International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I12P110

Volume 72 Issue 12, 107-115, December 2024

Original Article

Credit Rating Models based on Backpropagation Neural Networks, a Peruvian Case

César Canelo[1], Pedro Espinoza[2], Carlos Ponce[3]

[1,2,3]National University of Engineering, Peru.

[1]Corresponding Author : ccanelo@uni.edu.pe

Received: 01 July 2024 Revised: 07 October 2024 Accepted: 13 November 2024 Published: 25 December 2024

Abstract - *Microfinance institutions use various credit rating models to improve their credit risk management efficiency. These models are based on statistical and Artificial Intelligence (AI) techniques. We work with a database from a Peruvian microfinance company that, for security reasons, we only knew the description of the variables. This database contains the records of 15,569 borrowers with 26 variables for the classification of their clients and a final variable (27) for acceptance or rejection of credit. In the statistical part, an exhaustive study of the interdependence of the variables is carried out, discovering that the variable (27) depends on a single decision-making variable (8). The influence of the other variables on the decision-making variable is analyzed. In the (AI) part, various Backpropagation Neural Network (BPNN) structures are tested, taking the complete database and obtaining good precision in the prediction, reaching 97.68%. This high precision of the neural network is explained because the final variable (27) depends directly on the decision variable (8). With the same BPNN structures and taking variable (8) instead of variable (27), the precision obtained by the neural network decreases to 77.40%. Finally, maintaining the variable (27) as the expected value of the network and eliminating the variable (8) from the database, the precision of the neural network drops to 66.89%, which confirms that this variable is the most realistic.*

Keywords - *Microcredit, Backpropagation Neural Networks, Credit Risk, Machine learning in finance, Financial analytics.*

1. Introduction

Microfinance institutions and their credit technologies emerged in the 1980s. In the 1990s, the Rural Savings and Credit Banks (CRAC) were established as financial entities primarily focused on the agricultural sector, filling the gap left by the collapse of the Agrarian Bank, which faced a significant delinquent portfolio. Over time, these institutions began incorporating commercial credit lines and products for Mype (medium and small businesses) into their portfolio. Small and Microenterprise Development Companies (EDPYMES) were also created to specialize in microfinance and support the continuous growth of entities dedicated to this sector. According to Webb et al. [20], in the last decade, an increase in the number of financial institutions providing credit to microenterprises has been observed, leading to greater competition in the sector and a reduction in the average interest rate from 55 percent in 2002 to 32 percent in 2009. Consequently, the number of formal microcredits increased significantly, from 300,000 in 2002 to 2.1 million by 2010. The credit volume also expanded from S/ 1.5 billion to S/ 20.2 billion, representing 4.8 percent of the GDP over the same period, signifying a real growth of more than tenfold. By the end of 2011, the microfinance sector in Peru comprised 40 institutions, including one bank, three finance companies, 13 Municipal Savings and Credit Banks (CMAC), such as the Municipal Popular Credit Bank of Lima, 10 Rural Savings and Credit Banks, and 13 EDPYMEs. The success of the microfinance sector has fostered a favorable investment climate, attracting foreign funds specialized in this credit segment. Quispe, León, and Contreras [17] highlighted that Peru was recognized by The Economist Intelligence Unit as the country with the best business conditions for microfinance worldwide for four consecutive years (2008-2011) due to the significant advancements in microenterprise credit. The microfinance sector in Peru has become more competitive, and the entry of international commercial banks has forced microfinance institutions to be more efficient, specifically in improving credit risk control. For this reason, these entities seek to implement automatic credit scoring systems to reduce the cost of credit analysis, make faster credit decisions, and have better control over credit recovery. Derelioglu and Gürgen [8] present an approach for evaluating credit risk in small to medium-sized enterprises in Turkey, utilizing a Neural Feature Extraction (NRE) strategy focused on a Multilayer Perceptron (MLP). To achieve dimensionality reduction, feature selection is applied using Decision Trees (DT), followed by recursive feature extraction with support vector machines (RFE–SVM), and concluding with feature extraction via Factor Analysis (FA) and Principal Component Analysis (PCA). The reduced dataset is then analyzed using k-

Nearest Neighbors (k-NN), MLP, and SVM classifiers, with performance comparisons made through classification experiments. The dataset consists of 512 records. The non-parametric k-NN classifier makes final decisions for new data points by assessing the nearest class, with k set as an odd number to minimize class confusion. The experiments were performed using WEKA, with k set to 5 and Euclidean distance employed for measuring proximity. The MLP network structure transforms the d-dimensional input space into an h-dimensional space during training, where outputs are produced through nonlinear activations in the hidden units. Each training epoch involves sending weighted input sums to hidden units, which are then processed by a nonlinear activation function. This process repeats until a stopping criterion is met, such as reaching the maximum number of iterations or achieving a minimum error rate.

