A Stratified Sampling Algorithm for Artificial Neural Networks

38

Danilo Douradinho Fernandes, Gustavo Ravanhani Matuck, Denis Avila Montini, Luiz Alberto Vieira Dias, and Alessandra Avila Montini

Abstract

Artificial Neural Networks (ANN) MultiLayer Perceptron (MLP) are widely applied in a variety market segments to handle with real complex problems. The ability to deal with tasks in real time is essential in an environment that uses large volume do information available. In each new project, a decision-making system using ANN with time reduction and data processing is a key issue to test various learning algorithms; containing a variety of parameters when using this technology. From this starting point, the MLPs used data collected from a specific phenomenon and, based on statistical estimators, applied a data extraction algorithm for stratified sampling, aiming to reduce the time of ANN processing. In this context, this work proposes a Stratified Sampling algorithm (SSA), which was developed to minimize processing MLPs time without losing coverage and assertiveness, when comparing with training conducted on a population database. The case study consisted of a ANN performance influence with a population database and with its sample data obtained by the SSA model. This procedure with the RNAs aimed to evaluate the following properties: (1) meet the pre-established criteria of reliability of the model; (2) have a computer-automated procedure; (3) sort and select records more correlated, and (4) maintain sampling results within a track of assertiveness of total results obtained. From the realization of this case study, it was possible to identify the following gains made by the (1) reduction of ANN processing time by providing: (2) optimization of processing time; (3) automatic network selection; and (4) automatic parameters selection for training algorithms.

Keywords

Artificial Neural Network · Stratified Sampling Algorithm · Multilayer Perceptron

38.1 Introduction

There are several phenomena in nature that can only be explained by means of variables set. For example, the behavior of bank fraud in credit card use or credit risks analysis. For a wide range of making decision support situations, those behaviors can be monitored and classified through Artificial Intelligence (AI) techniques approaches. From this AI concepts, Artificial Neural Networks (ANN) MultiLayer Perceptron (MLP) [1] can be used to perform complex financial tasks.

Due to the databases size and the calculations amount required to reach the results, it has been possible to contextualize an inability to diagnose real time credit card risks analysis or detect frauds. In this context, to diagnose and improve results for this scenario, it was necessary to consider statistical approaches, providing a hybrid computational solution. However, was it possible to decrease data volume considering computational process and get the same performance rate? One possible way to handle this issue is the use of statistical concept called sampling.

The sampling approach aims to reduce the amount of data records from the database, obtaining a data sample with similar characteristics of the population database. However, there are some limitations that may occurs, such as the inability to deal with data mining without distorting behavior, as well as need to use a large data amount. This type of

D. D. Fernandes
Federal Institute of Education, Science and Technology of São Paulo (IFSP), Campinas, Brazil

G. R. Matuck · D. A. Montini (✉) · L. A. V. Dias
Computer Science Division, Brazilian Aeronautics Institute of Technology (ITA), São José dos Campos, Brazil

A. A. Montini
Department of Administration, University of São Paulo (USP), São Paulo, Brazil

S. Latifi (ed.), *Information Technology – New Generations*, Advances in Intelligent Systems and Computing 738,
https://doi.org/10.1007/978-3-319-77028-4_38

restrictions has been creating difficulties to apply real time computational models for financial segment [2].

Statistical sampling models can be applied in databases to be able to reproduce an original population pattern [8]. In ANN approaches conducted for finantial tasks, these models enable same detection performance level. However, in a shorter time interval [2].

There are some limitations when these techniques are used separately [2]. Due the high complexity of most nonlinear real problems, some techniques applied alone have more difficulties to address some behaviors. For example, situations that have many variables, many input/output mapping and huge amount of available data. In order to overcome these model's limitations, there is a possibility to use statistical data sampling for the neural system optimization learning [3–5], a stratified sampling algorithm based on an index of reliability, producing low computational processing cost [2].

In this work, a sample extraction algorithm has been enhanced, using statistical fundamentals in an algorithm that traverses the vector of size N. Using a computer program developed with MATLAB® software [6], some results and comments are shown in the following sections.

This article is organized as follows: Sect. 38.1 Introduction discusses elements of a stratified sampling characterization to interact with an MLP ANN; Sect. 38.2 the Stratified sampling applied for real problems is described; Sect. 38.3 the SSA algorithm is described; Sect. 38.4 Artificial Networks; Sect. 38.5 describes Neural Networks and Sampling for Banking Credit Risks Analysis; Sect. 38.6 Computational Model Results evaluation; Sect. 38.7 presents the conclusion and future work and, we finsih with Acknowledgements and general considerations.

38.2 Stratified Sampling Applied for Real Problems

The stratified sampling used in this work belongs to probability samples family and consists of dividing the entire population or "study object" in different subgroups or different strata. So, that an individual can be part only of a single stratum or layer [3].

