ISSN 2007-9737

Prediction of Enterprise Financial Health Using Machine Learning and Financial Reasons for Taiwan Economic Companies

Daniel Ruelas[1], Fernando-Gaxiola[1,*], Alain Manzo-Martinez[1],
Raymundo Cornejo-Garcia[1], Jesus Soto-Vega[2]

[1] Universidad Autónoma de Chihuahua, Facultad de Ingeniería,
Mexico

[2] Universidad Autónoma de Baja California, Facultad de Ingeniería,
Mexico

dan.rm91@gmail.com, {lgaxiola, amanzo, rcornejo} @uach.mx, jesussoto@uaim.edu.mx

Abstract. In actuality, the financial investment in Enterprises of the world is common. This investment is performed using internet platforms and value markets. This can generate a loss for many investors due to the uncertainty of the future financial health of the Enterprise. A comparison for the prediction of financial health based on algorithms of machine learning, particularly Support Vector Machine (SVM), Artificial Neural Networks (ANN), K-Nearest Neighbors (KNN), and Adaptive Neuro-Fuzzy Networks (ANFIS) is presented. The database of the Taiwan Economic Journal from 1999 to 2009 is used, with 95 financial ratios of enterprises financially healthiest and with bankruptcy problems. In the ANN, the epoch number, numbers of neurons, activation functions in each layer, loss function, and learning rate are tested; also, an architecture of a Convolutional Neural Network (CNN) is implemented. In SVM, the experiments are performed using different kernels, polynomial, RBF, linear. Besides, the size of C, size of gamma, and size of the polynomial are varied. In KNN, experiments with different numbers of neighbors, types of weight, and values of P are realized. In ANFIS, experiments with variants of the numbers of fuzzy rules, quantity and type of membership functions, number of epochs, and input dimensions are performed. Optimization using Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) of the three models with the best results are performed; the optimization is based on the search for the best hyperparameters that would provide a higher accuracy. The neural network models presented the best average for all the proposed tests.

Keywords. Prediction, machine learning, financial.

1 Introduction

Currently, the detection of financial problems through the use of intelligent algorithms is a subject that continues to be investigated. It has been proven that some algorithms can make predictions earlier and more effectively than a professional in the area. For example, Suryawanshi et al. [31], performed the prediction of cryptocurrencies price for Bitcoin, Ethereum, and Ripple using a Long Short-Term Memory architecture (LSTM).

Although the volatility of cryptocurrencies is high and almost unpredictable, this algorithm can help invest. There is also the case of trading, which talks about how Machine Learning (ML) algorithms can generate predictions that are impossible for humans to generate.

This is because trading operations are too fast for a human. In addition to the fact that, combined with a large amount of data, it reduces risks and obtains greater benefits.

It is estimated that, currently, 4 out of 5 trading operations are done automatically. For this task, Chen et al. [5] generated a trading algorithm using the Light Gradient Boosting Machine (LightGBM) algorithm to construct the minimum variance portfolio of the mean-variance model with a Conditional Value at Risk (CVaR) constraint, to generate an efficient investment portfolio.

Finding an effective solution for the issue of predicting financial bankruptcies is important for

doi: 10.13053/CyS-28-2-5021

different areas, be it business, government, or even social. The main reason for this work is to detect financial difficulties early, using machine learning algorithms, to avoid financial investments in companies that are at imminent risk of bankruptcy.

First, the concept of bankruptcy or financial bankruptcy in a company must be clarified. In López [21], the concept of bankruptcy is defined as the company's inability to meet its debts with the available resources, so it must cease its activities immediately. In little words, net patrimony is negative, when the total value of assets is not enough to pay off creditors. The main characteristics of bankruptcy according to López [21] are:

1 Situation of irreversible or permanent disappearance: It occurs when the company declares bankruptcy and is in the process of disappearing. It is permanent.

2 The assets are less than the liabilities: It occurs when in a company, the debts exceed the assets.

3 Affects the entire company: Creditors dispute parts of the company and legally affect the company as a whole. General bankruptcy can be avoided by selling subsidiaries.

4 It must be legally classified: It is a situation provided by law, to avoid any fraud.

In this work, a financial analysis is proposed, which is given from the observation of patterns of the financial ratios of companies. These financial reasons come from the financial models or basic financial statements.

The set of basic financial statements, the income statement, cash flow, and balance sheet are the general x-rays of the company and these are the financial ratios to use.

Different financial ratios use these basic financial statements to define the financial health of a company. Guajardo and Andrade [13] present some financial ratios, such as the acid test and accounts receivable turnover. The first ratios show how much liquidity the company has, so it divides the product of the subtraction of the current assets account (balance sheet) and inventories (balance sheet), between the current liabilities (also from the balance sheet).

Account turnover indicates how many times a year the accounts provided by the company are rotated. It is obtained by dividing net sales (income statement) by accounts receivable (balance sheet). As well as these examples, 95 financial ratios are raised in the investigation, which will be discussed later.

Santoso and Wibowo in 2018 [27], use machine learning models such as K-nearest neighbors, neural networks, support vector machines, and neuro-fuzzy networks, to classify a company as stable or bankrupt. Here are some examples of related work using similar algorithms and architectures:

Xie, Lu, and Yu in 2011 [39], mentioned in their research that they can predict the bankruptcy of a company using SVM, with variables similar to the previous ones, such as financial profitability and return on investment.

In 2017, Mselmi et al. [22], used logistic regression, neural networks, SVM, least squares, and a hybrid model of least squares with SVM to classify companies with possible bankruptcy.The main contributions of this research are as follow:

- Identify a functional financial dataset with financial ratios adequate for the research.
- Implementing and testing of Adaptive Neural Fuzzy Inference System (ANFIS), Neural Networks (NN), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM) to classify the financial state of a company through the use of financial ratios, derived from the basic financial statements.
- Optimization of the models implemented using bio-inspired optimization algorithms, like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), to improve the results for the classification.

Currently, technology is generating better living conditions and knowledge that helps in decision-making. Within this technology, machine learning algorithms are integrated in different areas. This type of technology has been used in most areas of daily life due to more efficient and accurate algorithms, and more capable hardware [25].

This research is important in the business, government, and social spheres since it has been proven that classification can be made using machine learning algorithms, with a degree of confidence greater than the financial experts in the area. With this, strong financial decisions can be made that solve problems such as knowing if it's possible to invest in a company and if it is redeemable through credit, among other things.

Currently, the industry does not have a sufficient supply of solutions on the market, therefore it is still being investigated, so the objective of this and other investigations is to risk people's assets as little as possible and generate knowledge that can be used for future research.