The SVM seeks to find the optimal hyperplane that maximizes the separation between different classes during classification. To maximise this margin, the margin is defined as the distance between the hyperplane and the nearest data points. The SVM classification decision depends on whether the output value is above 0, indicating a class of 1 for a 'good' client, or below, designating 0 for a 'bad' client. The MLP network is recognized as the most effective model for predicting if customers are 'good' or 'bad.' Khashman [11] employed supervised Backpropagation Neural Networks to evaluate credit risk under diverse learning schemes. The study used a dataset comprising 1,000 records, each with 29 variables.

Three different ANN architectures were trained, varying in the number of neurons in the hidden layer, and each model type was tested with nine distinct learning schemes. The network training process consisted of two main phases: training and validation, each involving different amounts of data. Among the nine schemes evaluated, initial training started with 100 records while validation used 900 records, with increments of 100 records per iteration until training reached 900 records and validation used 100 records. Early training phases involving fewer records exhibited high correlation, which decreased as more data was employed in validation, showing a lower correlation at the other extreme.

With a balanced proportion of data, the highest overall correlation is achieved. However, when this balance shifts towards either an increase or decrease, the correlation decreases significantly for both the training and validation phases and the overall outcome. The best results reported are accuracy rates of 99.17% for training and 66% for validation, with an overall accuracy of 85.9%. Given these suboptimal results, it can be conjectured that the disparity may be attributed to the large number of variables (20) compared to the relatively small dataset size (1,000 records) [11]. Khashman [12] introduces an innovative approach involving Emotional Neural Networks (EmNNs) and compares their effectiveness with conventional Neural Networks (NNs) in the context of credit risk assessment. While traditional NNs utilize the backpropagation learning method, EmNNs implement an emotional backpropagation process. This distinctive approach incorporates two additional neurons representing the emotional factors of anxiety and trust. The study employs an Australian credit dataset comprising 695 records with 14 variables. A total of 12 neural networks were tested, divided into six emotional and six conventional models, and assessed under three distinct learning schemes that modified the data flow between the training and validation phases. The results demonstrate that although both emotional and conventional models are effective for credit risk evaluation, the emotional model outperforms the conventional one in terms of processing speed and accuracy for credit application assessments.

Although Conventional Neural Networks (NNs) have proven effective, they lack human-like emotional components. Emotional NNs incorporate key emotional responses such as anxiety and confidence, which evolve throughout the training phase. Effective learning results in decreased anxiety and increased confidence as training progresses. Research has shown that emotional NNs surpass conventional models in several areas, including maximum training iterations, minimal error rates, decision-making speed in credit evaluations, and overall accuracy. Bekhet and Eleter [2] developed an advanced predictive model using Artificial Neural Networks specifically for Jordanian commercial banks, utilizing a dataset of 492 records and 12 variables from a Jordanian bank. The model implemented a three-layer Backpropagation Neural Network, with the input layer comprising 12 neurons corresponding to the input variables. The hidden layer employed a hyperbolic tangent activation function, while the output layer used a softmax activation function for the dependent variable.

The result of the neural model demonstrated a detection rate of 95% for correctly accepted requests and 89.9% for correctly rejected requests. Blanco, Pino-Mejías, Lara, and Rayo [4] conducted an in-depth study on credit scoring models for Microfinance Institutions (MFIs), employing Multilayer Perceptron (MLP) Neural Networks along with three parametric techniques: Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Logistic Regression (LR). For each of these four techniques, the corresponding mathematical models were presented. One of the primary objectives was to compare the performance of MLP models against the three parametric models. The study utilized a sample of 5,500 borrowers from a Peruvian MFI, incorporating both financial and non-financial variables. A total of 17 credit scoring models were evaluated, 14 of which were MLPs with different structures, and the remaining three were parametric models. The performance comparison was based on areas under the ROC curve (AUC) and misclassification costs. The MLPs were trained using six nonlinear function minimization algorithms: gradient descent, gradient descent with momentum, Quasi-Newton, Levenberg-Marquardt, scaled conjugate gradient, and resilient backpropagation. Additionally, it was noted that beyond these algorithms, an extensive list of methods is designed for this purpose. The MLP14 model, structured with three layers consisting of 20 neurons in the input layer, 3 neurons in the hidden layer, and 1 neuron in the output layer, exhibited the best performance. The training was conducted using the Quasi-Newton algorithm and evaluated with Qsas-N and M26ab R2010b systems. The MLP14 model demonstrated a reduction in misclassification costs by 8.06%, 7.04%, and 13.78% when compared to LDA, QDA, and LR, respectively, signifying that MLP-based scoring models have the potential to mitigate financial losses for Microfinance Institutions (MFIs) significantly. Bekhet and Eleter [3] examined the efficacy of two credit scoring methodologies for Jordanian commercial banks: a Neural Network with a Radial Basis Function (RBF) and a Logistic Regression model.