After the layers were defined, to create a data sample, the model selects individuals using any sampling technique in each layer separately [3].

A stratified sample is considered excellent, as this sampling that has a small standard deviation. This concept was fundamental to calculate: (1) layers size; (2) sample blocks size; as well as (3) sample size.

On the other hand, the extraction of the samples in each layer was progressively adjusted during the calculation of the standard deviation of the studied variables.

The program extracted from each layer only some data that met the population standard deviation, aiming that the final sample was obtained to obtain the best possible population representation [3].

The SSA developed was applied in a variety of real case studies, aiming to obtain representative samples, dealing with large volumes of data, in linear time "n". However, due the confidentiality context, in this article the case studies are explored to provide a better demonstration about the results and application possibilities.

Any population data can be divided into "n" subdata groups, or in "K" stratified cluster, with enough size to be processed by the available hardware used. For the experiments, the data cluster size was tested arbitrarily, exploring different scenarios. The data extraction process in the population database can be conducted just once, for each subdata cluster [3], Fig. 38.1 shows an overview.

38.3 SSA Algorithm

During the sampling process, the SSA model take decision based on statistical indicators. The algorithm was designed to identify which data records can statistically be selected for the sample database. The sample process stops after the criterias are reached, as well can also perform a swap of data records by another more appropriate between the sample and population databases.

The population database is divided in data cluster, with size "n" predefined by the user ($n > 0$). The input vector "V" of the data cluster is processed by the SSA to provide volume reduction of this vector "V" as the following steps:

SSA algorithm steps:
A- Plan (P) –Algorithm structuring:

1. Parameters definition for the algorithm execution.

B-Do (D) – Population Analysis

2. Get the data Cluster.
3. Parameters definition.
4. Correlation analysis (Cluster/Population).
5. Estimated population indicators calculations.

X1- Analysis and data visualization.

6. Cluster Indicators evaluation.
7. Outsiders analysis.
8. Indicators (cluster without Outsiders) evaluation.
9. Cluster Variances calculation sample size calculation.
10. Data analysis and visualization.
11. Sample extraction and Verification.
12. Implementation of Qualitative Sampling

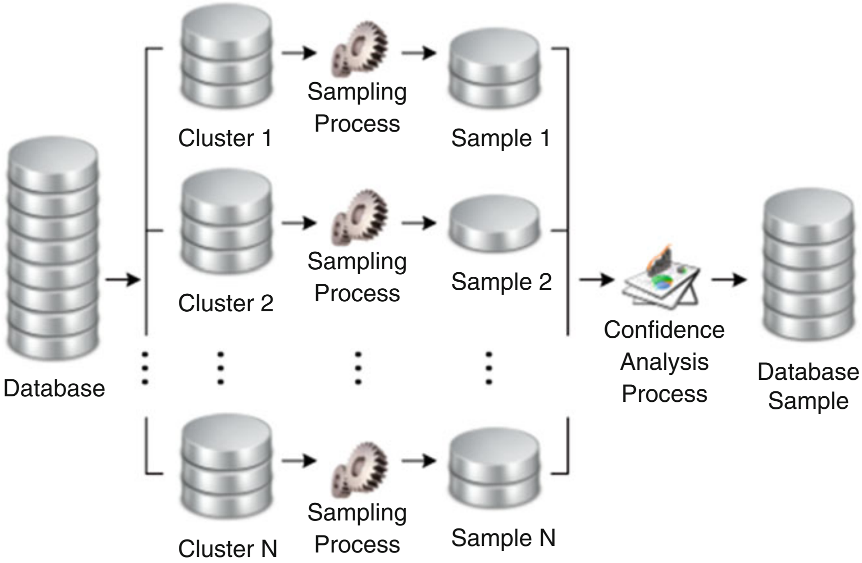


Fig. 38.1 Data clusters process overview

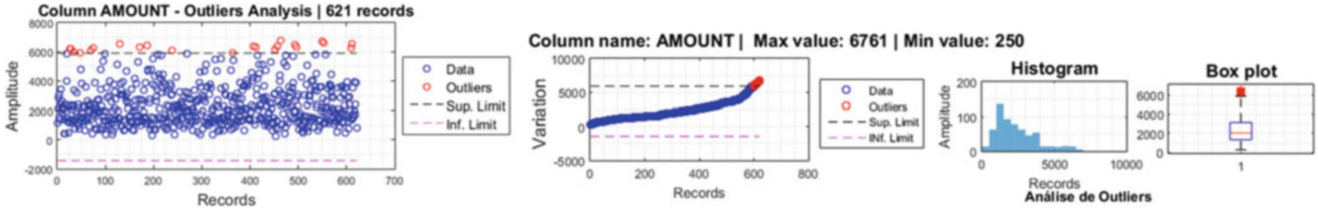


Fig. 38.2 Descriptive statistical indicators used in Algorithm AEN

C - Check (C) – Sapling consolidation

13. Final reconciliation of all clusters. Statistical indicators calculation and lower and upper limits of final sample.
14. X13 Data Visualization and Analysis

D – Act (A) - Corrective Actions and Results evaluation

15. Sample representativeness verification.

From line 1 to 14 of SSA algorithm steps, various statistical calculations are performed in sample obtained.

