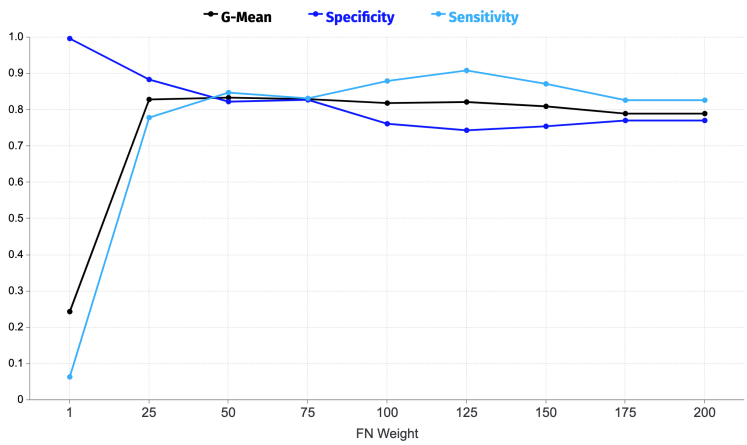

Figure 4.8: Sensitivity, Specificity, and G-Mean scores from employing Cost-Sensitive-PSO-Optimized-ANN on the Taiwanese firms' data for different False Negative weights.

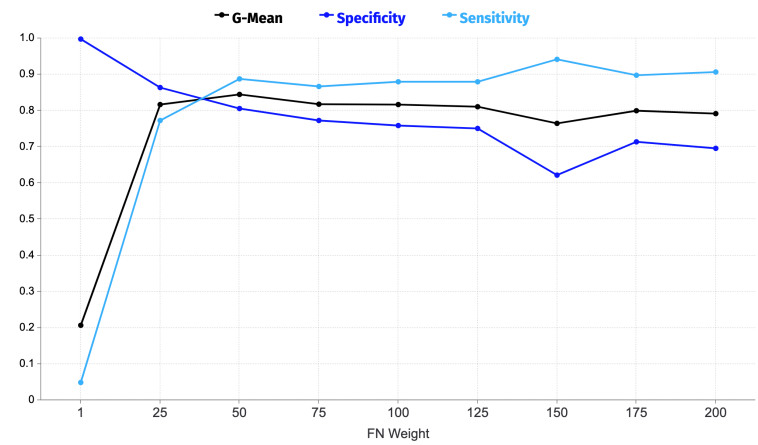

Figure 4.9: Sensitivity, Specificity, and G-Mean scores from employing Cost-Sensitive-CSO-Optimized-ANN on the Taiwanese firms' data for different False Negative weights.

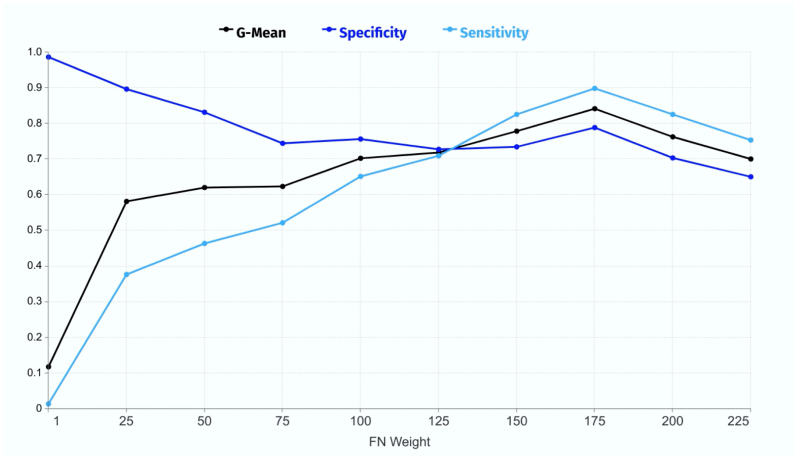

Figure 4.10: Sensitivity, Specificity, and G-Mean scores from employing Cost-Sensitive-PSO-Optimized-ANN on the Polish firms' data for different False Negative weights.

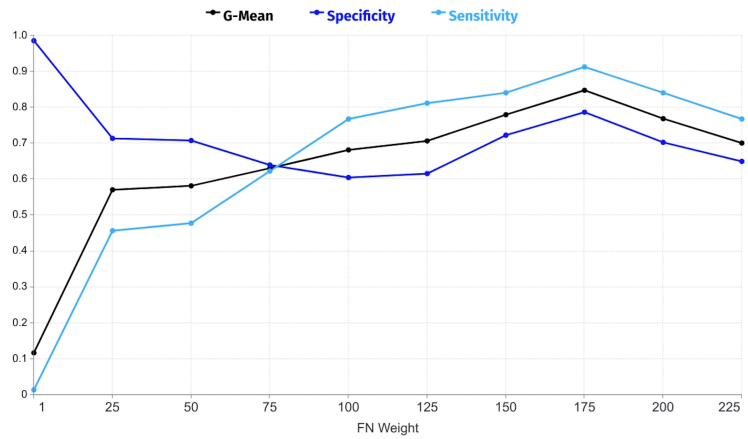

Figure 4.11: Sensitivity, Specificity, and G-Mean scores from employing Cost-Sensitive-CSO-Optimized-ANN on the Polish firms' data for different False Negative weights.

fitness functions, MSE, and accuracy. We ran four trials for each dataset, employing an ANN with a PSO optimizer and MSE as the fitness function in the first, a PSO optimizer and accuracy in the second, a CSO optimizer and MSE in the third, and a CSO optimizer and accuracy in the fourth. Each metric's mean, standard deviation, and maximum score were recorded.

The outcomes of every fitness function applied using the Spanish companies' dataset are displayed in Table 4.11. The results of every fitness function when the dataset of Taiwanese firms was used are presented in Table 4.12. Additionally, the results of all fitness functions when the dataset of Polish firms was used are shown in Table 4.13. There was a negative impact on the accuracy. However, the cost-sensitive MHOANN showed a significant improvement in predicting the minority class samples, significantly influencing the g-mean and a reasonable enhancement on the f1-score.

Comparing ANN with PSO as a classifier with a cost-sensitive-fitness-function to the identical classifier but with MSE-fitness-function using the dataset of Spanish companies, we observed a significant improvement in g-mean: MSE-fitness-function g-mean was 0.211 while the cost-sensitive fitness function g-mean was 0.842, also, an increase in the f1-score: MSE-fitness-function f1-score was 0.104 while cost-sensitive-

Table 4.11: The evaluation metrics outcomes of all fitness functions per optimization algorithm used on the dataset of Spanish companies. Boldface denotes the best average outcome for each metric.

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-Sensitive	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-Sensitive	CSO	0.768	0.771	0.001	**0.793**	0.801	0.001	**0.134**	0.150	0.001

Table 4.12: The evaluation metrics outcomes of all fitness functions per optimization algorithm used on the dataset of the Taiwanese Companies. Boldface denotes the best average outcome for each metric.

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-Sensitive	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-Sensitive	CSO	0.808	0.810	0.002	**0.845**	0.846	0.001	**0.237**	0.239	0.002

Table 4.13: The evaluation metrics outcomes of all fitness functions per optimization algorithm used on the dataset of Polish Companies. Boldface denotes the best outcome for each metric.

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-Sensitive	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-Sensitive	CSO	0.790	0.794	0.003	**0.848**	0.850	0.001	**0.150**	0.152	0.002

fitness-function f1-score was 0.141, and a decrease in the accuracy: MSE-fitness-function accuracy was 0.978 while cost-sensitive-fitness-function accuracy was 0.749, and, similar results were seen when comparing ANN with PSO as a classifier using a cost-sensitive-fitness function to the exact classifier with the accuracy-fitness-function. G-mean significantly increased: accuracy-fitness-function g-mean was 0.131 while cost-sensitive-fitness-function g-mean was 0.842, also f1-score improved: accuracy-fitness-function g-mean was 0.054 while cost-sensitive-fitness-function g-mean was 0.141, and the accuracy decreased: cost-sensitive-fitness-function accuracy was 0.979 while accuracy-fitness-function accuracy was 0.749. Likewise, when comparing ANN with CSO as a classifier with a cost-sensitive-fitness-function to the exact classifier but with MSE-fitness-function, the g-mean increased significantly: MSE-fitness-function g-mean was 0.211 while the cost-sensitive-fitness-function g-mean was 0.793, f1-score increased: MSE-fitness-function f1-score was 0.104, while cost-sensitive-fitness-function f1-score was 0.134, and the accuracy decreased: MSE-fitness-function accuracy was 0.980, while cost-sensitive-fitness-function accuracy was 0.768. Additionally, when comparing ANN with CSO as a classifier while using the cost-sensitive-fitness-function to the exact classifier but using the accuracy-fitness-function, we observed a significant improvement in g-mean: accuracy-fitness-function g-mean was 0.062 while the cost-sensitive-fitness-function was 0.793. Also, an increase in the f1-score: accuracy-fitness-function f1-score was 0.032, while the cost-sensitive-fitness-function f1-score was 0.134. However, we noticed a decrease in the accuracy: the accuracy-fitness-function accuracy score was 0.980, while the cost-sensitive-fitness-function accuracy was 0.768.