Research related to the investigation topic of the paper is described as follows:

Stasko et al. [29], utilized the Altman financial model to calculate a future prediction of bankruptcy in Companies from Letonia using 5 financial ratios; the result of this method is a probability of imminent bankruptcy, gray zone, and secure zone.

Tabbakh et al. [32], used a dataset from Polonia with 43,405 companies; For this case, remove the instances that have null data, normalization, and the "SMOTE" technique are used for preprocessing, besides the SVM model for prediction gave a 98.8 % in accuracy.

Kansal and Sharma [18], mentioned the use of SVM and Neural network models for the prediction in small and medium-sized companies of a database from France.

Arieshanti et al. [2], used a database of 240 companies with 30 financial ratios, and for the prediction implemented an SVM model with a lineal kernel, variable C of 1, and a neural network with sigmoidal activation function and 5,000 epochs for training.

Xie et al [39], used 260 Chinese companies with 28 financial ratios, half of the companies are bankrupt and half are not.

Two SVM models are implemented, first with the aforementioned database and another with aggregate corporate governance and external market variables.

Narvekar and Guha [24], proposed a SVM model for prediction in the database "Compustat". This database contains 75 financial ratios from 21,114 American companies, of which there are 1,212 companies in bankruptcy and 19,902 stables.

For preprocessing, null values are eliminated which allows removal of 18 financial ratios, and the SMOTE technique to balance the data. Santoso and Wibowo [27], proposed an SVM model with a linear kernel to perform a prediction in a database of Indonesian companies with 20 financial ratios.

Shetty et al. [28], proposed SVM and Neural network models for the prediction in small and medium-sized companies of a database from Belgium. Wang [37], proposed an Artificial Neural Network model for prediction in a database called "Qualitative Bankruptcy", which uses disconnection of neurons, truncation technique, softmax activation function, Adam optimizer, and categorical cross entropy loss function.

Abdou et al. [1], used a neural network model to be implemented in a database of 14 financial and 3 non-financial indicators of companies registered with banks in the Midwest.

Sudarsanam [30], proposed a neuro-fuzzy network model that used only the variables of the Altman method; this model is implemented in a database of 125 companies from the Indian economic monitoring center, where some of them are bankrupt.

Arora and Saini [3] proposed an ANFIS model with Altman's variables and 3 bell-type membership functions; for the prediction, 1,000 companies and a total of 4 years of maximum prediction are used.

Muslim and Dasril [23], used a KNN model on a database of companies in Poland, with 65 financial ratios, and the data are normalized and scaled.

In this work, take in consideration the literature, the implementation of several models of machine learning and a selected number of financial ratios is performed to achieve the prediction of bankruptcy.

The rest of this paper is organized as follows: Theoretical background is presented in Section 2, in which the theory of the financial statements and computational models are presented; the methodology implemented is given in Section 3.

Experiment analysis is presented in Section 4 and, Section 5 and 6 gives the Discussion and final Conclusion.

Table 1. Examples of financial ratios

Financial Statement	Ratio
LIQUIDITY	Current ratio
	Acid test
ACTIVITY	Inventory Turnover
	Average Inventory
	Accounts receivable turnover
	Average collection period
	Turnover of current assets
	Fixed asset turnover
	Total asset turnover
	Debt ratio
	Interest coverage
INDEBTEDNESS	Gross profit margin
	Operating sales margin
	Net profit margin
COST EFFECTIVENESS	Return on operating investment
	Return on total investment
	Income on capital

2 Theoretical Background

2.1 Financial Theory

In the financial field, it must be clarified which are the basic financial statements and the accounts that are within the financial statements. The following are defined as basic financial statements: balance sheet, income statement, statement of changes in equity, changes in financial situation, and cash flow [7].

In this work, only the balance sheet, the income statement, and the cash flow are worked on. This is because the financial ratios are calculated exclusively with the accounts that are within these financial statements.

2.1.1. Basic Financial Statements

The three basic financial statements are described as follows:

- “The balance sheet is the accounting document that reports on a specific date on the financial situation of a company, where the obligations, capital, properties in monetary value, and rights are presented”.
- “The income statement or profit and loss statement is defined as the document that provides detailed information on where the profit or loss of the accounting year is obtained.”
- “Cash flow is the statement that shows the movement of income, expenses and the availability of funds on a given date”.

2.1.2. Financial Ratios

The financial ratios come directly from the three basic financial statements, which represent a general overview of the company's situation, which means that you can consult the information of the entire company (financially speaking) in this executive summary (the three basic financial statements).

At the same time, these statements are those used by financial experts to generate some type of evaluation, either directly or with some financial elements involved, derived from cash flow, income statement, and balance sheet. In Table 1, Examples of financial ratios are shown [7].

The financial ratios are important to define a bankruptcy; because these can be compared in percentages or small values, which do not vary concerning the size of the company but vary in its real financial performance, in the defined time.

Stasko et al. [29], mention that there is a formula or indicator to predict bankruptcy, called Altman's Z-score. According to Vera [34], this model provides an accuracy of between 80% and 90% to know if a company is bankrupt or not.

2.2 Computational Models

As for the computational part, Neural Networks (ANN) [8, 16], Convolutional Neural Networks (CNN) [12, 41], Adaptive Neuro-Fuzzy Inference Systems (ANFIS) [16,17, 19], Artificial, Support Vector Machines (SVM) [36, 38, 26], and K-Nearest Neighbors (KNN) [6, 33] are implemented for the Prediction of Enterprise Financial Health. For optimization, Genetic Algorithms (GA) [40, 14] and Particle Swarm Optimization (PSO) [40, 9, 15] are used.

3 Methodology

The methodology used is focused on finding the best combination of parameters that made the machine learning algorithms work optimally, implemented on the same database.

3.1 Database

The database of Taiwan is used to test all the models, which contain 6,819 companies, labeled with 1 the 6,599 financially stable companies, and with 0 the 220 bankrupt companies, with 95 financial ratios (described in Table 16) [20]. These 95 financial ratios are used as attributes for inputs to train and test the machine learning models.

The data on these companies are from the Taiwan economic journal, from 1999 to 2009. This database is considered highly unbalanced because the bankrupt companies barely represent 3% of the total, which requires a dimension reduction or augmentation to balance the database [35].

3.2 Data Pre-Processing

Scaling and standardization of the data are performed to improve the performance of the algorithm using the Standard Scaler, which means putting the data with a mean of 0 and variance of 1. The variables are on different scales and by applying the standard scaler technique they remain on a similar scale.