The algorithm AEN complexity is estimated as follows, being log (10 n), 10 (n), O (n) advance only indicators calculations are held, where n is data amount generated. If > and > n, being O (n) complexity order (Fig. 38.2).

From SSA explanation above, identified that an asymptotic order was estimated in O(n) for each cluster. Even so, N times (n), an estimated order is maintained, clusters number regardless.

38.3.1 Statistical Concepts Used

The SSA proposed algorithm uses statistical indicators, in order to assist in selection sampling procedure criteria. A unique feature of this algorithm consisted in fact that their parameters use is flexible, and need to be continuously defined [6, 7].

In other words, numerical parameters definition can be assigned arbitrarily on bookmarks each. However, each descriptive statistical indicator [3] may or may not participate in data extraction criteria of sampling SSA. In sampling integrated process with ANN.

This approach advantage could be established, considering fundamentally: (1) Descriptive data analysis use techniques in a program to sort the data, keeping your representativeness; (2) Processing time reduction; (3) Some quantitative properties maintenance in final sample. The SSA algorithm can be used in different applications [10].

Table 38.1, we have the SSA Algorithm insert containing some requirements applied to RNA training process [3, 4].

Table 38.1 Parameters identified in SSA algorithm

Parameter ID	ID	Valuation	Range
1	Population size	621	
2	Sample size	175	
3	Confidence interval identified	0.99	
4	Cluster size	200	
5	Classification of interest columns	C	Continuous data type = “C”
6	Reliability parameter	0.7	0.7
7	Correlation test	1	Selected to explain problem
8	Indicators set selection, set number	1	1 set of indicators uses: ‘Arithmetic’; ‘Weighted Average’; ‘Median’; ‘Variance’; ‘Standard Deviation’; ‘Amplitude’, P; ‘Quartile 1’; ‘Quartile 2’; ‘Quartile 3’; ‘Quartile 4’
9	Graphs set selection, the number	1	1 set of indicators uses: (1) pizza; (2) bar; (3) points of the population; (4) population x sample points; (5) BoxPLot; (6) the confidence interval (CI); (7) convergence
10	Parâmetros de conexão a base de dados database connection parameters	1	1 set of connection parameters to the database: (1) path; (2) user; (3) password; (4) access privilege
11	Sample database connection parameters	1	1 set of connection parameters to the database: (1) path; (2) user; (3) password; (4) access privilege

In order to reduce: (1) the processing time, (2) bandwidth and (3) response time, a MLPs RNA technology was used by sampling aimed to reduce and avoid the large data amounts. After describing some SSA algorithm properties, then an introduction about this algorithm were presented.

This algorithm has several adjustable parameters to adjust the sampling performance in each new established phenomenon. A statistical parameterization analysis of SSA scores could be performed around a central concept (CI) confidence interval. Initially, each analyzed cluster is calculated, some indicators were described in Table 38.2.

The SSA algorithm calculates the values of the cluster, in other indicators [3], the values were accumulated to obtain a mean of all the multiple groupings, according to these data had to be processed as well as processed in the computer’s memory.