Using the dataset of Taiwanese enterprises, we also found comparable findings. In other words, when comparing ANN with PSO as a classifier using the cost-sensitive fitness function to the exact classifier using the MSE fitness function, we observed a significant improvement in the g-mean: MSE-fitness-function g-mean score was 0.332 while the cost-sensitive-fitness-function g-mean was 0.834, an increase in the f1-score: MSE-fitness-function f1-score was 0.186 while cost-sensitive-fitness-function f1-score was 0.242. However, accuracy dropped from 0.968 to 0.824. Additionally, when comparing ANN with PSO as a classifier while using cost-sensitive fitness function to the exact classifier while using accuracy as a fitness function, we noticed a significant improvement in g-mean: accuracy-fitness-function g-mean was 0.244 while cost-sensitive-fitness-function g-mean was 0.834, an improvement in the f1-score: accuracy-fitness-function f1-score was 0.110 while cost-sensitive-fitness-function f1-score was 0.242, but a decrease in the accuracy score: accuracy-fitness-function accuracy was 0.967 while cost-sensitive-fitness-function accuracy was 0.824. Similarly, when ANN using CSO as a classifier was compared to the exact classifier with MSE fitness function, we observed an improvement in the g-mean: MSE-fitness-function g-mean was 0.290 while cost-sensitive-fitness-function g-mean was 0.845; the improvement in the f1-score: MSE-fitness-function f1-score was 0.147 while cost-sensitive-fitness-function g-mean was 0.237, but a decrease in accuracy: MSE-fitness-function was 0.967 while cost-sensitive-fitness-function was 0.808. Similarly, when ANN with CSO was used as a classifier with a cost-sensitive fitness function compared to the same classifier with accuracy as the fitness function, the g-mean increased: accuracy-fitness-function g-mean was 0.207, while the cost-sensitive-fitness-function was 0.845. the f1-score increased: accuracy-fitness-function f1-score was 0.087 while cost-sensitive-fitness-function was 0.237, but the accuracy decreased: accuracy-fitness-function accuracy was 0.968 while cost-sensitive-fitness-function was 0.808.

Additionally, we found the same thing while utilizing the dataset of Polish enterprises. In other words, when comparing ANN with PSO as a classifier while using the cost-sensitive fitness function with the exact classifier using the MSE fitness function. We observed a significant improvement in the g-mean: MSE-fitness-function g-mean was 0.118 while cost-sensitive-fitness-function was 0.842, an increase in the f1-score: MSE-fitness-function f1-score was 0.019 while cost-sensitive-fitness-function g-mean was 0.149, but decrease in the accuracy: MSE-fitness-function was 0.970 while cost-sensitive-fitness-function was 0.790. Similarly, when comparing ANN with PSO as a classifier while using the cost-sensitive fitness function to the exact classifier using accuracy as a fitness function, we observed an improvement in the g-mean: accuracy-fitness-function g-mean was 0.118 while cost-sensitive-fitness-

function g-mean was 0.842, an improvement in the f1-score: accuracy-fitness-function f1-score was 0.018 while cost-sensitive-fitness-function f1-score was 0.149, but a drop in accuracy: accuracy-fitness-function accuracy was 0.967 while cost-sensitive-fitness-function accuracy was 0.790. Moreover, in the comparison between ANN using CSO as a classifier and the exact classifier using MSE fitness function, we noticed an increase in g-mean: MSE-fitness-function g-mean was 0.118. In contrast, cost-sensitive-fitness-function g-mean was 0.848, also an improvement in f1-score: MSE-fitness-function f1-score was 0.020 while cost-sensitive-fitness-function f1-score was 0.150, but a drop in accuracy: MSE-fitness-function accuracy was 0.970 while cost-sensitive-fitness-function accuracy was 0.790. Similarly, when ANN with CSO was used as a classifier with a cost-sensitive fitness function compared to the exact classifier with an accuracy fitness function, we observed an increase in g-mean: accuracy-fitness-function g-mean was 0.117, while cost-sensitive-fitness-function g-mean was 0.848. Also, the f1-score increased: the accuracy-fitness-function f1-score was 0.017 while cost-sensitive-fitness-function f1-score was 0.150. On the other hand, accuracy dropped: accuracy-fitness-function accuracy was 0.967 while cost-sensitive-fitness-function accuracy was 0.790.

Adding the cost to the FN instances increases the count of TP cases, which explains the improvement in f1-score and g-mean metrics. Nevertheless, it leads to higher FP occurrences, explaining why the accuracy score dropped. Then, while retaining the number of TP instances, we utilized majority-voting ensemble learning to reduce the count of FP cases.

Interestingly, a simple optimizer like CSO, which uses a straightforward method to update particles in the search space, can perform similarly to more complex optimizers when used with an MHOANN.

Although these tests demonstrated that PSO and CSO produced comparable outcomes when employing equivalent fitness functions, CSO was quicker; in particular, CSO performed 22.4% quicker using the dataset of Spanish companies, 34.4% faster using the dataset of Taiwanese companies, and 48.3% faster using the dataset of Polish enterprises, this was done with a population size of 50 and 100 iterations, Table 4.14 shows the actual execution times in seconds.

We observed a strong correlation between FN's weight and the measures evaluated in this study, as mentioned in Sub-Sect. 4.3.3.2. However, the choice of weight for this method can impact the specific metric the user chooses to focus on. A lower weight may result in a higher specificity score, while a higher weight may result in a higher sensitiv-

Table 4.14: Execution times for the PSO and CSO algorithms.

Optimizer	Dataset	Execution Time (s)		
		Avg.	Best	Std.
PSO	Spanish	197	189	5.7
CSO	Spanish	153	151	2.3
PSO	Taiwanese	1260	1213	58.7
CSO	Taiwanese	827	798	19.7
PSO	Polish	1778	1732	50.3
CSO	Polish	919	820	59.4

ity score. In this case, we opted to choose the weight that yields the best g-mean score, which is a balance between sensitivity and specificity.

4.3.3.3 *Effect of ensemble learning framework*

While the MHOANN with cost-sensitive fitness function greatly decreased the frequency of FN occurrences and significantly improved minor class instances prediction, there was an increase in FP. It is crucial to maintain high accuracy in the classification model, even though in these particular datasets, the minority class is more important and valuable than the majority class; to put it differently, incorrectly identifying a financially distressed company as solvent incurs a much greater cost than mistakenly identifying a financially stable company as being in distress [96].

The fundamental idea behind ensemble learning is that by combining many models, the shortcomings of one model will almost certainly be offset by those of other models, as explained in Sect. 4.3. Therefore, we created five training sets per dataset using sampling with replacement, trained the cost-sensitive MHOANN on each new dataset, produced the forecast on the test dataset already existing, and then utilized majority voting to get the final predictions.

The results of the cost-sensitive MHOANN on the dataset of Spanish enterprises are compared with the results of employing a cost-sensitive MHOANN in a majority voting ensemble learning system on the same dataset in Table 4.15. A comparison of the cost-sensitive MHOANN and cost-sensitive MHOANN in a majority voting ensemble learning

Table 4.15: Evaluation measurements comparison on the Spanish companies' dataset.

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
Rate of 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
Rate of change		15.0%	10.5%	15.0%	87.3%	12.6%

Table 4.16: Evaluation measurements comparison on the Taiwanese companies' dataset.

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
Rate of 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
Rate of change		8.4%	3.6%	8.4%	36.7%	6.2%

system on the dataset of Taiwanese enterprises is shown in Table 4.16. The same comparison is also shown for the dataset of Polish enterprises in Table 4.17. Reviewing the findings, we found that most assessment measures had improved; in particular, the accuracy score had increased, ranging from 8.4% to 15.0%, the g-mean had increased, ranging from 4.2% to 12.6%, and the f1-score had significantly improved, ranging from 36.7% to 87.3%.

The primary goal of ensemble learning is to develop strong prediction abilities that, at the very least, go beyond those of the individual techniques that make up the ensemble. To do this, the ensemble's weak learners must be accurate and diversified, as noted by [12]. The improvement in all measures demonstrates that PSO-optimized and CSO-optimized neural networks are diverse and accurate. Hence they are suitable to be used in a homogenous ensemble learning system.

Table 4.17: Evaluation measurements comparison on the Polish companies' dataset.

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
Rate of 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
Rate of change		12.5%	1.6%	12.7%	79.3%	7.0%

4.3.4 *Analysis and discussion*

The imbalanced nature of many real-world datasets can pose a challenge for machine learning algorithms, as they may often be biased towards the majority class, resulting in poor performance on the minority class. This can have significant implications in applications such as financial distress prediction, where the minority class is the most important to identify. In light of this, the proposed method in our experiments demonstrated a solid ability to address the issue of class imbalance. When using a cost-sensitive approach, the algorithm gives more weight to the minority class, effectively shifting the bias away from the majority class. This is reflected in the improved g-mean score, which considers the classification performance for both the majority and minority classes. Additionally, ensemble learning helped decrease the adverse side effects that may result from the bias shift while maintaining a high accuracy score. This suggests that the proposed method is a promising solution for addressing imbalanced datasets and has the potential to significantly improve the performance of machine learning algorithms in a variety of real-world applications.

Our hypothesis stated that applying a cost on misclassified positive class instances would increase the number of TP predictions and decrease the number of FN predictions. This was supported by the results of the experiments, which showed that the sensitivity score, or the ability to identify the minority class correctly, improved significantly. However, as expected, this came at the cost of an increase in the number of FP predictions and a decrease in the number of TN predictions. Despite this trade-off, the overall g-mean score, which measures the balance between sensitivity and specificity, improved in all experi-

ments. This can be attributed to the highly imbalanced nature of the dataset, where the number of instances belonging to the minority class $(TP + FN)$ is much smaller than the number of instances belonging to the majority class $(FP + TN)$. As a result, the improvement in sensitivity had a more significant impact on the g-mean score, outweighing the decrease in specificity. Overall, these results support the effectiveness of the proposed method for handling imbalanced datasets.

In addition to the positive impact of the cost-sensitive approach, the application of ensemble learning also resulted in an overall improvement in all evaluation measurements used. This demonstrates the diversity of the proposed method and, more specifically, the use of an MHOANN and its potential for use in a homogeneous ensemble learning system. Ensemble learning combines the predictions of multiple models, resulting in a stronger learner that can make more accurate predictions. In this case, ensemble learning resulted in a slight improvement in the g-mean score, as the number of FN was approximately maintained. In contrast, the number of FP decreased. This significantly improved the accuracy score, as the overall number of incorrect predictions was reduced. Overall, these results suggest that ensemble learning, in combination with the proposed cost-sensitive approach, is an effective method for addressing imbalanced datasets and improving the performance of machine learning algorithms.