3.2.1. Database Balance

A data balance is generated in the database so that it had the same number of bankrupt and stable companies. This means that an algorithm is used to increase the size of the database and thereby generate more information.

In this way, there are 6,599 bankrupt companies and 6,599 stable companies. For this purpose, the SMOTE technique is used [4]. This is used to increase the performance of the algorithms so they can identify patterns that are necessary to classify companies.

Table 2. Parameters of the chromosome for GA

Algorithm	Parameters
Support Vector Machine (SVM)	Kernel: RBF, Polynomial
	C: 2^-5 – 2^15
	Gamma: 2^-15 – 2^3
	Polynomial dimension: 3 - 6
	Chromosome size: 4
	Crossing probability: 70%
	Mutation probability: 25%
K-Nearest Neighbour (KNN)	Weights: Uniforms and distance.
	Number of neighbors: 1-10
	P: 1-2
	Chromosome size: 3
	Crossing probability: 70%
	Mutation probability: 33%
Artificial Neural Network (ANN)	Activation function: Relu, Linear, Sigmoidal, Tanh, Prelu, Selu, Elu.
	Learning rate: 0.01, 0.001, 0.0001, 0.00001.
	Optimizer: Adam, RMS, SGD, Adadelta, Adagrad, Adamax, Nadam, Ftrl.
	Number of layers: 1, 2, 3
	Number of neurons per layer: 1-99
	Chromosome size: 10
	Crossing probability: 70%
	Mutation probability: 10%
	Activation function: Relu, Linear, Sigmoidal, Tanh, Prelu, Selu, Elu.

Specifically, in this case, there are few bankrupt companies, so it is more difficult for the model to detect patterns of bankrupt companies in the training phase. Therefore, increasing the number of samples helps to train better.

3.2.2. Dimension Reduction

Within the parameters to manipulate, dimension reduction is used. This is often used to generate better performances for classification algorithms. In this particular case, the PCA algorithm is used. This algorithm is focused on the variance of the components of the database. Values are assigned according to the level of variance and the number of values to be searched is chosen.

Each algorithm is tested with different dimensions. Particularly for Artificial Neural Networks, KNN, and SVM, the dimension is reduced to 93, which is the dimension that generated the best results in the experimentation. It is a reduction of 2 dimensions, which caused some noise in the data. In ANFIS, the best result is provided by PCA with a value of 30.

3.2.3. Data Separation

The data from the database are separated into training and testing sets. The separation is 80% for training and 20% for testing. This number is used to avoid underfitting.

3.3 Classification Algorithms

The classification algorithms are: Artificial Neural Networks, KNN, SVM, and ANFIS. The metric to evaluate them is accuracy. Each of the methodologies of these algorithms is detailed below.

3.3.1. Support Vector Machine (SVM)

The parameters used in the SVM model are the RBF kernel. This kernel requires two parameters, C and Gamma. For this case, C=1000 and gamma =0.01 are used. These parameters are determined, after experimenting with different values of C and Gamma; this is the best architecture found.

3.3.2. K-Nearest Neighbors (KNN)

The parameters used in the KNN model are 1 in the number of neighbor and, “uniform” for the weights.

3.3.3. Artificial Neural Networks (ANN)

In the Artificial Neural Networks models, two options are used: convolutional and multilayer Neural Networks. The parameters used in the convolutional networks are: A 1-dimensional convolution layer, with 32 filters of size 8 and a relu-type activation function.

Another Max pooling layer with a pool size of 2 and stride none. A layer of flattening is also added. After this, a fully connected layer is defined, with 93 neurons and a RELU activation function.

Finally, an output layer with 2 neurons and RELU activation. We added 93 neurons in the fully connected layer since that is the number of ratios used. The loss function used is binary cross-entropy, since there are only two output classes, and the optimizer used is “Adam”. The model is trained with 100 epochs and a batch size of 16.

As for the multilayer neural network, the architecture is as follows: an input layer of 100 neurons, and a hidden layer of 100 neurons, both with RELU activation. Finally, a 1-neuron output layer with sigmoidal activation. The loss function used is binary cross-entropy, given that there are only 2 possible outputs and the Adam optimizer. It is trained with 100 epochs, and a batch of 10 is used.

3.2.4. Adaptive Neural Fuzzy Inference System (ANFIS)

The following parameters are used in the ANFIS model. In GENFIS, the partition is called a grid, which generates the greatest number of possible combinations of membership functions. Gaussian membership functions are used, which are best adapted to the type of data used. 40 epochs are used for training.

It is configured with the hybrid mean square error and backpropagation model. In GENFIS, a change is made to the generation of rules, to generate less and make the model faster and more efficient.

The rules are set equal to the number of membership functions, rather than the dot product of them.

3.4 Cross Validation and Confusion Matrix

A cross-validation of 10 is applied to generate better confidence in the algorithms, and prevent them from being the results of the arrangement of some data. Confusion matrix is used to validate the

Table 3. Results of MNN, CNN, KNN, SVM and ANFIS in accuracy values

Algorithms	Accuracy
Multi-Layer Neural Network (MNN)	98.86%
Convolutional Neural Network (CNN)	99.28%
K-Nearest Neighbors (KNN)	97%
Support Vector Machine (SVM)	98.40%
Adaptive Neural Fuzzy Inference System (ANFIS)	81.82%

Table 4. Confusion matrix of MNN, CNN, KNN, SVM and ANFIS (TP: True Positives, TN: True Negatives, FP: False Positives, FN: False Negatives)

Algorithms	TP	TN	FP	FN
MNN	1,304	1,306	30	0
CNN	1,315	1,306	19	0
KNN	1,245	1,304	89	2
SVM	1,294	1,304	40	2
ANFIS	1582	1658	415	297

efficiency of the algorithms in bankrupt and stable companies' classifications.

3.5 Optimization of Classification Algorithms

Genetic Algorithms and PSO are used to optimize the classification algorithms. Only the optimization is applied to Artificial Neural Networks (ANN), SVM, and KNN, since they are the models that generated the best predictions by more than 12% difference, concerning ANFIS and convolutional networks [10, 11].

3.5.1. Genetic Algorithms (GA)

The parameters used for optimizing the Multilayer Neural Networks architecture and KNN with genetic algorithm are: 100 generations, 200 individuals, one-point crossover, tournament selection method with size 6, real-type chromosome, and the fitness function to maximize the accuracy. In the case of SVM, it is limited to 30 generations, due to the computational capacity required by the algorithm. Table 2 shows the parameter ranges for each real-type gene that makes up each individual or chromosome.