Their dataset comprised 492 records with 12 variables. Testing with the Logistic Regression model yielded 79.6% correct identification of rejected applications and 88.4% accurate classification of approved applications, culminating in an overall classification rate of 84.8%. Further testing of the RBF network structure, including 12 input neurons, six hidden neurons, and two output neurons, achieved training and testing accuracy of 80.9% and 86.5%, respectively. In terms of accuracy, Logistic Regression (LR) proved superior to the Radial Basis Function (RBF) network in classifying approved applications (88.4% vs. 76.1%). Conversely, when predicting denied applications, RBF outperformed LR (84.2% vs. 78.9%). Malhotra et al. [15] aimed to evaluate the effectiveness of Neural Networks and statistical methods in detecting potential loan defaulters within the context of 12 credit cooperatives. Loan data from these cooperatives were combined to form a dataset of 1,100 records.

Six models were developed: Model 1 utilized Logistic Regression, while Models 2 and 3 employed inverse propagation neural networks with one hidden layer containing 50 neurons and two hidden layers with 35 neurons each, respectively. These networks were trained using the gradient descent algorithm with an adaptive step size. Models 4 and 5 were also inverse propagation neural networks but had one hidden layer with 50 neurons and two hidden layers with 40 and 10 neurons, respectively, and were trained using the Levenberg-Marquardt algorithm. Lastly, Model 6 employed LVQ (Learning Vector Quantization) neural networks. Table 3 presents a comparative analysis of the performance of the six models across both the training and testing phases. For training, 350 good and 350 bad clients were separated using a randomization algorithm, while 200 good and 200 bad clients were reserved for testing. They concluded that neural network models outperform the logistic regression model in identifying both good and bad credits. Among the six models evaluated, Model 3, which featured adaptive learning, demonstrated the highest accuracy in detecting good and bad loans. Zhao, Xu, Kang, Kabir, Liu, and Wasinger [22] presented an improved credit scoring model based on Multilayer Perceptron (MLP) Neural Networks. Their study emphasized three key aspects: (i) optimizing data distribution for the training, validation, and test phases using the average random choice method. They used the German Credit dataset, which includes 1,000 instances—700 accepted credits (A) and 300 rejected credits (B). For the testing phase, 70 instances from A and 30 from B were randomly set aside. From the remaining data, 560 instances from A and 240 from B were selected for training, while 70 from A and 30 from B were assigned for validation. After completing the three phases, the error between the MLP-NN outputs and the actual values was measured. This procedure was repeated 20 times, and the errors were averaged. This test was conducted for different MLP–NN structures. (ii) The study compared the effects of varying the number of instances allocated to training, validation, and testing phases, using the ratios 8:1:1, 9:0.5:0.5, and 6:2:2, concluding that the first ratio provided the best results. (iii) They evaluated 34 MLP–NN models with varying numbers of neurons in the hidden layer, analyzing the training, validation, and test phases. The lowest error rates in the test phase were observed when the hidden layer had 9, 10, or 12 neurons. The authors noted that the proposed model achieved an accuracy rate of 87%, surpassing the 82% accuracy reported in previous studies [22]. Kiruthika and Dilsha [13] compared Logistic Regression and Neural Network models for credit scoring. They used a dataset containing 520 records, with variables related to client information, financial characteristics of current operations, and macroeconomic context. Their findings indicated that there is no definitive parameter or rule for constructing an optimal model for either Neural Network or Logistic Regression, as each has its respective advantages and disadvantages.

The advantage of Neural Networks is their strong learning capacity without assumptions about the relationship between input variables. The Logistic Regression model also can handle non-linear relationships between variables, since it incorporates an exponential term in its function; however, this model requires knowing a priori the form of the non-linear relationship. To compare both models, they used the ROC index and the misclassification rate, the proportion of incorrect classifications for all classes among the total number in the sample. Logistic Regression models have a lower misclassification rate in the training data set; while, in the validation data set, the Neural Network models have the lowest misclassification rate. In all cases, Neural Network models have the highest ROC index. A good model has a relatively stable misclassification rate: a low difference in misclassification rate in training and validation and a higher ROC index. Therefore, it is concluded that the Neural Network is a better model. From the list of reviewed and commented articles, it can be concluded that models based on learning machines, such as neural networks support vector machines, are superior to classical statistical models, such as Logistic Regression and Discriminant Analysis. This study aims to create credit scoring models for microcredit applicant classification, utilizing Self-Organizing Maps (SOM) Networks and Support Vector Machines (SVM). To achieve this, a database comprising records of microfinance borrowers is utilized. The study is organized as follows: Section 2 provides a detailed description of the database and an examination of the variables for non-compliance prediction. Section 3 introduces the proposed models, which include Backpropagation Neural Networks, SOM Networks, and Support Vector Machines. Section 4 presents and discusses the results. Lastly, Section 5 outlines the study's conclusions.