Concerning execution performance, CSO demonstrated faster execution time compared to PSO. This can be attributed to the fact that only half of the population is updated in CSO, while the entire population is updated in PSO. As a result, CSO can complete the optimization process more efficiently and in less time. Overall, the results of the experiments indicate that CSO is a promising optimization method for addressing ANN. Its faster execution time and ability to find high-quality solutions make it a valuable tool for many real-world applications. Moreover, an interesting observation from the experiments is that a light optimizer with a simple mechanism, such as CSO, can achieve similar results to a more complex optimizer like PSO when used as an optimizer for MHOANN. This suggests that the simplicity and efficiency of CSO make it a viable alternative to more complex optimizers, especially when combined with the proposed method for addressing imbalanced datasets.

In Appendix B, we presented the convergence (learning) curve graphs for a sample run of both PSO and CSO on each dataset, using each of the fitness functions evaluated. These graphs provide a visual representation of the learning process, showing how the performance of the algorithms improves over time. Upon analyzing the charts, we observed

that the MSE and accuracy fitness functions had the lowest fitness values, demonstrating the model's remarkable accuracy, supported by the already-mentioned data. As was already noted, it is skewed in favor of the dominant class and inaccurately predicts the minority class. On the other hand, the cost-sensitive fitness function results in a greater fitness value, which is expected given that this function multiplies the number of FN by the cost. Additionally, in every experiment, the fitness score stabilizes at around 100 iterations, suggesting that additional training will not significantly impact the model's performance.

4.3.5 *Comparison to other approaches*

In previous research [6] using the same Spanish companies' dataset, the authors suggested a hybrid technique that combined ensemble strategies with oversampling. The authors employed five feature selection procedures to select the most prevalent features of insolvency forecast. Foremost, the authors likened four oversampling techniques and then used the C4.5 decision tree classifier to decide on the most useful oversampling technique. SMOTE delivered the best outcomes. After that, the authors experimented with various primary and ensemble well-known classification methods as the reference for the research. In Table 4.18, we presented the g-mean score of the standard classifiers from [6] compared with the g-mean scores of the two methods suggested in this thesis. Notice that the structure of the ensemble technique names is stated in the table: "Ensemble method/inducer/(iterations best number)". The results show that the suggested methods have a higher g-mean score than all other classifiers in the related study.

Then, the authors applied the same classification algorithms on an oversampled dataset using SMOTE to see the best-performing classifier, and the AB-Rep tree classifier was the best. In the last step, various feature selectors for feature selection and oversampling employing SMOTE and classification using the AB-Rep tree algorithm were used. The best outcomes based on the two strategies suggested in this study and the g-mean score in [6] are shown in Table 4.19. It is obvious that the suggested strategy considerably raises the g-mean score. This work revealed the advantages of employing ensemble learning to enhance financial distress prediction and the benefits of applying cost-sensitive learning to MHOANN.

As mentioned, the authors in [6] and we used the exact dataset. Still, some differences are worth noting: (1) 10-fold cross-validation was used in the related study, while we used a 66% to 34% split for the train

and test datasets. In the related research, 90% of the data was used for model training, but still, the proposed approach showed better results. (2) Number of runs is different; we executed five different runs, while in the related research, it was ten different runs.

On the other hand, a study published in [49] used the dataset of Taiwanese companies to investigate the potential for improving classifiers' performance in forecasting financial health by integrating FRs and CGIs. After combining these two types of features, the authors evaluated five different feature selection methodologies to reduce the dimensionality of the data. They found that the combination of FRs and CGIs achieved the best results when used with SVM and SDA feature selection method. This study did not use the g-mean as an evaluation metric; instead, the authors used FPR and FNR as their evaluation metrics.

FPR is also represented as 1 - Specificity as shown in equation 4.15 [97].

$$FPR = \frac{FP}{TN + FP} = 1 - Specificity \tag{4.15}$$

FNR is also represented as 1 - Sensitivity as shown in equation 4.16 [97].

$$FNR = \frac{FN}{TP + FN} = 1 - Sensitivity \tag{4.16}$$

Therefore, the g-mean score can be computed using equation 4.17.

$$g - mean = \sqrt{(1 - FPR) \times (1 - FNR)} \tag{4.17}$$

Table 4.20 presents the best results based on the g-mean computed from FNR and FPR in [49] and the two methods suggested in this thesis. The table shows that both of the proposed methods have a higher g-mean squared. These results are based on the analysis of the Taiwanese companies' dataset.

Table 4.18: The g-mean score for standard classifiers used in the related research compared to the two methods suggested in this thesis using the Spanish companies' dataset. Boldface denotes the best outcome for g-mean per classification approach.

	Classification algorithm	**G-Mean**
Simple classifiers	k-Nearest Neighbors [6]	0.367
	Multi-layer Perceptron [6]	0.427
	Random tree [6]	**0.602**
	Naïve Bayes [6]	0.402
	J48 [6]	0.583
	Rep tree [6]	0.336
Ensemble classifers	Bagging/J48/(10) [6]	0.488
	AdaBoost/J48(20) [6]	0.609
	Decision Tree/J48/(10) [6]	0.549
	Random Forest/J48(80) [6]	0.509
	Bagging/Rep tree/(80) [6]	0.315
	AdaBoost/Rep tree (90) [6]	**0.602**
	Decision Tree/Rep tree/(10) [6]	0.414
	Random Forest/Rep tree (10) [6]	0.094
	Bagging/Random tree/(100) [6]	0.491
	AdaBoost/Random tree/(10) [6]	0.574
	Decision tree /Random tree/(20) [6]	0.532
	RtF/Random tree/(30) [6]	0.518
	Random Forest/(50) [6]	0.464
Proposed	$\text{ENS\_PSONN}_{\text{cost}}$	0.877
	$\text{ENS\_CSONN}_{\text{cost}}$	**0.893**

Table 4.19: Best results based on the g-mean score from the hybrid method used in the related research contrasted to the two methods suggested in this thesis on the Spanish companies' dataset. Boldface denotes the best outcome for g-mean per each approach.

Classifier	**Oversampling**	**Feature Selection**	**G-Mean**
Random tree [6]	No	No	0.602
AdaBoost/J48(20) [6]	No	No	0.609
Random tree [6]	Yes	No	0.696
AdaBoost/Rep tree/(90) [6]	Yes	No	**0.730**
AdaBoost/Rep tree/(90) [6]	Yes	Yes	0.720
ENS_PSONN$_{cost}$	No	No	0.877
ENS_CSONN$_{cost}$	No	No	**0.893**

Table 4.20: Best g-mean computed from the related research contrasted to the two methods suggested in this thesis on the Taiwanese Companies' Dataset. Boldface denotes the best outcome for g-mean.

Classifier	**G-Mean**
SVM+SDA+FC [49]	0.814
ENS_PSONN$_{cost}$	0.875
ENS_CSONN$_{cost}$	**0.897**

5

CONCLUSIONS

CONTENTS

As discussed, this thesis aims to improve machine learning algorithms' ability to predict companies' financial distress in two separate ways accurately, first, by comparing the effect of using eleven different oversampling methods, and second, by creating a new framework to tackle the imbalance problem in the used datasets by using an MHOANN with cost-sensitive fitness function as the base-learners in ensemble learning. Hence, in this chapter, we discuss the significant findings aligned with the objectives outlined in Chapter 1. It also discusses upcoming research that will progress in this field.

5.1 CONCLUSIONS BASED ON THE EXTERNAL METHOD

Oversampling techniques are a type of data preprocessing that can be used to enhance the performance of machine learning methods on unbalanced datasets. These methods generate artificial minority class samples, increasing the data's overall proportion. In this research, we used eleven oversampling techniques to sample a real dataset for bankruptcy prediction and evaluated their performance using a C4.5 Decision Tree classifier. The SMOTE-ENN method yielded the best results, with an FNR of 12.9% and a g-mean value of 0.87. Additionally, the decision tree produced by this method had the smallest number of leaf nodes, indicating that it could effectively classify the data with a more straightforward and interpretable model. Overall, our findings indicate that oversampling methods can be a valuable tool for enhancing the predictive capability of machine learning algorithms in the context of bankruptcy prediction.

Our study found that all the oversampling techniques we applied resulted in improved performance in terms of FNR compared to using the raw imbalanced dataset. This suggests that these methods can effectively reduce the number of false negatives, which can be especially important in situations where the negative class is more costly to miss (e.g., in fraud detection or bankruptcy prediction). Furthermore, our analysis showed that some oversampling methods improved performance more effectively than others. For example, the SMOTE-ENN method had the best FNR of all the techniques we tested, indicating that it was the most successful at correctly identifying instances of the minority class.

Overall, our findings demonstrate the potential of oversampling techniques to enhance the performance of machine learning algorithms on imbalanced datasets, particularly in the context of bankruptcy prediction. These methods can be beneficial for reducing the number of false negatives and improving the overall prediction capability of the model.

A potential reason for the solid performance of SMOTE-based methods in our study could be because of how these techniques generate synthetic minority samples. Specifically, SMOTE-based algorithms replicate instances of the minority class based on their neighbors in the feature space, with the number of neighbors serving as a critical parameter that can influence the resulting synthetic samples. We found that using $k = 13$ neighbors yielded the best results for the SMOTE-ENN method on the dataset we analyzed. It is important to note that the selection of the number of neighbors can be a complex and task-dependent issue,

as it can affect both the quality and diversity of the synthetic samples produced by the algorithm. In general, larger values of k may lead to the more accurate reproduction of the local structure of the minority class. Still, they may also result in less diversity among the synthetic samples and potentially overfitting the training data.