3.5.2. Particle Swarm Optimization (PSO)

The hyper-parameters and parameters of the chromosomes used for optimizing the KNN, SVM, and ANN are the same as the ones implemented in GA.

4 Simulation Results

The results are obtained in two stages. The first stage is achieved through experimentation and knowledge of the literature. In the second stage, the optimization of the models is performed. In both cases, accuracy, F1-score, confusion matrix, and precision metrics are used to verify the effectiveness of each model.

The results are validated using K-fold, to avoid having a performance resulting from the memorization of the algorithms or randomness of the data. Finally, a statistical validation is performed to determine if the results are significantly different. Additionally, the optimizations, the architectures and accuracy obtained in each experiment are shown.

4.1 Results with Experimentation

Table 3 shows the results obtained in terms of accuracy of the four algorithms to be compared.

It is observed that the best result is obtained by the convolutional networks algorithm (CNN). For just under half a percentage point, the multi-layer Neural Networks (MNN) is in second place.

In general, all algorithms exceeded 97%, except the ANFIS algorithm, which is positioned almost 16 points below the KNN algorithm. It can be determined that there is a high efficiency in all algorithms.

Table 4 presents results obtained from the confusion matrix; it can be seen that except for ANFIS, all the algorithms have few false negatives. This means that algorithms rarely make a mistake when classifying a company as "bankrupt".

Most of the errors are found in false positives, which means that there are some companies classified as stable, which are bankrupt.

Table 5. Precision of MNN, CNN, KNN, SVM and ANFIS

Algorithms	Accuracy
MNN	99%
CNN	99%
KNN	97%
SVM	98%
ANFIS	82%

Table 6. F1 score of MNN, CNN, KNN, SVM and ANFIS

Algorithms	Accuracy
MNN	99%
CNN	99%
KNN	97%
SVM	98%
ANFIS	82%

Table 7. K-fold of 10 for MNN, CNN, KNN, SVM and ANFIS

K-Fold	MNN	CNN	KNN	SVM	ANFIS
1	98.94	50	93.86	93.25	55.83
2	99.17	53.26	96.59	94.86	50.45
3	98.79	88.56	95	93.10	38.18
4	99.09	50	93.56	91.49	43.33
5	99.55	84.09	97.57	95.16	48.79
6	99.47	71.59	98.33	96.48	52.58
7	99.09	56.36	97.04	96.48	40.68
8	99.39	67.27	97.50	95.60	39.02
9	98.86	50.04	98.18	96.62	39.31
10	99.47	60.65	97.19	95.59	37.45

In Table 4, the results of the Confusion Matrix are consistent with the accuracy and in the case of CNN and ANN, they only show failures in false positives, this means bankrupt companies are detected but they are not in bankruptcy.

Normally in databases for predicting financial bankruptcies (for example, those of related work), there are many false positives and false negatives, since the normal thing in these databases is that they are highly unbalanced, so the result of 30 and 19 false positives for CNN and ANN respectively, can be interpreted as a good result. We should also not forget to mention that the false negatives in the case of CNN and ANN are 0.

Regarding KNN, it can be seen that there are 89 false positives and 2 false negatives. Regarding SVM, 40 and 2 are obtained respectively. This means that both are quite competent for the type of data that is handled. The ANFIS algorithm, on the other hand, has many false positives (415) and false negatives (297), so it is another verification that it is not the most suitable for this case.

Tables 6 and 7 show the values obtained for precision and f1 score. The results are consistent and practically the same as those previously obtained with accuracy. This is because precision is calculated by dividing true positives by multiplying true positives and false positives. It can be seen that there are very few false positives in all algorithms except ANFIS (approximately 2 to 6%). In the case of F1 score, it is also based on false positives and false negatives, which are very low, so the results do not vary significantly.

The K-fold cross-validation technique is used to validate the algorithms, it is necessary to obtain accurate results and not products of random situations.

Table 7 shows the results obtained by K-fold of 10. In general, it can be observed that, except in the case of MNN, all the algorithms decreased the accuracy score concerning the results in Table 3. However, in the case of CNN and ANFIS, extremely fluctuating and low results are shown, compared to Table 3.

Given these results, it can be seen that neither of the two algorithms managed to have a result similar to that of the test without k-fold. As for MNN, there are results even greater than those obtained in the experimentation phase without k-fold.

For SVM and KNN, there are results very close to those obtained in the previous phase. Table 8 shows the average k-fold of 10. This can be interpreted as the true efficiency of the algorithms since they are tested at different data arrangements, where enough experiments are generated to determine the real effectiveness of the algorithms. It can be seen that MNN increased its effectiveness from 98.86% to 99.18%.

Table 8. Average of K-fold of 10 for MNN, CNN, KNN, SVM and ANFIS

MNN	CNN	KNN	SVM	ANFIS
99.18	63.18	96.48	94.86	44.56

Table 9. Results of Friedman test

Sums	ANN	SVM	KNN
-	3	1	2
-	3	1	2
-	3	1	2
-	3	1	2
-	3	1	2
-	3	1	2
-	3	1	2
-	3	1	2
-	3	1	2
-	3	1	2
Sum	**30**	**10**	**20**
Sum Square	**900**	**100**	**400**

Table 10. Wilcoxon rank results

Average	ANN	SVM	KNN
-	23	3	5
-	26	6	13
-	21	2	7
-	24.5	1	4
-	30	8	18
-	28.5	11.5	20
-	24.5	11.5	15
-	27	10	17
-	22	14	19
-	28.5	9	16
Average	**25.5**	**7.6**	**13.4**

This means that it is an efficient algorithm and it has been the one that has given the best results in terms of accuracy and adapted best to changes, so, it is not a product of memorization or over-training. In the case of SVM, a reduction of almost 4 points can be seen, where it continues to remain above 90%. However, after the K-Fold, at this stage, it can be determined that its ranking has been lower than KNN, which lost half a percentage point.

Finally, it should be mentioned that ANFIS presented a very low result, where it does not even reach 50% after the k-fold, therefore, it can be seen that it only worked with one data arrangement and that it was probably memorizing the data that are given to him, so when it was subjected to this technique, his performance dropped greatly.

In the case of CNN, it became the second worst algorithm and the explanation is that by moving the data, CNN was not able to learn efficiently. Probably due to the filtering and pooling techniques, they generated a complicated model to work with, without forgetting that it is an algorithm generally designed to work with image bits with binary numbers and not with the type of information that financial ratios have.