2. Data and Variables

2.1. The Dataset

A dataset from a Peruvian microfinance entity (Caja Municipal de Ahorro y Crédito) was utilized, containing borrower information from 2011–2014. The dataset included the loan's characteristic variables and the borrower's behavioural variables. The Excel database comprised 15,569 rows of records and 27 columns of variables. Column 27 indicated accepted and rejected credits, with 15,256 accepted and 313 rejected. For security reasons, the data was provided as an Excel table and a separate document describing the 27 variables, as shown in Table 1. The dependency relationship between the 26 columns and column 27 was not disclosed.

2.2. Description of Input and Output Variables

Consequently, in the Excel database, there is for each client a row o records with 26 values, which correspond to the input variables (V1, V2, .., V26) and a value that corresponds to the response variable (V27) that indicates the acceptance or rejection of the credit. A statistical study was conducted to discover the interdependence of the 27 variables, summarized in Table 2. The histogram (Table 3) reveals 15,256 clients with accepted credit and 313 rejected. Let the matrix B=[N, V8, V27] have 3 columns, where N are the integers from 1 to 15569, V8 is ordered from smallest to largest, and V27. It is observed in B that up to rows 15,256, the values of V8 vary from 0 to 30, and the values of V27 are all zeros. After this row, the values of V8 grow from 31 to 162, and those of V27 are constant a V8 and V27 are constant and equal to 1.

Table 1. Description of the variables

Variable	Description
V1	Type of currency
V2	Amount of credit granted
V3	Capital debt to date
V4	Type of Credit: 0) Loans to small businesses, 1) Loans to microbusinesses, 2) Revolving consumer loans, 3) non-revolving consumer loans, 4) Mortgage loans for housing.
V5	Classification of the debtor according to the SBS: 0) Normal, 1) Credit with possible loss, 2) Deficient, 3) Doubtful, 4) Loss.
V6	Classification of the debtor according to Microfinance: 0) Normal, 1) Credit with possible loss, 2) Deficient, 3) Doubtful, 4) Loss.
V7	Days late at the end of the month.
V8	Days late to pay the last installment.
V9	Average number of days late in payment of installments in the last 6 months.
V10	Accounting provision for possible loss.
V11	Capital debt in force after 30 days.
V12	Capital debt due after 30 days.
V13	Capital debt in judicial collection of the operation.
V14	Interest not received from credits in judicial collection or overdue.
V15	Interest due not paid by the debtor.
V16	Disbursement date.
V17	Payment scheme.
V18	Number of grace days for capital payment according to schedule.
V19	Due date of the last installment.
V20	Due date of the next installment.
V21	Periodicity of installments.
V22	Number of scheduled installments.
V23	Number of installments paid.
V24	Indicator if the debtor is refinanced with Agricultural Financial Rescue.
V25	Agency code.
V26	Annual effective rate.
V27	Acceptance or rejection of credit (0: Accepted, 1: Rejected).

Table 2. Database statistics

Variable	Min	Max	Mode	Mean	Deviation Standard	CCO
V1	1	2	1	1	0.03	-0.01
V2	300	297770	1000.00	5879.45	10957.9	0.01
V3	16.24	297770	1000.00	5078.65	10475.32	0.01
V4	8	13	12.00	11.07	1.1	-0.03
V5	0	4	0.00	0.15	0.59	0.43
V6	0	4	0.00	0.13	0.55	0.45
V7	-419	268	-15.00	-14.98	37.64	0.26
V8	0	162	0.00	2.88	8.32	**0.78**
V9	-125	88	0.00	0.02	4.6	0.21
V10	0.83	48000	20.00	136.1	756.33	0.13
V11	0	297770	1000.00	5034.86	10480.37	0
V12	0	26527	0.00	38.05	453.83	0.22
V13	0	29822	0.00	5.75	350.02	0.04
V14	0	19653	0.00	164.71	498.09	0.05
V15	0	5903	0.00	9.03	94.62	0.19
V16	40546	40847	40751.00	40711.96	84.95	-0.14
V17	1	5	3.00	2.94	0.35	0.03
V18	0	360	0.00	6.35	21.54	0.03
V19	40714	48131	41001.00	41198.31	438.13	-0.01
V20	40579	41209	40849.00	40862.16	37.64	-0.26
V21	1	365	30.00	35.84	35.29	-0.02
V22	1	240	12.00	15.65	14.3	0.01
V23	0	37	0.00	3.47	2.77	0.03
V24	1	1	1.00	1	0	NaN
V25	1	44	1.00	18.98	13.38	-0.01
V26	12.01	293.79	58.27	51.09	18.25	0.04
V27	0	1	0.00	0.02	0.14	1