On the other hand, smaller values of k can introduce more variety into the synthetic samples but may also reduce their fidelity to the actual minority class distribution. Overall, our results suggest that SMOTE-based methods can effectively address imbalanced datasets in machine learning tasks, particularly when coupled with an appropriate choice of the number of neighbors. Further research could investigate the impact of this parameter on the performance of SMOTE and other oversampling methods in a broader range of datasets and tasks.

5.2 CONCLUSIONS BASED ON THE INTERNAL METHOD

In order to deal with the unbalanced distribution of financial distress datasets and improve the prediction of minor class instances, this thesis suggested using an MHOANN with a PSO or CSO as the optimization strategy and a cost-sensitive fitness function within a majority-voting ensemble learning system. Data from firms in Poland, Taiwan, and Spain were used in the tests. After that, the results of the recommended approach were compared to those obtained using the same MHOANN with a PSO or CSO but with accuracy or MSE fitness functions.

By minimizing prejudice regarding the minority class instances, our suggested strategy for forecasting financial crises performed better. Our tests' findings demonstrated that adopting a cost-sensitive fitness function significantly influenced the minority class's ability to be accurately predicted in unbalanced datasets. The g-mean showed a significant improvement, and the f1-score also saw a modestly good impact. Higher values suggest greater performance in both courses. The g-mean metric assesses the balance between the True Positive and Negative rates. The g-mean score in our investigation considerably increased when a cost-sensitive fitness function was used, indicating that this strategy significantly raised the model's accuracy for both minority and majority classes. Another statistic that takes into account a model's recall and precision is the f1-score, with higher values indicating greater performance. In our scenario, using a cost-sensitive fitness function had a promising impact on the f1-score, demonstrating that this strategy enhanced the model's overall precision-to-recall balance. Overall, these findings imply that the suggested technique can handle unbalanced

datasets in the context of predicting financial distress. Using a cost-sensitive fitness function and a PSO or CSO-optimized MHOANN, it is possible to achieve improved performance and avoid biased results.

Ensemble learning is a machine learning technique that combines the predictions of multiple models to create a more accurate and robust forecast. The majority-voting ensemble learning system works by training various models on the same dataset and then using their predictions to make a final prediction. In this case, the majority voting system involved training multiple models and then using the majority vote of their predictions as the final prediction. Using this system allowed the model to take advantage of the strengths of each model and make more accurate predictions. This ultimately resulted in an improvement in the accuracy and g-mean scores of the model and a significant increase in the f1-scores. Overall, the majority-voting ensemble learning system proved to be a valuable addition to the model, significantly improving its performance and making it more reliable for many applications. The improvement in all metrics indicates the versatility and accuracy of PSO- and CSO-optimized neural networks.

One of the main limitations of this work was the need for more access to a domain expert who could provide insights into the appropriate weights to assign to FN, which is very common in the context of cost-sensitive learning [98]. In many cases, obtaining the input of a domain expert can be crucial in determining the most effective approach to a given problem, which was unfortunately not possible in this case. Despite this limitation, the proposed method for assigning weights to FN instances could still achieve good results. However, it would have been beneficial to have the opportunity to compare the proposed approach to the recommendations of a domain expert to determine the best possible weight for FN instances.

5.3 FUTURE WORK

Several avenues for future research could build upon the findings of our study. One possibility would be to investigate alternative classification methods beyond C4.5, which may be more sensitive to the impact of oversampling techniques on imbalanced datasets. This could involve testing various methods, like support vector machines, random forests, or neural networks, to see how they compare to C4.5 in terms of performance and sensitivity to resampling. Another area of potential investigation could be evaluating other oversampling methods on different datasets. Overall, there are many directions in which this re-

search could be extended to enhance further our understanding of how to classify imbalanced datasets in machine learning tasks effectively. By continuing to explore the use of other oversampling methods and different classification methods, we can make valuable contributions to this vital area of study.

Moreover, regarding the internal approach, there are several directions in which we aim to develop further and apply the proposed method. One key area of focus is expanding the technique to other bankruptcy datasets. While the process has shown promising results on the current datasets, it is essential to fully determine its generalizability to other datasets to understand its capabilities and limitations. In addition to exploring the use of the approach with other bankruptcy datasets, we also want to investigate its potential for use in other imbalanced classification problems. Imbalanced classification is a familiar challenge in machine learning, and developing practical approaches for addressing it can have broad applications across a range of domains. Also, we are targeting alternative techniques for hyperparameter tuning, including using techniques such as AutoML [99] to find the costs of misclassified instances. Hyperparameter tuning is essential in developing any machine learning model, and finding the most effective approach can significantly impact the model's performance. By investigating a range of hyperparameter tuning methods, we aim to identify the most effective strategy for improving the performance of the proposed method. Overall, our goal is to continue advancing the proposed method and making it a valuable tool for addressing imbalanced classification problems in various domains.

Part III

APPENDICES & BIBLIOGRAPHY

A

PUBLICATIONS

CONTENTS

A.1 INTERNATIONAL JOURNALS WITH IMPACT FACTOR

1. Safi, S. A. D., Castillo, P. A., & Faris, H. (2022). Cost-Sensitive Metaheuristic Optimization-Based Neural Network with Ensemble Learning for Financial Distress Prediction. Applied Sciences, 12(14), 6918, https://doi.org/10.3390/app12146918.

A.2 INTERNATIONAL CONFERENCES

1. Alswiti, W., Faris, H., Aljawazneh, H., Safi, S., Castillo, P., Mora, A. & Alsawalqah, H. (2018, September). Empirical evaluation of advanced oversampling methods for improving bankruptcy prediction. In Proceedings of the International Conference on Time Series and Forecasting (ITISE 2018) (pp. 1495-1506).

2. Safi, S., Jawazneh, H., Mora, A. M., García-Sánchez, P., Faris, H., & Castillo, P. A. (2020). Identifying Botnets by Analysing Twitter Traffic during the Super Bowl. In IJCCI (pp. 147-154).

B

LEARNING CURVE GRAPHS

In this appendix, we present learning (convergence) curve graphs for a sample run of MHOANN on each dataset, using each of the fitness functions evaluated and the PSO and CSO optimizers. The convergence curve graphs provide a visual representation of the learning process, showing how the fitness of the algorithms improves over time.

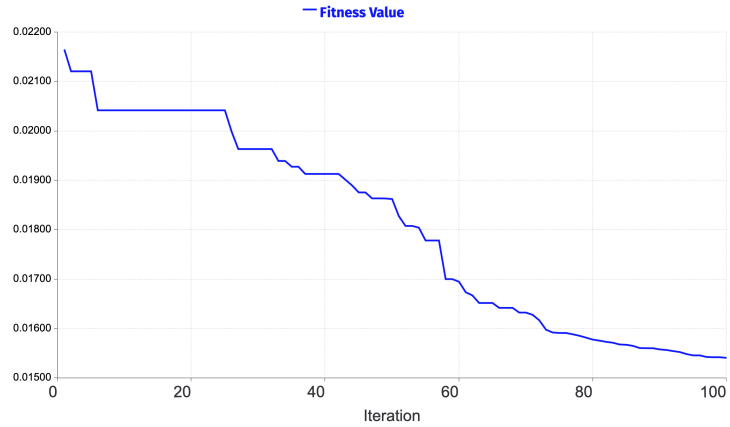

Figure B.1: MOHANN's (with MSE fitness function and PSO optimizer) learning curve on the Spanish firms' data.

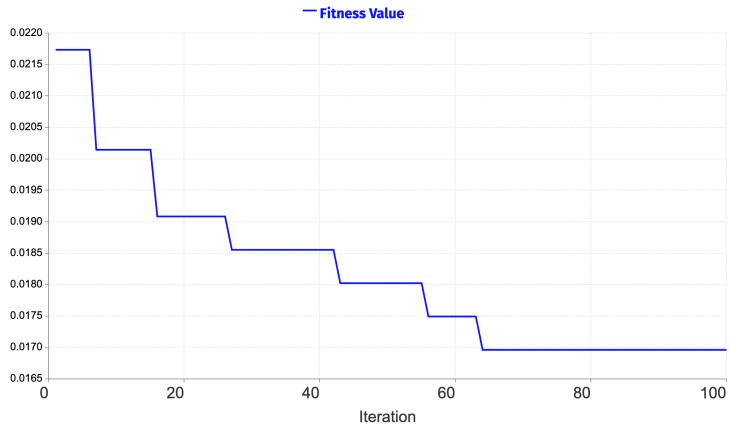

Figure B.2: MOHANN's (with accuracy fitness function and PSO optimizer) learning curve on the Spanish firms' data.

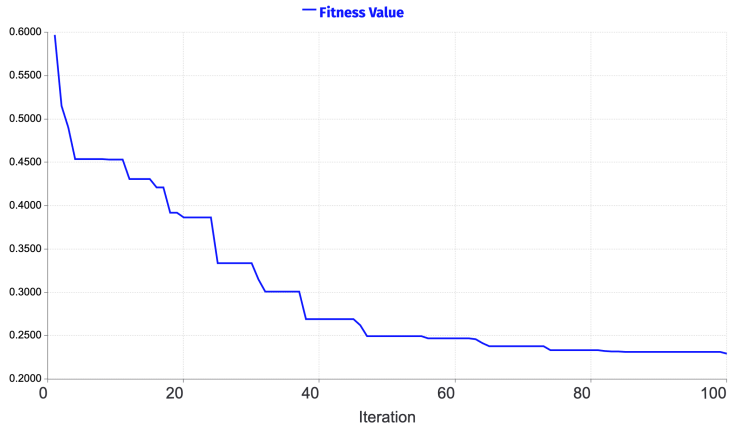

Figure B.3: MOHANN's (with cost-sensitive fitness function and PSO optimizer) learning curve on the Spanish firms' data.