Table 38.2 Descriptive statistical indicators used in the algorithm AEN

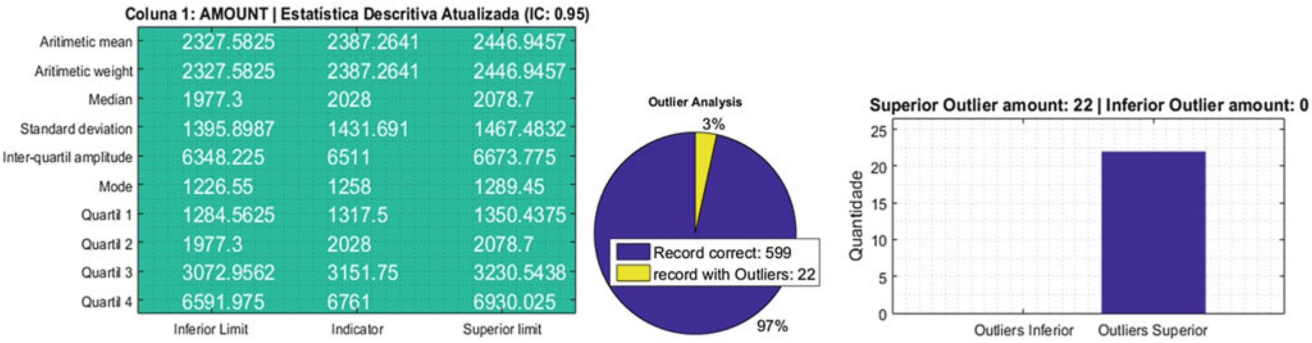
Confidence interval (CI)		0.99	0.01
Average values from “N” cluster-assumed to population values			
Statistical indicators	Lower bound	Indicated value r	Upper bound
‘Arithmetic Mean’	2375.33	2387.26	2399.20
‘Weighted Average’	2375.33	2387.26	2399.20
‘Median’	2017.86	2028.00	2038.14
‘Variance’	2,039,490.33	2,049,739.02	2,059,987.72
‘Standard Deviation’	1424.53	1431.69	1438.85
‘Interquartile Range’	6478.45	6511.00	6543.56
‘Quartile 1’	1311.41	1318.00	1324.59
‘Quartile 2’	2017.86	2028.00	2038.14
‘Quartile 3’	3133.26	3149.00	3164.75
‘Quartile 4’	6727.20	6761.00	6794.81

38.3.2 Data Representativeness Evaluation

In this session, some graphs obtained during the process were presented. During the realization of the representation, the same calculation application filters were applied for both the population and the samples. In this context, in order to filter and remove the Outliers, an identified way to perform the confidence interval (CI) calculation was designed to test decrementally from 1 to 0.00. For this achievement, the population data were used in all indicators, being exemplified in Table 38.2 and in Fig. 38.3. In these cases, the same results obtained in sample were presented in Table 38.2.

This was a structured form that the SSA algorithm used to compare the Reliability Index (IC). This was one of tests performed to identify whether the sampling was within the confidence range obtained from the population.

The consolidated data from this SSA program processing was shown in Fig. 38.3. The SSA program recursively examined the NEA to refine and progressively improve and dynamize the IC.



Coluna 1: AMOUNT | Estatística Descritiva Atualizada (IC: 0.95)

	Inferior Limit	Indicator	Superior limit
Aritimetic mean	2327.5825	2387.2641	2446.9457
Aritimetic weight	2327.5825	2387.2641	2446.9457
Median	1977.3	2028	2078.7
Standard deviation	1395.8987	1431.691	1467.4832
Inter-quartil amplitude	6348.225	6511	6673.775
Mode	1226.55	1258	1289.45
Quartil 1	1284.5625	1317.5	1350.4375
Quartil 2	1977.3	2028	2078.7
Quartil 3	3072.9562	3151.75	3230.5438
Quartil 4	6591.975	6761	6930.025

Fig. 38.3 Descriptive statistical indicators used in SSA Algorithm

By means of these parameterized tests, the algorithm recursively processed to the moment that SSA stop conditions program were satisfied and the final sample was obtained.

38.4 Artificial Neural Networks

Artificial Intelligence (AI) is a knowledge field that has being used in various market segments all around the world [4]. Nowadays AI can be experienced on market in a variety of services, products and business analysis. The Artificial Neural Networks (ANNs) is a specific AI subset that has a similar behavior as human brain natural network, both can analyze information, interpreting the data and making decisions [4, 7].

There are many types of ANNs in the scientific literature. A common neural network architecture is called MultiLayer Perceptron (MLP) [4, 7], widely applied in real complex problems. MLP network is structured by layers of neurons, an input layer, one or more hidden layers and the output layer, performing nonlinear interpolations for many problems. There are many learning algorithms that can be used to perform learning tasks, a common one used is called Backpropagation algorithm.

Financial Market attracts interest of many applications of AI, especially dealing with banking frauds detection, stock markets and bank credit risks analysis. Such issues have distinct characteristics like big data information available for analysis (non-structured) and many different variables involved. High computational power, complex computational algorithms and big data analysis capability, should support this kind of environment, providing real time effective solutions.

38.5 Neural Networks and Sampling for Banking Credit Risks Analysis

This work concentrates details only for the application with banking credit risk analysis. The computational model developed was also applied and evaluated in others real problems case studies. For example, applied for banking frauds detection and tax evaders identification, facing with millions of data, high power computed required and showing good performance results.

In this work, a sampling algorithm and ANN computational model were developed to perform a banking credit risk analysis.

This task is often done by finance specialist professionals, that analyze a variety amount of information, providing a score to allow or not financial credit for customers. To provide better analysis in real time, these companies are also using expert computational systems to handle this problem and reach good results.