4.2 Statistical Verification of the Results with Experimentation

The Friedman test with the Chi-square statistic is used to statistically verify which algorithm is the best. For this test, a minimum of 3 variables are required. The test is performed with the 3 best algorithms and their respective K-Fold analyses of 10 entries, ANN, KNN and SVM.

This is because the difference between the other two algorithms is too wide and would only introduce noise into this test. The null hypothesis is that the mean of each population is the same and the alternative hypothesis is that at least one is different. The specific result is shown in Table 9:

Therefore, N is equivalent to 10, K equals 3, Q equals 20, and the value of P is 0.000045. Since the P value is less than 0.05, the null hypothesis is rejected.

It can be determined that they are significantly different and that the neural network is significantly better. A Nemenyi test is also performed. The values in the Wilcoxon range are shown in

Table 11. Results of the optimization with GA for ANN, SVM and KNN in accuracy values

Experiment	ANN	SVM	KNN
1	99.50	99.16	97.46
2	99.43	98.90	97.42
3	99.43	99.24	97.42
4	99.39	99.20	97.46
5	99.39	99.20	97.46
6	99.18	99.31	97.42
7	99.30	99.24	97.46
8	99.43	99.20	97.50
9	99.39	99.24	97.42
10	99.43	99.16	97.38
Average	**99.39%**	**99.19%**	**97.44%**

Table 12. Results of the optimization with PSO for ANN, SVM and KNN in accuracy values

Experiment	ANN	SVM	KNN
1	99.50	99.01	97.15
2	99.39	98.90	97.12
3	99.35	98.82	97.08
4	99.70	98.82	97.19
5	99.50	98.90	97.19
6	99.35	98.86	97.19
7	99.50	99.01	97.12
8	99.46	98.75	97.19
9	99.58	98.82	97.12
10	99.39	98.82	97.12
Average	**99.47%**	**98.87%**	**97.15%**

Table 10. The critical value of the Nemenyi test for alpha of 0.05, infinite N, and K of 3 is 9.22612. When calculating the difference in means of the Wilcoxon values, the following is determined:

- The mean difference between ANN-SVM is 17.90 > 9.22612, therefore there is a significant difference.
- The mean difference between ANN-KNN is 12.10 > 9.22612, therefore there is a significant difference.
- The mean difference between KNN-SVM is 5.80 < 9.22612, therefore there is no significant difference.

4.3 Results for Optimization with GA and PSO

ANN, SVM, and KNN are optimized with GA and PSO. The results are reported below. The Table 11 shows the results obtained through GA. In Table 11, it can be seen that SVM and ANN maintained results greater than 99%. There is, therefore, a successful search for parameters, since in all cases the results were improved to those obtained in the experimentation phase shown in Table 3.

It can also be observed that there was an increase in KNN, that is, it remained in a very similar range all the time. It must be remembered that there are very few parameters in this algorithm, so it is consistent that they remain in similar ranges. The Table 12 shows the results obtained through PSO. The best architecture of each of the algorithms was the following:

- ANN: accuracy of 99.50%, first hidden layer with 94 neurons and activation function relu, first hidden layer with 1 neuron and activation function RELU, Adam optimizer, learning rate of 0.01, and output layer with 1 neuron and function sigmoidal activation.
- SVM: accuracy 99.31%, C=64, gamma of 0.125, and RBF kernel.
- KNN: accuracy: 97.50%, 2 neighbors, uniform weight and P=1.

In Table 12, it can be seen that ANN maintained results greater than 99%. There are also 2 results greater than 99% in SVM. In the case of KNN, there are very small improvements, concerning the experimentation phase, but these results are higher. Therefore, there is a successful search for parameters, since in all cases it was possible to improve the results to those obtained in the experimentation phase shown in Table 3. The best

Table 13. K-fold of 10 for ANN-PSO, SVM-GA, and KNN-GA

K-Fold	ANN-PSO	SVM-GA	KNN-GA
1	98.56	97.50	94.92
2	98.56	98.56	97.34
3	98.81	98.63	95.53
4	98.71	97.04	97.72
5	98.85	98.86	98.40
6	98.99	99.69	93.56
7	98.92	99.31	98.03
8	98.96	99.16	97.72
9	99.01	99.54	98.48
10	98.95	99.46	98.18
Average	**98.83%**	**98.78%**	**96.99%**

Table 14. Results of Friedman test for optimized algorithms

Sums	ANN	SVM	KNN
Sum	23.5	25.5	11
Sum Square	552.25	650.25	121

Table 15. Wilcoxon rank results for optimized algorithms

Average	ANN	SVM	KNN
Average	19.70	20	6.8

architecture of each of the algorithms is the following:

- ANN: accuracy of 99.70%, a first hidden layer with 100 neurons and RELU activation function, a second hidden layer with 100 neurons and Tanh activation function, a third hidden layer with 1 neuron and RELU activation function, Adam optimizer, learning rate of 0.001 and output layer with 1 neuron and sigmoidal activation function.

- SVM: accuracy 99.01%, C=4096, gamma of 0.0125, and RBF kernel.

- KNN: accuracy 97.19%, 2 neighbors, uniform weight and p=2.

The 10-fold cross-validation is used for the best-optimized algorithms of ANN, SVM, and KNN, these being ANN-PSO, SVM-GA, and KNN-GA. Table 13, shows the results of each fold.

The ANN is placed within a range of difference in the results of half a percentage point; While the KNN has up to 5 points of difference; Regarding the SVM classifier, a maximum of 99.46 and a minimum of 97.04 are also observed, almost 2 and a half points difference.

Improvements are obtained in KNN and SVM concerning the average obtained without optimization, so both algorithms achieved the objective. In the case of ANN, the result is lower, therefore, it cannot be determined that a better architecture was obtained.

4.4 Statistical Verification of the Results for Optimization with GA and PSO

The Friedman and Nemenyi test are used to statistically verify which algorithm is the best, with the same values and parameters used in the subsection 5.2. The specific result for the Friedman test is shown in Table 14:

Therefore, N is equivalent to 10, K equals 3, Q equals 12.35, and the value of P is 0.002080. Since the P value is less than 0.05, the null hypothesis is rejected. It can be determined that they are significantly different and that the neural network is significantly better.

For the Nemenyi test, the values in the Wilcoxon range are shown in Table 15.

The critical value of the Nemenyi test for alpha of 0.05, infinite N, and K of 3 is 9.22612. When calculating the difference in means of the Wilcoxon values, the following is determined:

- The mean difference between ANN-KNN is 12.90 > 9.22612, therefore there is a significant difference.
- The mean difference between ANN-SVM is 0.30 > 9.22612, therefore there is no significant difference.
- The mean difference between KNN-SVM is 13.90 < 9.22612, therefore there is a significant difference.