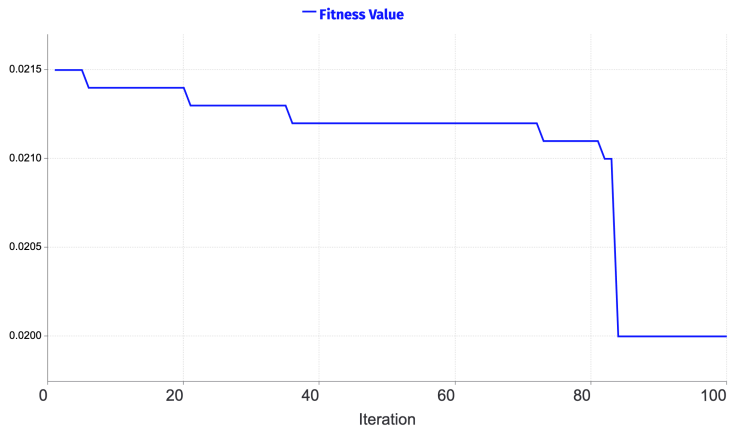

Figure B.4: MOHANN's (with MSE fitness function and CSO optimizer) learning curve on the Spanish firms' data.

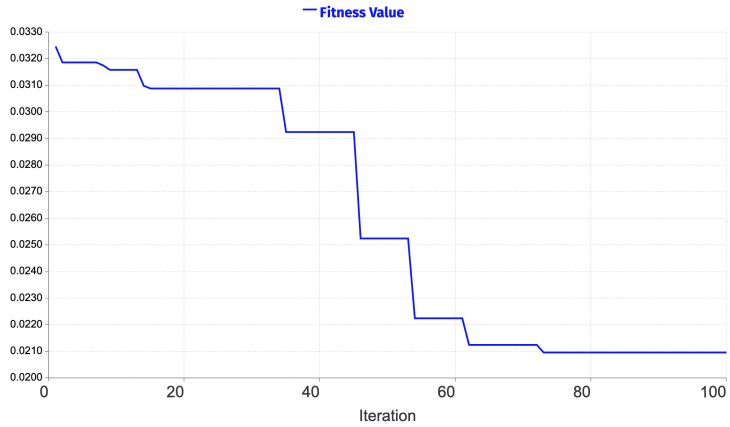

Figure B.5: MOHANN's (with accuracy fitness function and CSO optimizer) learning curve on the Spanish firms' data.

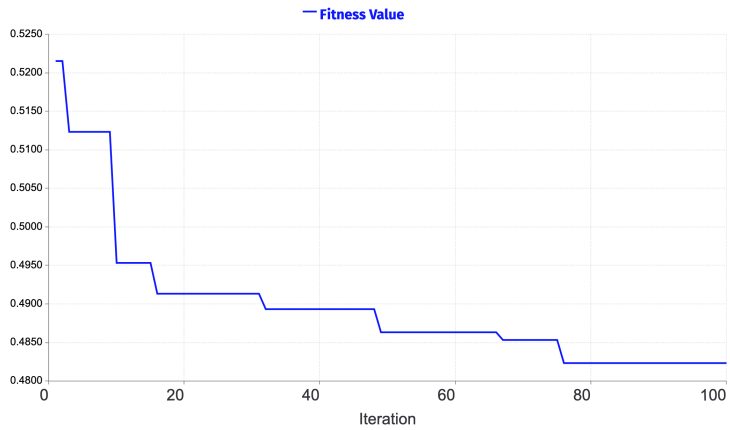

Figure B.6: MOHANN's (with cost-sensitive fitness function and CSO optimizer) learning curve on the Spanish firms' data.

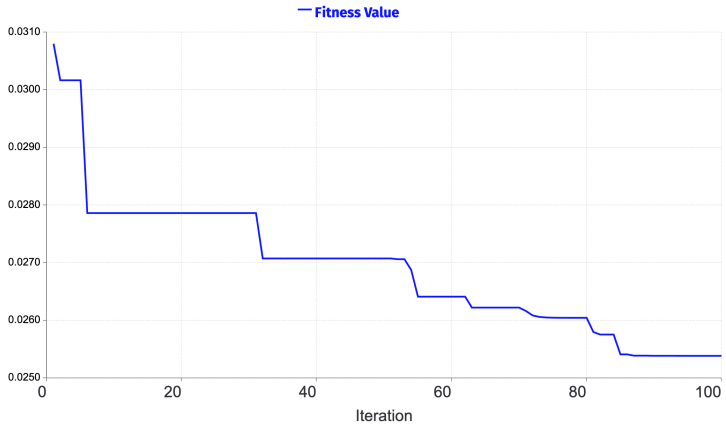

Figure B.7: MOHANN's (with MSE fitness function and PSO optimizer) learning curve on the Taiwanese firms' data.

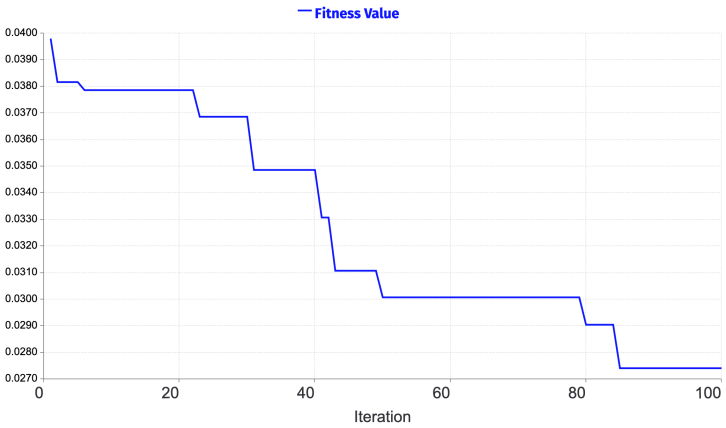

Figure B.8: MOHANN's (with accuracy fitness function and PSO optimizer) learning curve on the Taiwanese firms' data.

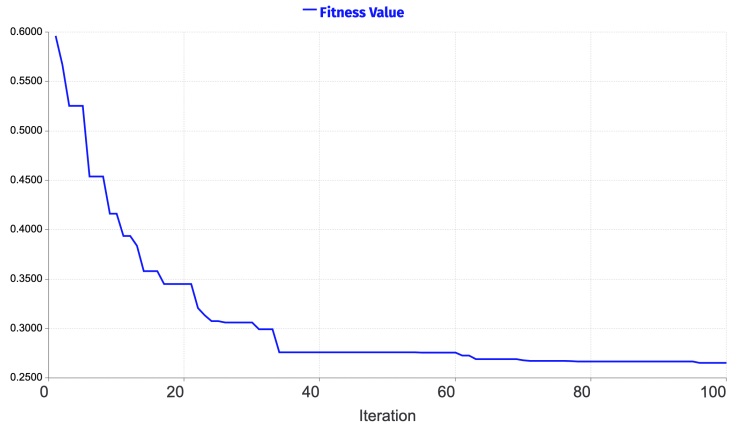

Figure B.9: MOHANN's (with cost-sensitive fitness function and PSO optimizer) learning curve on the Taiwanese firms' data.

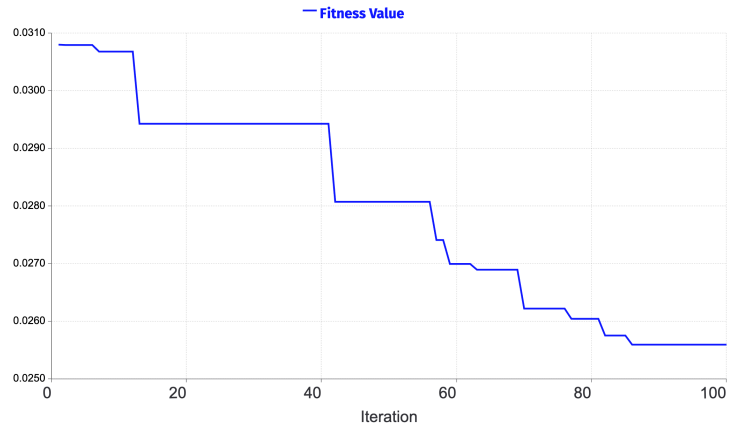

Figure B.10: MOHANN's (with MSE fitness function and CSO optimizer) learning curve on the Taiwanese firms' data.

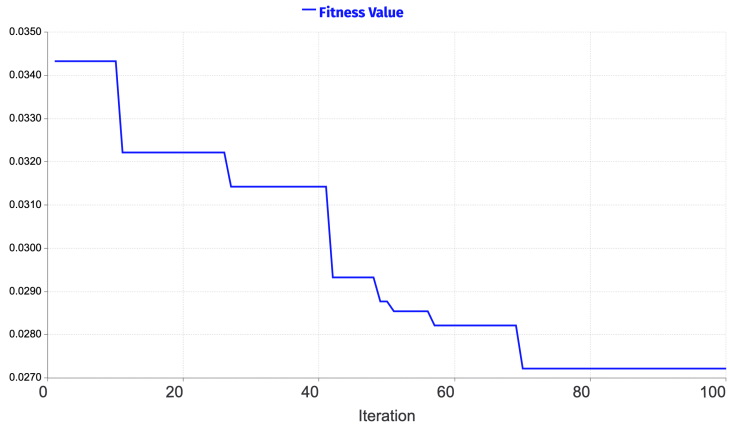

Figure B.11: MOHANN's (with accuracy fitness function and CSO optimizer) learning curve on the Taiwanese firms' data.

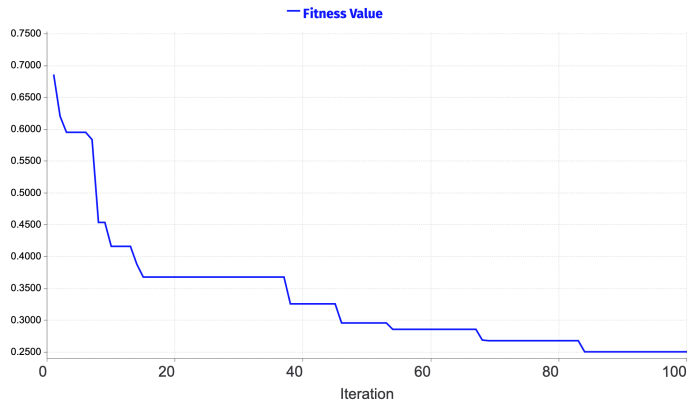

Figure B.12: MOHANN's (with cost-sensitive fitness function and CSO optimizer) learning curve on the Taiwanese firms' data.