For this study case, the data used have information about bank's customers, like age, profession, if have kids, if is employed or not, has car (new or used), education and so on. According with this information, 70% of customers were considered as good credit, with low risk. The other 30% of customers were labeled as bad credit, with high risk of compliance.

This unbalanced data proportion of good and bad credit profile can influence the neural network model performance during the learning phase. To avoid unwanted behavior to analyze credit risks by neural networks, some experiments performed data sampling process with good credit user profile.

An exploratory data analysis is performed in the database and the data behavior influence for credit risks was statistically studied. Some data variables can influence the neural networks learning performance and should be individually structured.

For this study, the neural network model was composed by a MLP neural network, its architecture was configured combining variety for strategies like, for example, with one and two hidden layers, different learning algorithm functions, momentum term, learning rate adaptative, different error thresholds, different data proportions for training, validation and test sets, different approaches for data normalization and so on.

After execution all experiments, the results were analyzed, aiming to find the best configuration for banking credit risks analysis.

38.5.1 Strategy Applied for Banking Credit Risk Analysis

Figure 38.4 shows the computational model process adopted for the banking credit risks analysis simulations. In each process step it was considered a variety of specific approaches. For a better comprehension about this study case simulation, four experiments were conducted, using or not the removing database's outliers and performing or not the sampling process.

This scenario provides a contextualization for understanding the ANN performance influence combining the sample process and data quality (without outliers). Figure 38.5 shows the experiments approaches.

The model works as follow. First the database (1) provides the data information used as an input for the computational model.

The next phase is to perform data analysis (2) considering different aspects like, for example, quantitative and qualitative data, missing data, performing data treatment and exploring some statistical approaches. This type of data analysis is required to produce a structured data (3), with useful information considered for banking risk analysis.

The stratified sampling algorithm developed (4) considered the following scenario: only records with good credit were sampled by SSA. The records labeled with bad credit were maintained integrally with the data sample obtained of the records with good label. The database sample was applied for the neural model learning process. Figure 38.6 shows the sampling process conducted for credit risks analysis.

Figure 38.7 shows de Neural network model (5) applied for experiments. The structured database was pre-processed by data normalization and division for training, validation and test sets. The MLP architecture was also configured according different strategies. The MLP network is trained to identify good and bad credit risks. After de MLP network training process, the post processing phase analyses the learning performance by the model.