Table 3. Histogram of V27 with properties

Properties:

Data: [15569×1 double]

Values: [15256 313]

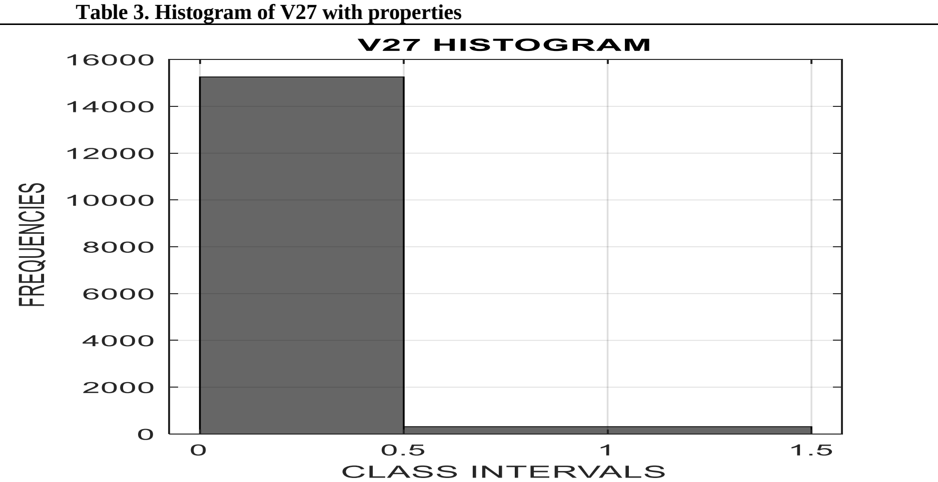
NumBins: 2

BinEdges:[0 0.5000 1.5000]

BinWidth: 'nonuniform'

BinLimits: [0 1.5000]

Normalization: 'count'



2.3. Study of Variables V8 and V27

Table 2 reveals a high correlation of 0.78 between variables V8 and V27 and the others do not exceed 0.45. On the other hand, the histogram (Table 3) reveals that in V27, there are 15,256 components with a value of 0 (customers with accepted credit) and 313 components with a value of 1 (rejected customers). Let B=[V8, V27] be the 2-columnmatrix and N the column of integers from 1 to 15569. In B, column V8 is sorted from lowest to highest together with V27 and the result is stored in BO. Let MO=[N, BO] be the 3-column matrix. It can be observed in MO that up to rows 15,256, the values of V8 vary from 0 to 30, and the values of V27 are all zeros. After this row, the 313 values of V8 grow from 31 to 162, and those of V27 are all constant and equal to 1.From all this, we conclude that the variable V27 depends only on V8 and not on the other variables.

Table 4. Correlations of variables V27 and V8 with the others

Higher V27 correlations with the other variable			Higher V8 correlations with the other variables		
Variables	Variables	Correlations	Variables	Variables	Correlations
V27	V8	78%	V8	V6	53%
V27	V6	45%	V8	V5	51%
V27	V5	43%	V8	V7	36%
V27	V7	26%	V8	V20	-36%
V27	V20	-26%	V8	V16	-24%
V27	V12	22%	V8	V23	10%
V27	V9	21%			
V27	V15	20%			
with the rest below 14%			with the rest below 10%		

Table 5. Variables that have correlations, in absolute value >=70%

Variables	Variables	Correlations
V3	V11	100%
V7	V20	-100%
V2	V3	99%
V19	V22	97%
V5	V6	95%
V16	V23	-87%
V17	V21	-86%
V8	V27	78%
V11	V14	73%
V14	V18	72%

The following relationship gives the aforementioned dependency:

$$V27 = \begin{cases} 0 & if\ \ 0 \leq V8 \leq 30 \\ 1 & if\ \ 31 \leq V8 < 162 \end{cases} \quad (1)$$

In Table 4, it can be seen that the variable V27 correlates with V8 by 78%, followed by V5 and V6 by 43% and 45% respectively. With the others it is below 27%. The V8 variable correlates more with the V5 and V6 variables at 51% and 53% respectively, with the others it is below 37%. The variables V3 and V11, V7 and V20 have very high correlations. Table 5 shows a list of highly correlated variables.