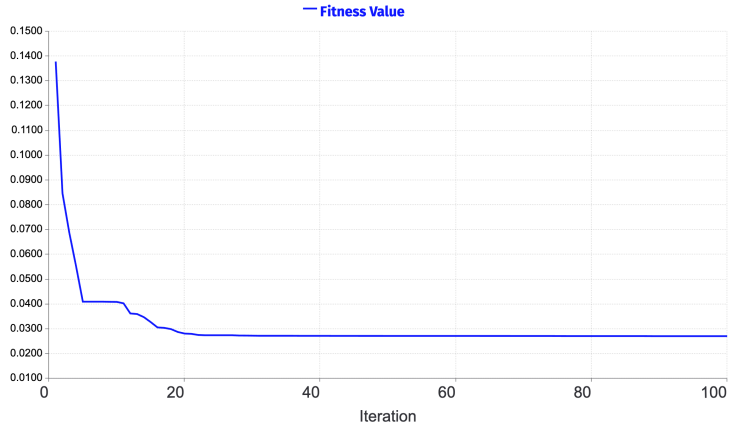

Figure B.13: MOHANN's (with MSE fitness function and PSO optimizer) learning curve on the Polish firms' data.

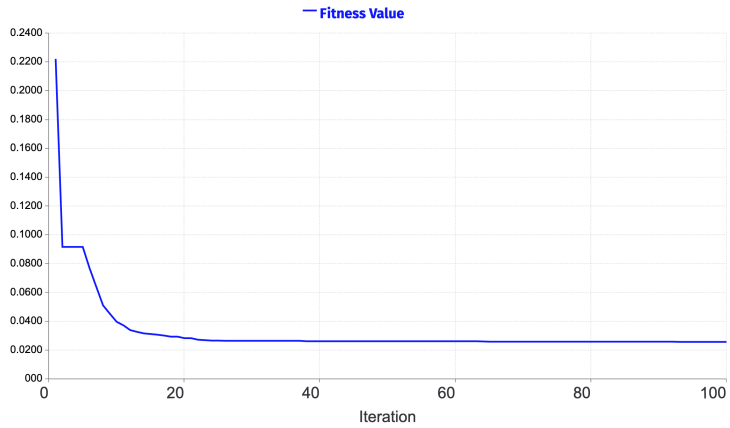

Figure B.14: MOHANN's (with accuracy fitness function and PSO optimizer) learning curve on the Polish firms' data.

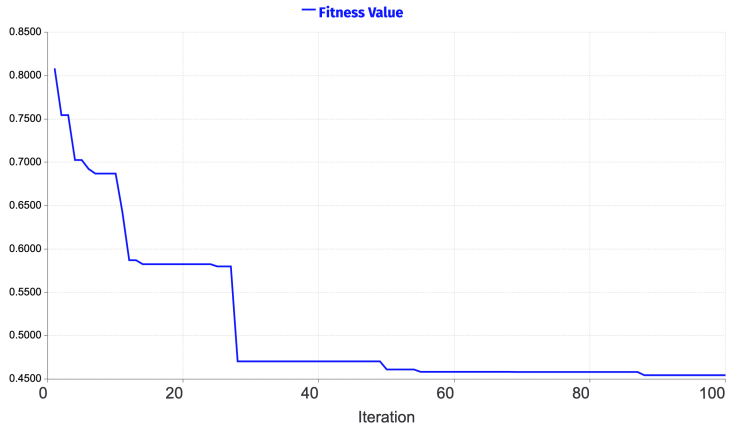

Figure B.15: MOHANN's (with cost-sensitive fitness function and PSO optimizer) learning curve on the Polish firms' data.

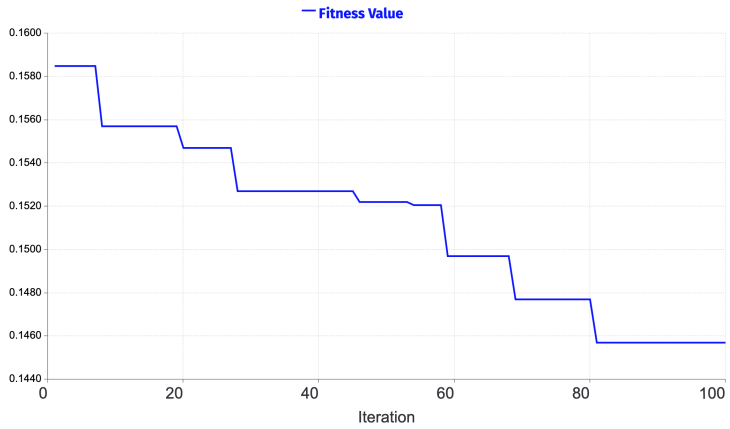

Figure B.16: MOHANN's (with MSE fitness function and CSO optimizer) learning curve on the Polish companies' data.

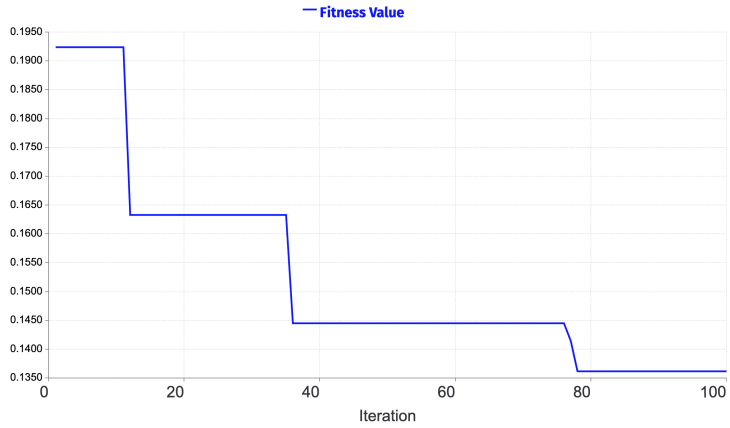

Figure B.17: MOHANN's (with accuracy fitness function and CSO optimizer) learning curve on the Polish firms' data.

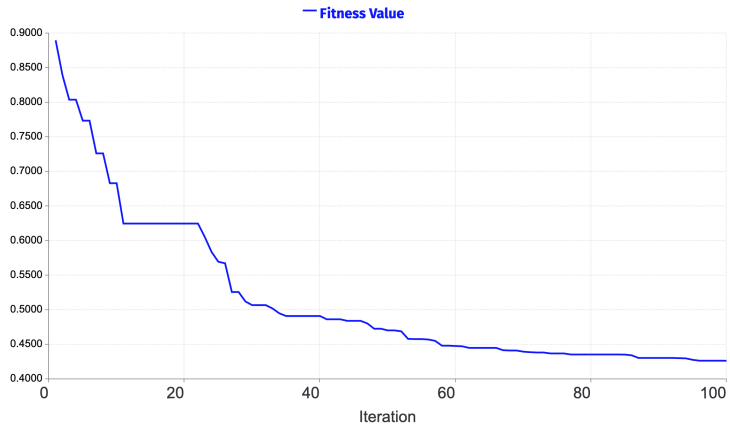

Figure B.18: MOHANN's (with cost-sensitive fitness function and CSO optimizer) learning curve on the Polish firms' data.

C

GRANTS AND SPECIAL ACKNOWLEDGEMENTS

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

BIBLIOGRAPHY

[1] Dragana Bešlić Obradović, Dejan Jakšić, Ivana Bešlić Rupić, and Mirko Andrić. "Insolvency prediction model of the company: the case of the Republic of Serbia". In: *Economic research-Ekonomska istraživanja* 31.1 (2018), pp. 139–157 (Cited on page 3).

[2] Edward I Altman and Edith Hotchkiss. *Corporate financial distress and bankruptcy: Predict and avoid bankruptcy, analyze and invest in distressed debt*. Vol. 289. John Wiley & Sons, 2010 (Cited on page 3).

[3] Yanan Zhang et al. "Towards augmented kernel extreme learning models for bankruptcy prediction: algorithmic behavior and comprehensive analysis". In: *Neurocomputing* 430 (2021), pp. 185–212 (Cited on page 3).

[4] Charu C Aggarwal. "Data classification". In: *Data mining*. Springer. 2015, pp. 285–344 (Cited on page 4).

[5] Vaishali Ganganwar. "An overview of classification algorithms for imbalanced datasets". In: *International Journal of Emerging Technology and Advanced Engineering* 2.4 (2012), pp. 42–47 (Cited on page 4).

[6] Hossam Faris et al. "Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: a case from the Spanish market". In: *Progress in Artificial Intelligence* 9.1 (2020), pp. 31–53 (Cited on pages 5, 33, 37, 66, 68, 69).

[7] D Devikanniga, K Vetrivel, and N Badrinath. "Review of meta-heuristic optimization based artificial neural networks and its applications". In: *Journal of Physics: Conference Series*. Vol. 1362. 1. IOP Publishing. 2019, p. 012074 (Cited on pages 5, 11, 17).

[8] Federico Marini and Beata Walczak. "Particle swarm optimization (PSO). A tutorial". In: *Chemometrics and Intelligent Laboratory Systems* 149 (2015), pp. 153–165 (Cited on pages 6, 17, 19).

[9] Ran Cheng and Yaochu Jin. "A competitive swarm optimizer for large scale optimization". In: *IEEE transactions on cybernetics* 45.2 (2014), pp. 191–204 (Cited on pages 6, 17, 20, 21).