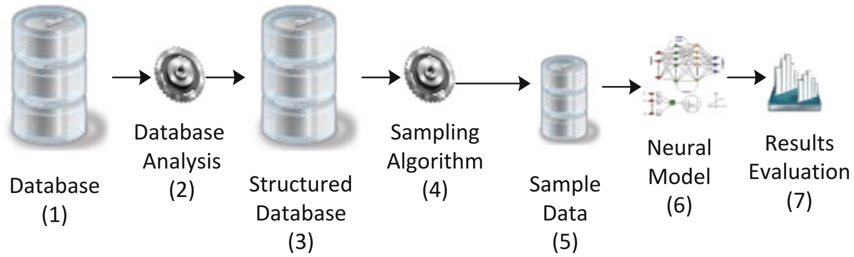


Fig. 38.4 Computational model approach

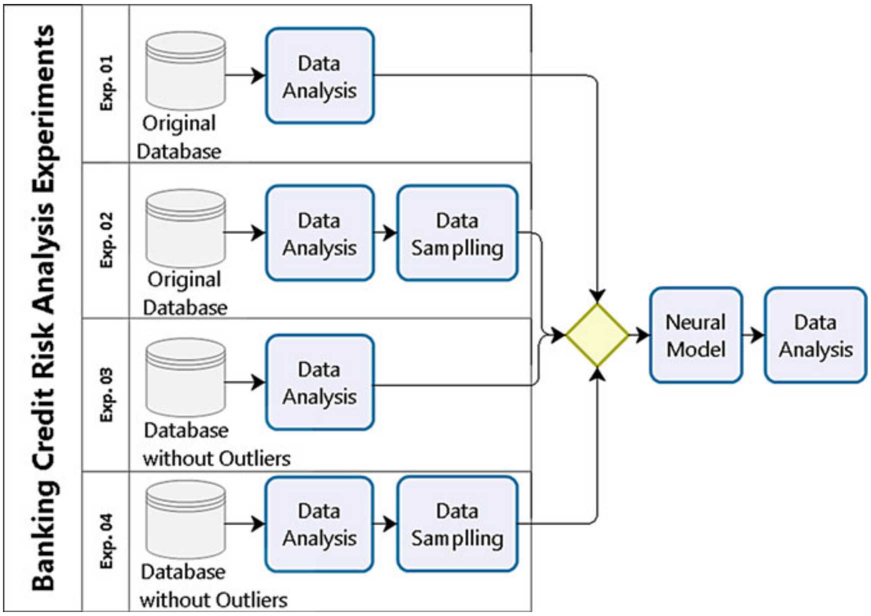


Fig. 38.5 Experiments approaches

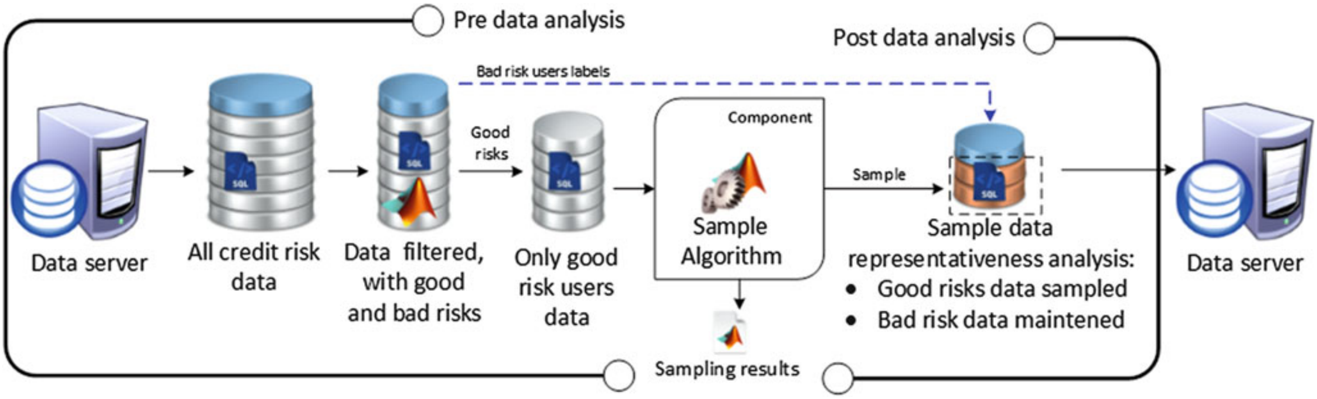


Fig. 38.6 Sampling process approach

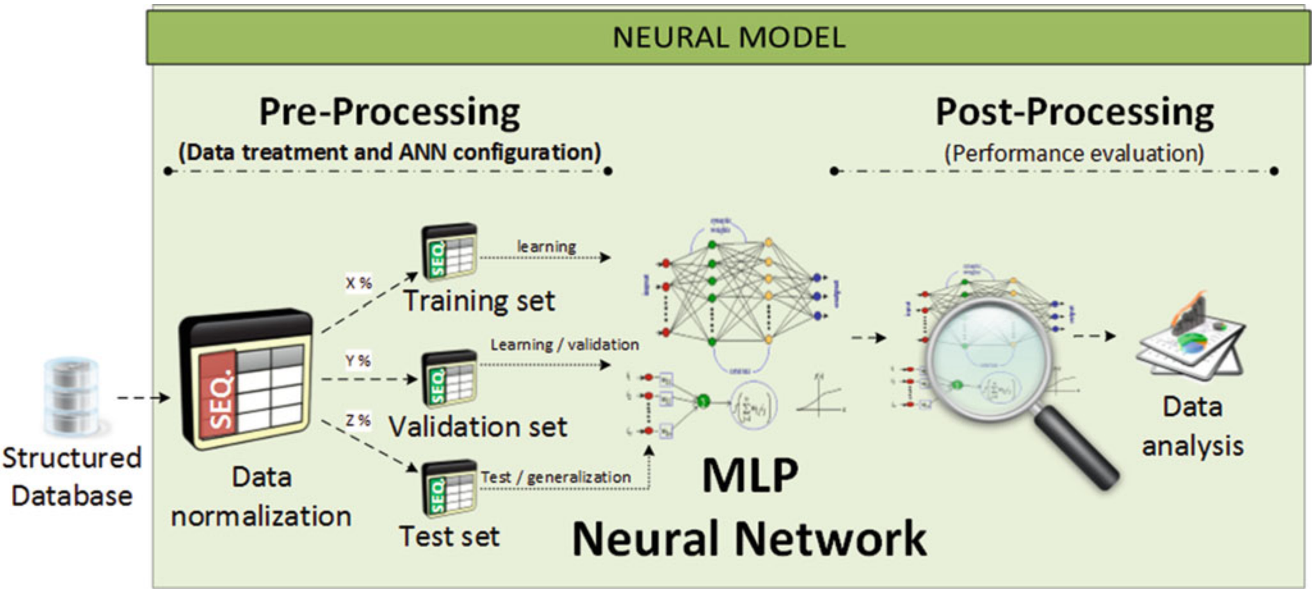


Fig. 38.7 Neural Network Model

38.5.2 Sampling Algorithm and MLP Neural Networks Application

The development computational model used MATLAB® software. Once implemented, the execution works automatically for each experiment, evaluating different neural networks configurations. An html report was produced with all simulation details.