5 Discussion

In terms of accuracy (Table 3), the best algorithm is CNN, with 99.28%, followed by ANN with 98.86%. It can also be seen that SVM and KNN had outstanding results since they obtained an accuracy of 98.28 and 97%, respectively. In general, both models can be used due to their accuracy above 97%, which makes them reliable.

On the other hand, ANFIS is placed at 81.82%, so for the moment it can be ruled out as a good algorithm for this database, given the tested architectures.

These data do not yet demonstrate statistical differences; a K-fold of 10 was required to generate statistical tests.

Regarding the precision parameter (Table 5) and the F1 score (Table 6), they are added to analyze if there is any important difference; however, the results are the same as discussed above, there are no variations in comparison with accuracy. This is because both measures are calculated through false positives and false negatives.

K-Fold cross-validation is used to validate the previous data, with size 10, as shown in Table 7. In this case, there are important variations.

First of all, the CNNs in this test did not obtain results higher than 89%. Likewise, it can be seen that the results are very fluctuating and are between 50% and 88.56%. This generated an average of 63.18%.

This variation can be explained because the weights in the CNNs are random; this means that perhaps, given some weights and given the training and testing data structure, the CNN can memorize the data.

Another algorithm that behaved extremely inefficiently is ANFIS, which when applying the K-fold, its performance dropped to 44.56% on average. It may be due to memorization as in the previous case.

Also, some membership functions could have been better adapted to the data of the first test, but in the consecutive tests, the algorithm is no longer as efficient.

In SVM there is also a significant reduction, of just over 3 percentage points. In this case, it is placed at an average of 94.86%, so it is possible to determine that it is still a good algorithm, with a good architecture for predicting financial bankruptcies, given this database.

In KNN there is also a slight reduction, of half a percentage point. In this case, the reduction is minimal, so it can be determined that given the architecture used and given this database, it is an extremely efficient algorithm to predict a financial bankruptcy.

If a final product is developed that requires little computational capacity, this algorithm would be the best option, since it consumes very few computational resources and generates an efficient result.

Regarding the ANN, there are improves, as seen in Table 8, on average it is achieved at 99.18%; This means, in the k-fold test the ANN showed the best average of prediction of financial bankruptcy.

The Friedman test is used for the three best algorithms to generate a statistical verification, with the results obtained by the K-fold validation.

It can be seen in the previous section that a result of P of 0.000045 is generated, which means that the null hypothesis is rejected since it is less than 0.05. This hypothesis says that the population average is the same in the three algorithms, therefore, at least one is different.

Regarding the Nemenyi test, it can be determined that significant differences occur between ANN, concerning SVM and KNN with a 5% significance, given that it is less than 0.05.

In the optimizations phase, ANN, SVM, and KNN are used, since the difference between CNN and ANFIS is great and the execution time would be much higher. Regarding ANNs, the best result obtained in GA is 99.50% in terms of accuracy.

Likewise, in SVM 99.20% is obtained and in KNN 97.46%. Comparatively to the results of the first test, an increase of almost 1 percentage point can be seen in the ANN and in SVM, as well as half a percentage point in KNN.

In PSO, the best value in terms of accuracy is obtained, since 99.70% is achieved in the ANN, 99.01% in SVM, and 97.15% in KNN.

Table 16. Financial ratios from the database of Liang and Tsai [20]

Return on total assets before taxes	**Return on total assets after taxes**	**Return on assets before interest and depreciation, after taxes**
Gross operating margin	Gross sales margin	Operating profit rate
Net interest rate before taxes	Net interest rate after taxes	Non-operating net income ratio.
Continuous interest rate	Operating Expense	Rate Research and development expense rate
Cash flow rate	Interest rate on interest-bearing debt	Effective tax rate
Net value per share (A)	Net value per share (B)	Value per share (C)
Earnings per share for the last four seasons	Cash flow per share	Earnings per share
Operating profit per share	Net earnings per share before taxes	Sales gross profit growth rate
Operating profit growth rate	Net profit growth rate after tax	Regular net profit growth rate
Continuous growth rate of net income	Total assets growth rate	Net worth growth rate
Total Asset Return Growth Rate Ratio	Cash Reinvestment Percentage	Current Radius
Acid Test	Interest Expense Ratio	Total Debt/Net Worth
Debt Ratio	Net Worth/Assets	Long-Term Fund Suitability Index
Debt dependence	Contingent liabilities / net worth	Operating profit / paid-in capital
Net income before taxes / paid-in capital	Inventory and accounts receivable	Total asset turnover
Accounts Receivable Turnover	Average Receivable Days	Inventory Turnover Rate
Frequency of fixed asset turnover	Net worth turnover rate	Income per person
Operating profit per person	Allocation rate per person	Working capital to total assets
Quick assets / total assets	Current assets / total assets	Cash / total assets
Quick assets / current liabilities	Cash / current liabilities	Current liabilities with assets
Operating funds to liabilities	Inventory/working capital	Inventory / current liabilities
Current liabilities/liabilities	Working capital/equity	Current liabilities/equity
Long-term liabilities with current assets	Total income / total expenses	Expenses / total assets
Current asset turnover rate	Rapid asset turnover rate	Working capital turnover rate
Cash turnover ratio	Cash flow to sales	Fixed assets to assets
Current liability to liability	Current liability to equity	Equity to long-term liability
Cash flow to total assets	Cash flow to liabilities	Cash flow from operations to assets
Cash flow to equity	Current liabilities with current assets	Liability-asset mark
Net income to total assets	Total assets to price	Total assets to price of gross domestic product
Interval without credit	Gross profit on sales	Net profit from stockholders' equity
Liabilities versus equity	Degree of financial leverage	Interest coverage ratio
Net income indicator	Shareholders' equity to liabilities	

The results in this case are consistent. In the optimizations, the execution times are extremely high. The ANNs took about 1 week per run, of which, 10 are done for PSO and 10 for GA. Likewise, in SVM, the generation reduction had to be done, since some polynomial kernel architectures could not complete the execution, due to the level of complexity and the penalty that is given.

The KNN algorithm is the fastest in this aspect since it could have executions of 6 hours or less, depending on the machine where the algorithm is run. Finally, a K-Fold is made of the best-optimized results, where the surprise is that the ANN architecture is not better than the architecture obtained in the empirical experimentation phase.