[10] Yulong Wang, Haoxin Zhang, and Guangwei Zhang. "cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks". In: *Swarm and Evolutionary Computation* 49 (2019), pp. 114–123 (Cited on page 6).

[11] A Kaveh and VR Mahdavi. "A hybrid CBO–PSO algorithm for optimal design of truss structures with dynamic constraints". In: *Applied Soft Computing* 34 (2015), pp. 260–273 (Cited on page 6).

[12] Mohamed Hosni, Ibtissam Abnane, Ali Idri, Juan M Carrillo de Gea, and José Luis Fernández Alemán. "Reviewing ensemble classification methods in breast cancer". In: *Computer methods and programs in biomedicine* 177 (2019), pp. 89–112 (Cited on pages 6, 63).

[13] Bartosz Krawczyk. "Learning from imbalanced data: open challenges and future directions". In: *Progress in Artificial Intelligence* 5.4 (2016), pp. 221–232 (Cited on page 10).

[14] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. "Learning from class-imbalanced data: Review of methods and applications". In: *Expert systems with applications* 73 (2017), pp. 220–239 (Cited on pages 10, 16).

[15] Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. "Machine learning with oversampling and undersampling techniques: overview study and experimental results". In: *2020 11th international conference on information and communication systems (ICICS)*. IEEE. 2020, pp. 243–248 (Cited on page 11).

[16] Mayuri S Shelke, Prashant R Deshmukh, and Vijaya K Shandilya. "A review on imbalanced data handling using undersampling and oversampling technique". In: *Int. J. Recent Trends Eng. Res* 3.4 (2017), pp. 444–449 (Cited on page 11).

[17] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. "A study of the behavior of several methods for balancing machine learning training data". In: *ACM SIGKDD explorations newsletter* 6.1 (2004), pp. 20–29 (Cited on pages 12, 13).

[18] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357 (Cited on page 12).

[19] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning". In: *International conference on intelligent computing*. Springer. 2005, pp. 878–887 (Cited on page 13).

[20] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem". In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2009, pp. 475–482 (Cited on page 14).

[21] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning". In: *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE. 2008, pp. 1322–1328 (Cited on page 14).

[22] Sheng Tang and Si-Ping Chen. "The generation mechanism of synthetic minority class examples". In: *2008 International Conference on Information Technology and Applications in Biomedicine*. IEEE. 2008, pp. 444–447 (Cited on page 15).

[23] Jerzy Stefanowski and Szymon Wilk. "Selective pre-processing of imbalanced data for improving classification performance". In: *International Conference on Data Warehousing and Knowledge Discovery*. Springer. 2008, pp. 283–292 (Cited on page 15).

[24] Krystyna Napierała, Jerzy Stefanowski, and Szymon Wilk. "Learning from imbalanced data in presence of noisy and borderline examples". In: *International conference on rough sets and current trends in computing*. Springer. 2010, pp. 158–167 (Cited on page 15).

[25] Gilles Cohen, Mélanie Hilario, Hugo Sax, Stéphane Hugonnet, and Antoine Geissbuhler. "Learning from imbalanced data in surveillance of nosocomial infection". In: *Artificial intelligence in medicine* 37.1 (2006), pp. 7–18 (Cited on page 16).

[26] Alberto Fernández, Victoria LóPez, Mikel Galar, Mar'1A José Del Jesus, and Francisco Herrera. "Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches". In: *Knowledge-based systems* 42 (2013), pp. 97–110 (Cited on page 16).

[27] Adel Ghazikhani, Reza Monsefi, and Hadi Sadoghi Yazdi. "Online cost-sensitive neural network classifiers for non-stationary and imbalanced data streams". In: *Neural computing and applications* 23.5 (2013), pp. 1283–1295 (Cited on page 16).

[28] Cristiano L Castro and Antônio P Braga. "Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data". In: *IEEE transactions on neural networks and learning systems* 24.6 (2013), pp. 888–899 (Cited on page 16).

[29] Chris M Bishop. "Neural networks and their applications". In: *Review of scientific instruments* 65.6 (1994), pp. 1803–1832 (Cited on page 17).

[30] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999 (Cited on page 17).

[31] Semih Kalender and Unal Kose. "Simulated annealing algorithm for training artificial neural networks". In: *Expert Systems with Applications* 33.1 (2007), pp. 129–134 (Cited on page 17).

[32] X Tao, Y Chen, and S Wang. "Simulated annealing neural network classifier with hybrid learning algorithm". In: *Neurocomputing* 65.1 (2005), pp. 189–206 (Cited on page 17).

[33] Xin-She Yang. "Metaheuristic optimization: algorithm analysis and open problems". In: *International Symposium on Experimental Algorithms*. Springer. 2011, pp. 21–32 (Cited on page 18).

[34] Shubham Gupta, Hammoudi Abderazek, Betül Sultan Yıldız, Ali Riza Yildiz, Seyedali Mirjalili, and Sadiq M Sait. "Comparison of metaheuristic optimization algorithms for solving constrained mechanical design optimization problems". In: *Expert Systems with Applications* 183 (2021), p. 115351 (Cited on page 18).

[35] Surabhi Kaul and Yogesh Kumar. "Nature-Inspired metaheuristic algorithms for constraint handling: challenges, issues, and research perspective". In: *Constraint Handling in Metaheuristics and Applications* (2021), pp. 55–80 (Cited on page 18).

[36] Russell Eberhart and James Kennedy. "A new optimizer using particle swarm theory". In: *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*. Ieee. 1995, pp. 39–43 (Cited on page 19).

[37] Hou-Ping Dai, Dong-Dong Chen, and Zhou-Shun Zheng. "Effects of random values for particle swarm optimization algorithm". In: *Algorithms* 11.2 (2018), p. 23 (Cited on page 20).

[38] Omer Sagi and Lior Rokach. "Ensemble learning: A survey". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4 (2018), e1249 (Cited on pages 22, 48).

[39] Thomas G Dietterich. "Ensemble methods in machine learning". In: *Multiple Classifier Systems* (2000), pp. 1–15 (Cited on pages 22, 23).

[40] Cha Zhang and Yunqian Ma. *Ensemble machine learning: methods and applications*. Springer, 2012 (Cited on page 22).

[41] R Polikar. *Ensemble Learning in Ensemble Machine Learning: Methods and Applications; Zhang, C., Ma, Y., Eds*. 2012 (Cited on page 23).

[42] Alexander Strehl and Joydeep Ghosh. "Impact of noise on clustering". In: *Journal of Machine Learning Research* 1 (2000), pp. 5–32 (Cited on page 23).

[43] Thomas G Dietterich et al. "Ensemble learning". In: *The handbook of brain theory and neural networks* 2.1 (2002), pp. 110–125 (Cited on page 23).

[44] Robi Polikar. "Ensemble based systems in decision making". In: *IEEE Circuits and systems magazine* 6.3 (2006), pp. 21–45 (Cited on page 23).

[45] P Ravi Kumar and Vadlamani Ravi. "Bankruptcy prediction in banks and firms via statistical and intelligent techniques–A review". In: *European journal of operational research* 180.1 (2007), pp. 1–28 (Cited on page 27).

[46] Edward I Altman. "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy". In: *The journal of finance* 23.4 (1968), pp. 589–609 (Cited on page 27).

[47] Vicente García, Ana Isabel Marqués, and Jose Salvador Sánchez. "Improving risk predictions by preprocessing imbalanced credit data". In: *International conference on neural information processing*. Springer. 2012, pp. 68–75 (Cited on page 28).

[48] José A Sáez, Julián Luengo, Jerzy Stefanowski, and Francisco Herrera. "SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering". In: *Information Sciences* 291 (2015), pp. 184–203 (Cited on page 28).

[49] Deron Liang, Chia-Chi Lu, Chih-Fong Tsai, and Guan-An Shih. "Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study". In: *European Journal of Operational Research* 252.2 (2016), pp. 561–572 (Cited on pages 28, 38, 67, 69).

[50] Tuong Le, Mi Young Lee, Jun Ryeol Park, and Sung Wook Baik. "Oversampling techniques for bankruptcy prediction: Novel features from a transaction dataset". In: *Symmetry* 10.4 (2018), p. 79 (Cited on page 28).

[51] Sergio González, Salvador García, Sheng-Tun Li, and Francisco Herrera. "Chain based sampling for monotonic imbalanced classification". In: *Information Sciences* 474 (2019), pp. 187–204 (Cited on page 28).

[52] Sheikh Rabiul Islam, William Eberle, Sheikh K Ghafoor, Sid C Bundy, Douglas A Talbert, and Ambareen Siraj. "Investigating bankruptcy prediction models in the presence of extreme class imbalance and multiple stages of economy". In: *arXiv preprint arXiv:1911.09858* (2019) (Cited on page 28).

[53] Peter Gnip, Liberios Vokorokos, and Peter Drotár. "Selective oversampling approach for strongly imbalanced data". In: *PeerJ Computer Science* 7 (2021), e604 (Cited on page 29).

[54] Myoung-Jong Kim, Dae-Ki Kang, and Hong Bae Kim. "Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction". In: *Expert Systems with Applications* 42.3 (2015), pp. 1074–1082 (Cited on page 29).

[55] Ibomoiye Domor Mienye and Yanxia Sun. "Performance analysis of cost-sensitive learning methods with application to imbalanced medical data". In: *Informatics in Medicine Unlocked* 25 (2021), p. 100690 (Cited on page 29).

[56] Barenya Bikash Hazarika and Deepak Gupta. "Density-weighted support vector machines for binary class imbalance learning". In: *Neural Computing and Applications* 33.9 (2021), pp. 4243–4261 (Cited on page 29).

[57] Paul PM Pompe and AJ Feelders. "Using machine learning, neural networks, and statistics to predict corporate bankruptcy". In: *Computer-Aided Civil and Infrastructure Engineering* 12.4 (1997), pp. 267–276 (Cited on page 30).