38.6 Computational Model Results Evaluation

The database was composed by 1.000 registries and 30 different variables about costumer's information, 700 records labelled with good credit and 300 with bad credit of compliance. All data were analyzed and pre-processed, resulting a structured database to be used in the simulations experiments.

For the experiments conducted using data sampling process, only quantitative data variables were considered by the SSA. It was tested different parameterization aspects for the SSA sampling process. The best configuration found reduced the 700 registries labelled with good credit to 362 patterns, 51% of reduction. The model representativeness obtained 84% confidence interval. After the sampling process by SSA, the 300 registries labelled with bad credit were grouped with 362 registries sampled with good credit, totalizing 652 patterns for the neural model phase.

Before the neural model learning process, the data should be binaryzed and normalized between [0,1] range, considering quantitative and qualitative data aspects. Each normalized registry/pattern was composed by a vector with 39 data entries. All processed data were divided into training, validation and test sets, with different amount proportions for each group. For this study case, the sort data considered proportions like 70%, 80% and 90% for training set respectively, 15%, 10%, and 5% for validation and test set.

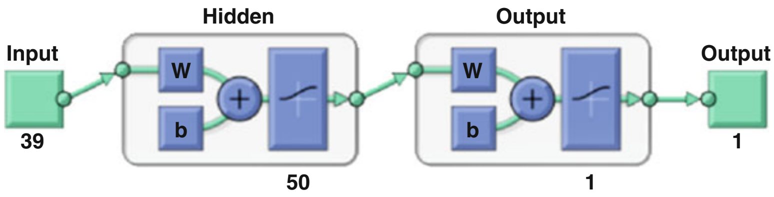
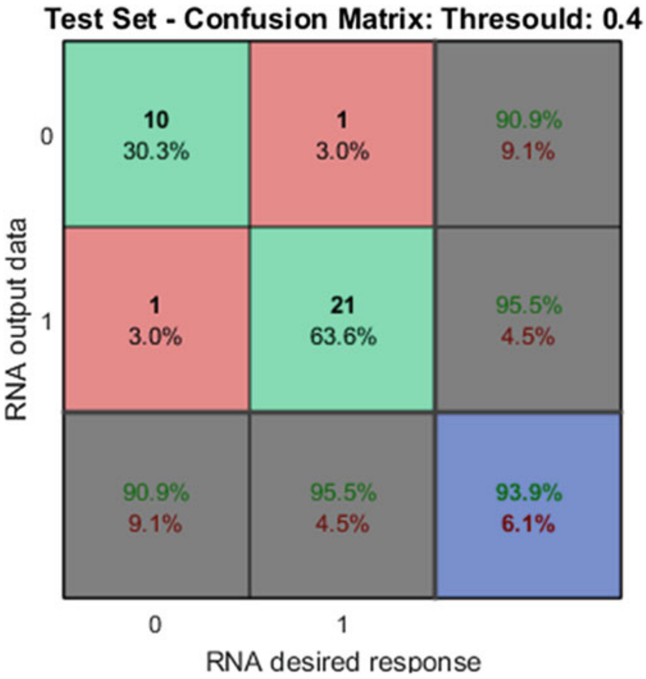


Fig. 38.8 MLP Neural Network Architecture



Test Set - Confusion Matrix: Thresould: 0.4

RNA output data \ RNA desired response	0	1	
0	10 30.3%	1 3.0%	90.9% 9.1%
1	1 3.0%	21 63.6%	95.5% 4.5%
	90.9% 9.1%	95.5% 4.5%	93.9% 6.1%

Fig. 38.9 Test set matrix confusion by ANN

The MLP neural network considered different learning algorithms like, for example, Levenberg-Marquardt, Scaled Conjugate Gradient, Bayesian Regularization, Conjugate Gradient, and Gradient Descent with Momentum. Random initial weights and bias values were used for the learning process [3].

After widely exploration of all experiments for banking credit risk analysis, the best ANN model found has only one hidden layer, with 50 neurons (Fig. 38.8). Logistic sigmoidal functions were applied for all neurons layers, Bayesian Regularization [9] training algorithm, adaptative momentum term and learning rate parameters were considered for this MLP network.

For the model learning performance evaluation, the test set was analyzed. Figure 38.9 shows the MLP network matrix confusion for the test set classification by ANN.

Other analysis carried out for this case study was to analyze the coverage and assertiveness of the neural model classification for unseen data (test set). Figure 38.10 shows set Coverage X Assertiveness and Test Set Alerts produced analysis.

Using 0.4 value for threshold, neural outputs more than this threshold value normalize to 1 and 0 instead, Figure 11 shows the MLP performance, considering coverage, assertiveness, alerts produced, with correct and wrong classification.