3. Research Methodology

3.1. Preparing the Database for the Neural Networks

The database described in 2.1 is imported into the Matlab environment, where the first 26 columns are stored as the rows of the matrix D and the 27th column as the row of the matrix F. 20% of the columns of the matrices D and F are separated and stored respectively in Pb and Tb; this separation is done using an equivalence class modulo 5, for example, taking the sequence of integers: clas1={1, 6, 11, 16, …. 15566} which in total are 3114 columns. The rest of the columns are stored in P and T, respectively. Then, the BPNN is trained. For this, P and T are normalized. Their rows are transformed into rows with mean zero and standard deviation 1, which are then stored in Pn and tn, respectively. This process is achieved with the mapstd function of MATLAB2018a. Next, a second transformation of Pn into another matrix pn is performed, using the Matlab function processpca, which reduces the number of rows using principal component analysis. In this way, the matrices pn and tn are finally obtained, with 12,455 columns each. The network is trained with pn and tn utilizing cross-validation, which is done in three phases: Training, Validation, and Testing. To do this, the columns of these two matrices are separated into three groups, following an idea similar to what was done in the previous separation: 50% of the columns are for training, 25% are for validation, and 25% are for testing.Once the three phases are completed, the BPNN's forecasting capacity is examined using the columns saved in the Pb and Tb matrices.

3.2. Backpropagation Neural Networks (BPNN)

In this work, supervised neural networks are considered because there is a set of data called expected values, which is a matrix with a single row and M columns: $T = [t_1, t_2, \dots, t_M]$ and on the other hand side of a set of input data to the network, which is a matrix of n rows by M columns: $P = [p_1, p_2, \dots, p_M]$, and what is expected is that the network's response to this data is as close as possible to T.

3.2.1. The Artificial Neuron

The basic unit of a neural network is the artificial neuron, a mathematical model made up of a row matrix w of n components, called weights, by a bias b, which is another variable and a transfer function f: R→R. The response of the neuron for a column p of the data matrix P, which enters it, is

$$q=f(wp+b) \quad (2)$$

Where wp is the product of row w and column p (2) The neuron's response depends on the values of w and b, which when created are variables, but the network becomes numerical values after training. A layer of m neurons is simply a set of m neurons organized in parallel, which act independently of each other. Consequently, the mathematical model of one of these layers is a vector transformation with vector variables:

$$Q=F(Wp+B), \quad (3)$$

Where W is a matrix of mxn and B another of mx1 each row of Q is of the form described in (2).

3.2.2. A Neural Network of m Layers

Since a layer of neurons is a vector transformation, then a neural network is a concatenation or composition of l transformations. In the literature, we talk about the input, hidden, and output layers. In this work, as already said at the beginning, the last layer is composed of a single neuron because the expected values form a matrix $T = [t_1, t_2, \dots, t_M]$ of one row and M components. The response of the neural network for each column p_k of $P = [p_1, p_2, \dots, p_M]$ will be a function of the form: $Net(p_k, x)$ where x represents the weights and biases of all the neurons that comprise the neural network. The response of the BPNN is a row matrix of M columns:

$$Net(P) = [\, Net(p_1, x), Net(p_2, x), \dots, Net(p_M, x)] \quad (4)$$

Training a Backpropagation Neural Network During training, what is expected is that the error in mean square:

$$E(x) = \tfrac{1}{2}\|Net(P) - T\|^2 = \tfrac{1}{2}[(Net(p_1, x) - t_1)^2 + \cdots + (Net(p_M, x) - t_M)^2] \quad (5)$$

Reaches its local or global minimum at a point x*, that is, $E(x^*) \approx 0$.

This means that all of the weights and bias matrices of the neural network adopt numerical values with which the neural network $Net(z, x^*)$ becomes a non-linear transformation, which, for each point z of $\mathfrak{R}^n$ returns a numerical value and, in particular, when $z = p_k$ we have $Net(p_k, x^*) \approx t_k$ which is the expected value of the network. The success or failure of training a BPNN neural network is measured by the Pearson correlation between the network response $Net(P)$ and the row matrix T. Acceptable correlations in this type of problem must be above 85 %.