[58] Jinwoo Baek and Sungzoon Cho. "Bankruptcy prediction for credit risk using an auto-associative neural network in Korean firms". In: *2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings.* IEEE. 2003, pp. 25–29 (Cited on page 30).

[59] Indranil Bose and Raktim Pal. "Predicting the survival or failure of click-and-mortar corporations: A knowledge discovery approach". In: *European Journal of Operational Research* 174.2 (2006), pp. 959–982 (Cited on page 30).

[60] Tzong-Huei Lin. "A cross model study of corporate financial distress prediction in Taiwan: Multiple discriminant analysis, logit, probit and neural networks models". In: *Neurocomputing* 72.16-18 (2009), pp. 3507–3516 (Cited on page 30).

[61] Ali Mansouri, Arezoo Nazari, and Morteza Ramazani. "A comparison of artificial neural network model and logistics regression in prediction of companies' bankruptcy (A case study of Tehran stock exchange)". In: *International Journal of Advanced Computer Research* 6.24 (2016) (Cited on page 30).

[62] David J Montana, Lawrence Davis, et al. "Training feedforward neural networks using genetic algorithms." In: *IJCAI*. Vol. 89. 1989, pp. 762–767 (Cited on page 30).

[63] Rui Mendes, Paulo Cortez, Miguel Rocha, and José Neves. "Particle swarms for feedforward neural network training". In: *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*. Vol. 2. IEEE. 2002, pp. 1895–1899 (Cited on page 31).

[64] Abdollah Ansari, Ibrahim Said Ahmad, Azuraliza Abu Bakar, and Mohd Ridzwan Yaakub. "A hybrid metaheuristic method in training artificial neural network for bankruptcy prediction". In: *IEEE Access* 8 (2020), pp. 176640–176650 (Cited on page 31).

[65] Israa Al-Badarneh, Maria Habib, Ibrahim Aljarah, and Hossam Faris. "Neuro-evolutionary models for imbalanced classification problems". In: *Journal of King Saud University-Computer and Information Sciences* (2020) (Cited on page 31).

[66] Seyed Jalaleddin Mousavirad, Gerald Schaefer, Seyed Mohammad Jafar Jalali, and Iakov Korovin. "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*. 2020, pp. 1402–1408 (Cited on page 31).

[67] Ghazaleh Alibabaee and Mohammadhamed Khanmohammadi. "The Study of the Predictive Power of Meta-heuristic Algorithms to Provide a Model for Bankruptcy prediction". In: *International Journal of Finance & Managerial Accounting* 7.26 (2022), pp. 33–51 (Cited on page 31).

[68] Esteban Alfaro, Noelia García, Matías Gámez, and David Elizondo. "Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks". In: *Decision Support Systems* 45.1 (2008), pp. 110–122 (Cited on page 31).

[69] Jie Sun and Hui Li. "Listed companies' financial distress prediction based on weighted majority voting combination of multiple classifiers". In: *Expert Systems with Applications* 35.3 (2008), pp. 818–827 (Cited on page 31).

[70] Bartosz Krawczyk, Michał Woźniak, and Gerald Schaefer. "Cost-sensitive decision tree ensembles for effective imbalanced classification". In: *Applied Soft Computing* 14 (2014), pp. 554–562 (Cited on page 31).

[71] Maciej Zięba, Sebastian K Tomczak, and Jakub M Tomczak. "Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction". In: *Expert systems with applications* 58 (2016), pp. 93–101 (Cited on pages 32, 38).

[72] Tomasz Pisula. "An ensemble classifier-based scoring model for predicting bankruptcy of polish companies in the Podkarpackie Voivodeship". In: *Journal of Risk and Financial Management* 13.2 (2020), p. 37 (Cited on page 32).

[73] Kun Chang Lee, Ingoo Han, and Youngsig Kwon. "Hybrid neural network models for bankruptcy predictions". In: *Decision Support Systems* 18.1 (1996), pp. 63–72 (Cited on page 32).

[74] Sung-Hwan Min, Jumin Lee, and Ingoo Han. "Hybrid genetic algorithms and support vector machines for bankruptcy prediction". In: *Expert systems with applications* 31.3 (2006), pp. 652–660 (Cited on page 32).

[75] Yuchun Tang, Yan-Qing Zhang, Nitesh V Chawla, and Sven Krasser. "SVMs modeling for highly imbalanced classification". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.1 (2008), pp. 281–288 (Cited on page 32).

[76] D Karthik Chandra, Vadlamani Ravi, and Indranil Bose. "Failure prediction of dotcom companies using hybrid intelligent techniques". In: *Expert Systems with applications* 36.3 (2009), pp. 4830–4837 (Cited on page 32).

[77] Nazeeh Ghatasheh et al. "Cost-sensitive ensemble methods for bankruptcy prediction in a highly imbalanced data distribution: A real case from the Spanish market". In: *Progress in Artificial Intelligence* 9.4 (2020), pp. 361–375 (Cited on page 33).

[78] Wirot Yotsawat, Pakaket Wattuya, and Anongnart Srivihok. "A Novel Method for Credit Scoring Based on Cost-Sensitive Neural Network Ensemble". In: *IEEE Access* 9 (2021), pp. 78521–78537 (Cited on page 33).

[79] Yao Zou, Changchun Gao, and Han Gao. "Business Failure Prediction Based on a Cost-Sensitive Extreme Gradient Boosting Machine". In: *IEEE Access* 10 (2022), pp. 42623–42639 (Cited on page 33).

[80] I Román, ME Gómez, JMD la Torre, JJ Merelo, and AM Mora. "Predicting financial distress: Relationship between continued losses and legal bankrupcy". In: *Proceedings of the 27th Annual Congress European Accounting Association, Dublin, Ireland*. 2006 (Cited on page 36).

[81] Wedyan Alswiti et al. "Empirical evaluation of advanced oversampling methods for improving bankruptcy prediction". In: *Proceedings of the International Conference on Time Series and Forecasting (ITISE 2018)*. 2018, pp. 1495–1506 (Cited on page 38).

[82] Steven L Salzberg. *C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993*. 1994 (Cited on page 39).

[83] Shoushan Li, Zhongqing Wang, Guodong Zhou, and Sophia Yat Mei Lee. "Semi-supervised learning for imbalanced sentiment classification". In: *Twenty-Second International Joint Conference on Artificial Intelligence*. 2011 (Cited on page 39).

[84] Lawrence A Weiss and Vedran Capkun. "The impact of incorporating the cost of errors into bankruptcy prediction models". In: *Available at SSRN 651261* (2005) (Cited on page 40).

[85] Salah Al-Deen Safi, Pedro A Castillo, and Hossam Faris. "Cost-Sensitive Metaheuristic Optimization-Based Neural Network with Ensemble Learning for Financial Distress Prediction". In: *Applied Sciences* 12.14 (2022), p. 6918 (Cited on page 42).

[86] Xin Yao. "Evolving artificial neural networks". In: *Proceedings of the IEEE* 87.9 (1999), pp. 1423–1447 (Cited on page 45).

[87] Giuliano Armano, Michele Marchesi, and Andrea Murru. "A hybrid genetic-neural architecture for stock indexes forecasting". In: *Information Sciences* 170.1 (2005), pp. 3–33 (Cited on page 45).

[88] Yuehui Chen, Bo Yang, Jiwen Dong, and Ajith Abraham. "Time-series forecasting using flexible neural tree model". In: *Information sciences* 174.3-4 (2005), pp. 219–235 (Cited on page 45).

[89] Xin Yao and Yong Xu. "Recent advances in evolutionary computation". In: *Journal of Computer Science and Technology* 21.1 (2006), pp. 1–18 (Cited on page 45).

[90] Larry Bull. "On model-based evolutionary computation". In: *Soft Computing* 3.2 (1999), pp. 76–82 (Cited on page 46).

[91] Beatriz A Garro and Roberto A Vázquez. "Designing artificial neural networks using particle swarm optimization algorithms". In: *Computational intelligence and neuroscience* 2015 (2015) (Cited on page 47).

[92] Juan Carlos Gómez, Fernando Hernández, Carlos A Coello Coello, Guillermo Ronquillo, and Antonio Trejo. "Flame classification through the use of an artificial neural network trained with a genetic algorithm". In: *Mexican International Conference on Artificial Intelligence*. Springer. 2013, pp. 172–184 (Cited on page 47).

[93] Hossam Faris, Ibrahim Aljarah, Nailah Al-Madi, and Seyedali Mirjalili. "Optimizing the learning process of feedforward neural networks using lightning search algorithm". In: *International Journal on Artificial Intelligence Tools* 25.06 (2016), p. 1650033 (Cited on page 49).

[94] You Shyang Chen. "Building a Hybrid Prediction Model to Evaluation of Financial Distress Corporate". In: *Applied Mechanics and Materials*. Vol. 651. Trans Tech Publ. 2014, pp. 1543–1546 (Cited on page 49).

[95] Qi Yu, Yoan Miche, Eric Séverin, and Amaury Lendasse. "Bankruptcy prediction using extreme learning machine and financial expertise". In: *Neurocomputing* 128 (2014), pp. 296–302 (Cited on page 49).

[96] Vicente Garcia, Ana I Marques, and J Salvador Sanchez. "An insight into the experimental design for credit risk and corporate bankruptcy prediction systems". In: *Journal of Intelligent Information Systems* 44.1 (2015), pp. 159–189 (Cited on page 62).

[97] David MW Powers. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In: *arXiv preprint arXiv:2010.16061* (2020) (Cited on page 67).

[98] Alberto Fernandez, Salvador Garcia, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*. Vol. 10. Springer, 2018 (Cited on page 74).

[99] Zuohui Fu, Yikun Xian, Shijie Geng, Gerard de Melo, and Yongfeng Zhang. "Journal: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021". In: () (Cited on page 75).