This, like the result of the CNNs, could be because the model in terms of accuracy managed to obtain a superior result through memorization, which in the end, if these data are moved, it does not generate such efficient results.

Either way, the result is 98.83%, which generates a good and usable architecture. In KNN there is an improvement of 0.3% and in SVM 3.92%. SVM had the greatest increase in optimizations and helped generate a model that is almost on par with the ANN. This means that validations or tests can be done with both algorithms, whenever greater certainty is required.

Regarding statistical validation, the Friedman test is also used with the K-fold of 10, where a P value less than 0.05 is obtained, which rules out the null hypothesis. This means that the differences are statistically significant.

In the Nemenyi test, it can be observed that significant statistical differences occurred between ANN and KNN. There is also a significant difference between KNN and SVM. In both cases, Nemenyi values lower than 0.05 are generated, which is the level of significance. Regarding ANN and SVM, it can be stated that they do not have a significant difference, since their value was higher than the significance level of 0.05.

Given these statistical results, it can be determined that the ANN shows the best results, with a significant statistical difference. Finally, comparing the values obtained, the best result in accuracy terms is obtained by the optimized ANN-PSO, with an accuracy of 99.70%, followed by the optimized SVM-GA value of 99.24%, and finally the optimized KNN-GA value of 97.50%; the best result in accuracy average with k-fold of 10 is obtained by the ANN of empirical analysis with 99.18% value, followed by the SVM-GA value of 98.78%, and the KNN-GA value of 96.99%.

6 Conclusions

Based on the analysis of the results, it can be concluded that the model that has the best performance in predicting financial bankruptcy in the database of Taiwanese companies is the ANN, with 99.18% accuracy values and validating their performance using cross validation K-Fold.

This model has a high computational cost since it is a network with a very high number of connections, so its use would require a little more resource than less precise models. Naturally, it has the advantage of having greater precision, when doing a deep analysis, than other models.

Analyzing in terms of computational cost, execution time, and accuracy of the algorithm, it could be determined that KNN (96.73%) is a good option. KNN is an algorithm that is used on a large scale due to its low computational cost and low complexity, so it would have that advantage on its side.

However, the accuracy shown by ANN would be sacrificed. It should be noted that the computer where the optimization algorithms are executed is not a high-processing one, as it has a 2.90 GHz Intel I7 processor, 16 GB of RAM and a graphic card of 2 GB.

The greatest justification for this work is to generate a functional model that provides greater certainty to the investments of businessmen, governments, and people in general.

With the results provided by these computational models, one can be certain that this objective is covered, in Taiwanese companies that are listed on the stock exchange and that have basic financial statements, with 99.18% effectiveness.

It can also be concluded that the general average of 80% reliability of the “Altman” financial prediction tool was exceeded.

The optimizations generated better results in KNN and SVM with little difference, but no improvement in ANN. The time taken in the executions must be evaluated with the results obtained, as well as the carbon footprint generated by having a computer on for a long time.

Also, in this area, it should be considered for future research, not to base the optimizations on accuracy values with a single execution, but rather to base them on K-fold directly, so as not to have inefficient results in the end as happened with the ANN.

It must also be considered that the experiment could last 10 times longer or many more computational resources, so it must be assessed how necessary this implementation would be, given the results already obtained. In general, if this research is compared with those generated by the authors of the related work, it can be concluded that competent results were achieved comparatively to the best results of each work, even better in most cases.

However, it must be mentioned that they have not been compared with the same databases, so this comparison, for the moment, would only be in terms of accuracy, which is incomplete and would require, in the future, comparison with the same database of each author to be able to define it.

It can also be concluded that, with only the financial ratios of a company, its financial bankruptcy can be predicted. Particularly, with the 95 financial ratios used in this work. In future work, other models for predicting financial bankruptcy will be considered, like LSTM network, recurrent network, variants of neuro-fuzzy networks, etc.

References

1. **Abdou, H., Abdallah, W., Mulkeen, J., Ntim, C. G., Wang, Y. (2017).** Prediction of financial strength ratings using machine learning and conventional techniques. Investment Management and Financial Innovations, Vol. 14, No. 4, pp. 194–211.
2. **Arieshanti, I., Purwananto, Y., Ramadhani, A., Nuha, M. U., Ulinnuha, N. (2013).** Comparative study of bankruptcy prediction models. TELKOMNIKA (Telecommunication Computing Electronics and Control), Vol. 11, No. 3, pp. 591. DOI: 10.12928/telkomnika.v11i3.1143.
3. **Arora, N. Saini, J. R. (2013).** Time series model for bankruptcy prediction via adaptive neuro-fuzzy inference system. International Journal of Hybrid Information Technology, Vol. 6, No. 2, pp. 51–64.
4. **Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002).** Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, Vol. 16, pp. 321–357. DOI: 10.1613/jair.953.
5. **Chen, Y., Liu, K., Xie, Y., Hu, M. (2020).** Financial trading strategy system based on machine learning. Mathematical Problems in Engineering, Vol. 2020, pp. 1–13. DOI: 10.11 55/2020/3589198.
6. **Cunningham, P., Delany, S. J. (2021).** K-nearest neighbour classifiers - A tutorial. ACM Computing Surveys, Vol. 54, No. 6, pp. 1–25. DOI: 10.1145/3459665.
7. **Dantine, J. P., Donaldson, J. B. (2014).** Intermediate financial theory. Third edition, Academic Press.
8. **Ekman, M. (2021).** Learning deep learning: Theory and practice of neural networks computer vision, natural language processing, and transformers using tensorflow. Addison-Wesley Professional.
9. **Gad, A. G. (2022).** Particle swarm optimization algorithm and its applications: A systematic review. Archives of Computational Methods in Engineering, Vol. 29, No. 5, pp. 2531–2561. DOI: 10.1007/s11831-021-09694-4.
10. **Gaxiola, F., Melin, P., Valdez, F., Castro, J. R. (2017).** Optimization of deep neural network for recognition with human iris biometric measure. Advances in Intelligent Systems and Computing, pp. 172–180. DOI: 10.1007/978-3-319-67137-6_19.
11. **Gaxiola, F., Melin, P., Valdez, F., Castro, J. R. (2016).** Optimization of type-2 and type-1 fuzzy integrator to ensemble neural network with fuzzy weights adjustment. Studies in Computational Intelligence, pp. 39–61. DOI: 10.1007/978-3-319-47054-2_3.
12. **Goodfellow, I., Bengio, Y., Courville, A. (2016).** Convolutional networks. Lecture slides for Chapter 9 of Deep Learning.