4. Test Results

4.1. With Backpropagation Neural Networks (BPNN)

Tests have been carried out with different BPNN architectures, taking into account the database described in section 3.1. The network that gives the best results is a 4-layer network with 14 neurons in the first layer, 10 in the second, 8 in the third and 1 in the output layer (In Matlab, this structure is represented as follows: [14, 10, 8, 1]). The transfer functions in the first, second and third layers are the hyperbolic tangent: $tansig(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{2}{1+e^{-2x}} - 1$, in the fourth layer, is the identity function. The BPNN training is performed by activating the Matlab function trainlm, which minimizes the mean square error function (5).

The algorithm that encloses this minimization function is based on articles by Levenberg (1944) and Marquardt (1963). This structure achieves a correlation between the expected values and the BPNN response of 97.68%. When tansig(x) is changed to $logsig(x) = \frac{1}{1+e^{-x}}$ a correlation of 97.09% is achieved. The tests were performed with the BD database described in section 2.2. With the complete BD, then with the BD in which the V8 variable replaces the V27 variable and finally with the BD in which the V8 variable is eliminated. Table 6 summarizes the results achieved in these three situations.

5. Discussion of Results

5.1. Backpropagation Neural Networks (BPNN)

The test results shown in section 4.1 show that the highest precision is 97.68% and has been obtained with the complete database. This high precision of the neural network is explained by the fact that variable V27 depends directly on variable V8. By replacing the variable V27 with the variable V8, the precision obtained by the neural network decreases to 77.39%, but it is still a good precision. It can be stated that variable V8 is a hidden response variable that is the most realistic. Finally, by eliminating variable V8 from the database, the precision of the neural network drops to 66.89%, which confirms that variable V8 is the most realistic and that the remaining 25 variables have a low impact on variable V27. However, in reality, the microfinance company decided to accept or reject its clients, depending solely and directly on the variable V8; that is, if they paid their installments within 30 days, they were accepted. Otherwise, they went to the group of those rejected. Table 7 shows the parameters used and the results obtained for the neural network prediction in studies carried out by other authors and in the present work.

Table 6. Test results with BPNN

Database	Global Precision	Architecture	Observation
(27 variables)	0.97682	[14, 10, 8, 1]	Complete database, with 26 independent variables and the V27 variable of credit acceptance or rejection.
(26 variables)	0.77389	[14, 10, 8 ,1]	Database with 25 variables, in which variable V8 replaces variable V27.
(26 variables)	0.66897	[14, 10, 8, 1]	Database with 25 variables, V8 variable removed.

Table 7. Comparison of the parameters used and the results obtained for the prediction of the neural network in the present study and studies carried out by other authors

Authors	No. of clients	No. of variables	No. of ANN layers	Correlation R	Area under the ROC curve
[11]	1000	20	3	85.9%	
[12]	690	14	3	88.84%	
[2]	492	12	3	91.3%	
[3]	492	12	3	81.5%	
[4]	5451	39	3		0.9543
[15]	1100	9	4	70.5%	
In the present study	15569	26	3	97.68%	

6. Conclusion

a) The variable V27 depends only on V8 and not other variables. This dependency is described by the relationship:

$$V27 = \begin{cases} 0 & if\ \ 0 \le V8 \le 30 \\ 1 & if\ \ 31 \le V8 < 162 \end{cases} \quad (1)$$

This shows that the microfinance company mentioned in 2.1 did not use the 25 different input variables of V8 to decide whether to accept or reject the continuation of the loan. This situation is also reflected in Table 4, where the highest correlation of V27 is with V8 (78%), with the other variables being below 45%.

b) The conclusions of part (a) raise the need to study variable V8 (which is the days late in paying the last installment). That is, it is about seeing what variables the decision made by the client depends on. to pay your fee within 30 days or more. Table 4 reveals that it depends on the variables V5 and V6 by 51% and 53%, respectively; on the others, it is below 36%.

c) With the BPNN, a good level of precision is obtained in the prediction, in particular with the one that has the structure [14, 10, 8, 1] and transfer functions tansig(s) for the first three layers and the identity function for the fourth layer. With this network, a correlation of 97.68% is achieved and obtained by processing the entire database. This high precision of the neural network is explained by the fact that variable V27 depends directly on variable V8.

By replacing the V27 variable with the V8 variable, the precision obtained by the neural network decreases to 77.39%, but it is still a good precision. It can be stated that variable V8 is a hidden response variable that is the most realistic. Finally, by eliminating the variable V8 from the database, the precision of the neural network drops to 66.89%, which confirms that the variable V8 is the most realistic and that the remaining 25 variables have a low impact on the variable. V27.

d) The way in which they should use technologies. Modernize your database's structure and functions so you can communicate with the BPNN on decision issues in real time.

e) Recommendations to microfinance institutions for using BPNN to predict clients' credit behaviour, which would allow them to make better decisions to accept or reject.