13. **Guajardo, G., Andrade, N. (2012).** Estados financieros. Contabilidad para no contadores.
14. **Hamdia, K. M., Zhuang, X., Rabczuk, T. (2020).** An efficient optimization approach for designing machine learning models based on genetic algorithm. Neural Computing and Applications, Vol. 33, No. 6, pp. 1923–1933. DOI: 10.1007/s00521-020-05035-x.
15. **Houssein, E. H., Gad, A. G., Hussain, K., Suganthan, P. N. (2021).** Major advances in particle swarm optimization: theory, analysis, and application. Swarm and Evolutionary Computation, Vol. 63, pp. 100868. DOI: 10.1016/j.swevo.2021.100868.
16. **Jang, J. S. R., Sun, C. T., Mizutani, E. (1997).** Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence. IEEE Transactions on automatic control, Vol. 42, No. 10, pp. 1482-1484.
17. **Jang, J. (1993).** ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 665–685. DOI: 10.1109/21.256541.
18. **Kansal, A. K., Sharma, S. (2019).** A methodological review of financial distress prediction techniques. 4th International Conference on Advances in Management & Digital Sciences, Vol. 1, pp. 294–301.
19. **Karaboga, D., Kaya, E. (2019).** Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey. Artificial Intelligence Review, Vol. 52, No. 4, pp. 2263–2293. DOI: 10.1007/s10462-017-9610-2.
20. **Liang, D., Tsai, C. F. (2020).** Taiwanese bankruptcy prediction. UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/Taiwanese+Bankruptcy+Prediction.
21. **López, J. F. (2019).** Bankruptcy. Economipedia.com. https://economipedia.com/definiciones/quiebra.html#:~:text=Una%20quiebra%20o%20bancarrota%20es,su%20actividad%20de%20forma%20permanente.
22. **Mselmi, N., Lahiani, A., Hamza, T. (2017).** Financial distress prediction: the case of french small and medium-sized firms. International Review of Financial Analysis, Vol. 50, pp. 67–80. DOI: 10.1016/j.irfa.2017.02.004.
23. **Muslim, M. A., Dasril, Y. (2021).** Company bankruptcy prediction framework based on the most influential features using XGBoost and stacking ensemble learning. International Journal of Electrical and Computer Engineering (IJECE), Vol. 11, No. 6, pp. 5549. DOI: 10.11591/ijece.v11i6.pp5549-5557.
24. **Narvekar, A., Guha, D. (2021).** Bankruptcy prediction using machine learning and an application to the case of the COVID-19 recession. Data Science in Finance and Economics, Vol. 1, No. 2, pp. 180–195. DOI: 10.3934/dsfe.2021010.
25. **Park, M. S., Son, H., Hyun, C., Hwang, H. J. (2021).** Explainability of machine learning models for bankruptcy prediction. IEEE Access, Vol. 9, pp. 124887–124899. DOI: 10.1109/access.2021.3110270.
26. **Pisner, D. A., Schnyer, D. M. (2020).** Support vector machine. Machine Learning, pp. 101–121. DOI: 10.1016/b978-0-12-815739-8.00006-7.
27. **Santoso, N., Wibowo, W. (2018).** Financial distress prediction using linear discriminant analysis and support vector machine. Journal of Physics: Conference Series, Vol. 979, pp. 012089. DOI: 10.1088/1742-6596/979/1/012089.
28. **Shetty, S., Musa, M., Brédart, X. (2022).** Bankruptcy prediction using machine learning techniques. Journal of Risk and Financial Management, Vol. 15, No. 1, pp. 35. DOI: 10.3390/jrfm15010035.
29. **Stasko, A., Birzniece, I., Kebers, G. (2021).** Development of bankruptcy prediction model for Latvian companies. Complex Systems Informatics and Modeling Quarterly, No. 27, pp. 45–59. DOI: 10.7250/csimq.2021-27.02.
30. **Sudarsanam, S. K. (2016).** A fuzzy neural network model for bankruptcy prediction. Journal of Engineering Computers & Applied Sciences (JECAS), Vol. 5, No. 6, pp. 33–40.
31. **Suryawanshi, R., Takavane, T., Shah, T., Mathur, R. (2022).** Stocks and cryptocurrency price prediction using long short term memory.

International Journal of Engineering Research & Technology, Vol. 11, No. 5, pp. 560–564.

32. **Tabbakh, A. (2021).** Bankruptcy prediction using robust machine learning model. Turkish Journal of Computer and Mathematics Education, Vol. 12, No. 10, pp. 3060-3073.
33. **Taunk, K., De, S., Verma, S., Swetapadma, A. (2019).** A brief review of nearest neighbor algorithm for learning and classification. International Conference on Intelligent Computing and Control Systems, pp 1255–1260. DOI: 10.1109/iccs45141.2019.9065747.
34. **Vera-García, I. (2017).** El modelo Z de Altman como herramienta financiera para pronosticar o predecir el desempeño financiero de las empresas mexicanas cotizadas. Caso de las empresas manufactureras del sector alimenticio. Master degree thesis, Universidad Autonoma del Estado de Hidalgo, UAEH Digital Library. http://dgsa.uaeh.edu.mx:8080/bibliotecadigital/handle/231104/2381.
35. **Wang, H., Liu, X. (2021).** Undersampling bankruptcy prediction: Taiwan bankruptcy data. PLOS ONE, Vol. 16, No. 7, pp. e0254030. DOI: 10.1371/journal.pone.0254030.
36. **Wang, L. (2005).** Support vector machines: theory and applications. Singapur: Springer. Vol 177.
37. **Wang, N. (2017).** Bankruptcy prediction using machine learning. Journal of Mathematical Finance, Vol. 07, No. 04, pp. 908–918. DOI: 10.4236/jmf.2017.74049.
38. **Wu, J. (2020).** Essentials of pattern recognition: An accessible approach. Cambridge University Press.
39. **Xie, C., Luo, C., Yu, X. (2010).** Financial distress prediction based on SVM and MDA methods: the case of chinese listed companies. Quality & Quantity, Vol. 45, No. 3, pp. 671–686. DOI: 10.1007/s11135-010-9376- y.
40. **Yang, X. S. (2020).** Nature-inspired optimization algorithms. Academic Press.
41. **Zhang, A., Lipton, Z. C., Li, M., Smola, A. J. (2021).** Dive into deep learning. ArXiv. DOI: 10.48550/ARXIV.2106.11342.

Article received on 31/01/2024; accepted on 12/03/2024.
[*]Corresponding author is Fernando-Gaxiola.