References

[1] Lobna Abid, Afif Masmoudi, and Sonia Zouari-Ghorbel, “The Consumer Loan’s Payment Default Predictive Model: An Application of the Logistic Regression and the Discriminant Analysis in a Tunisian Commercial Bank,” *Journal of the Knowledge Economy*, vol. 9, no. 3, pp. 948-962, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Hussain Ali Bekhet, and Shorouq Fathi Kamel Eletter, “Credit Risk Management for the Jordanian Commercial Banks: A Business Intelligence Approach,” *Australian Journal of Basic and Applied Sciences*, vol. 6, no. 9, pp. 188-195, 2012. [Google Scholar] [Publisher Link]

[3] Hussain Ali Bekhet, and Shorouq Fathi Kamel Eletter, “Credit Risk Assessment Model for Jordanian Commercial Banks: Neural Scoring Approach,” *Review of Development Finance*, vol. 4, no. 1, pp. 20-28, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[4] Antonio Blanco et al., “Credit Scoring Models for the Microfinance Industry Using Neural Networks: Evidence from Peru,” *Expert Systems with Applications*, vol. 40, no. 1, pp. 356-364, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[5] Daniel Carpio Martin, “Mapas Autoorganizados (Self-Organizing Maps),” *University of Valladolid*, 2020. [Google Scholar] [Publisher Link]

[6] Guido J. Deboeck, “Financial Applications of Self-Organizing Maps,” *Neural Network World*, vol. 8, no. 5, pp. 213-241, 2000. [Google Scholar]

[7] Guido Deboeck, and Teuvo Kohonen, *Visual Explorations in Finance: with Self-Organizing Maps*, Springer Finance, 2010. [Google Scholar] [Publisher Link]

[8] Gülnur Derelioğlu, and Fikret Gürgenm, “Knowledge Discovery Using Neural Approach for SME’s Credit Risk Analysis Problem in Turkey,” *Expert Systems with Applications*, vol. 38, no. 8, pp. 9313-9318, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[9] Lu Han, Liyan Han, and Hongwei Zhao, "Orthogonal Support Vector Machine for Credit Scoring," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 2, pp. 848-862, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[10] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*, 2nd ed., Springer Series in Statistics, 2009. [Google Scholar] [Publisher Link]

[11] Adnan Khashman, "Neural Networks for Credit Risk Evaluation: Investigation of Different Neural Models and Learning Schemes," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6233-6239, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[12] Adnan Khashman, "Credit Risk Evaluation Using Neural Networks: Emotional Versus Conventional Models," *Applied Soft Computing*, vol. 11, no. 8, pp. 5477-5484, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[13] Kiruthika, and M. Dilsha, "A Neural Network Approach for Microfinance Credit Scoring," *Journal of Statistics and Management Systems*, vol. 18, no. 1-2, pp. 121-138, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[14] Kenneth Levenberg, "A Method for the Solution of Certain Problems in Least Squares," *Quarterly of Applied Mathematics*, vol. 2, pp. 164-168, 1944. [CrossRef] [Google Scholar] [Publisher Link]

[15] Rashmi Malhotra, and D.K. Malhotra, "Identifying Potential Loan Defaulters in the Credit Union Environment: A Comparative Analysis of Statistical and Neural Network Models," *Journal of Information Technology Case and Application Research*, vol. 2, no. 2, pp. 20-48, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[16] Donald W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431-441, 1963. [CrossRef] [Google Scholar] [Publisher Link]

[17] Quispe Zenón, León David, and Contreras Alex, "The Successful Development of Microfinance in Peru," *Currency Magazine*, no. 151, pp. 13-18, 2012. [Google Scholar] [Publisher Link]

[18] J. Vesanto, and E. Alhoniemi, "Clustering of the Self-Organizing Map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586-600, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[19] Nathalie Villa-Vialaneix et al., "SOMbrero: SOM Bound to Realize Euclidean and Relational Outputs," *R Package*, 2020. [CrossRef] [Publisher Link]

[20] Lucy Conger, Patricia Inga, and Richard Charles Webb, *The Mustard Tree: History of Microfinance in Peru*, University of San Martin de Porres, 2009. [Google Scholar] [Publisher Link]

[21] Yongqiao Wang, Shouyang Wang, and K. K. Lai, "A New Fuzzy Support Vector Machine to Evaluate Credit Risk," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 820-831, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[22] Zongyuan Zhao et al., "Investigation and Improvement of Multi-Layer Perception Neural Networks for Credit Scoring," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3508-3516, 2015. [CrossRef] [Google Scholar] [Publisher Link]