The four experiments conducted for banking credit risk analysis showed good results for the neural network generalization (test set), up to 90% of coverage and assertiveness.

The experiments conducted with sampling process take much less computational time consumption and reach good results. These results show de possibility of use this approach for real time problems.

38.7 Conclusion and Further Work

This work presented a sampling Stratified Sampling Algorithm for data reduction. SSA uses statistical descriptive data analysis techniques to sort data, allowing sample selection containing only the most relevant data for application.

The SSA model was applied in different real databases, reducing in same cases 90% of volume data and scoring the samples with 80% or more for confidence intervals.

The experiments conducted showed the ability for the SSA computational model handle with huge amounts of data, processing in linear time. All the samples extracted considering different databases (some databases with millions of data), producing samples with high confident intervals. The sampling results presented in this work, considering the banking risk credit analysis case study, showed the strategy for the sampling process, reducing the unbalanced data by good credit risk customers and reflecting a good neural network performance.

As a future work, extend the SSA algorithm to handle qualitative data, as well as performing an extra permutation data between sample and database, increasing the confidential interval for the data sample obtained by the model.

Acknowledgements and General Considerations The Research Group on Software Engineering thanks for institutions and research groups like the Brazilian Aeronautics Technological Institute (ITA) and Administration Institute Foundation (FIA), for the contributions, support and cooperation.

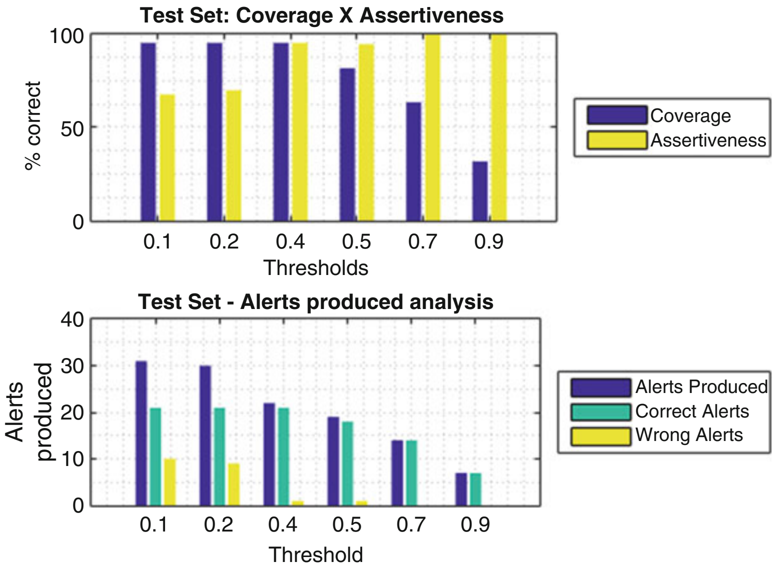


Fig. 38.10 Test set performance Analysis

References

1. S. Haykin, *Neural Networks and Learning Machines* (Pearson Education, Inc., Upper Saddle River, NJ, 2009)
2. D.A. Montini, G.R. Matuck, A.M. da Cunha, L.A.V. Dias, A sampling diagnostics model for neural system training optimization, in *2013 10th International Conference on Information Technology: New Generations* (Las Vegas, 2013)
3. G.R. Matuck, J.R. Barbosa, C. Bringhenti, I. Lima, Multiple faults detection of gas turbine by MLP neural network, in *ASME Turbo Expo 2009, Power for Land, Sea, and Air* (Orlando, 2009)
4. S. Russell, P. Norvig, *Artificial Intelligence. A Modern Approach* (Prentice Hall, 2003)
5. D.A. Montini, P.M. Tasinaffo, A.A. Montini, L.A.V. Dias, A.M. Cunha, Um Meta-Algoritmo para Otimização de Planejamento em Linha de Produção de software, in *VIII International Conference on Engineering and Computer Education—ICECE 2013* (Luanda, 2003)
6. The MathWorks, Inc, MATLAB Software [Online], https://www.mathworks.com/products/matlab.html. Accessed 06 Apr 2017
7. W.J. Stevenson, B.J. Isselhardt, *Study Guide to Accompany Business Statistics: Concepts and Applications*, 2nd edn (Harper & Row, 1978)
8. C.M. Bishop, *Neural Networks for Pattern Recognition* (Clarendon Press, Oxford, 1995)
9. D.A. Montini, D. Battaglia, G.R. Matuck, A chi-square methodology applied in deviations control of project plan to support the RIMAM model, in *2014 11th International Conference on Information Technology: New Generations* (Las Vegas, 2014)
10. Z. Yue, Z. Songzheng, L. Tianshi, Bayesian regularization BP Neural Network model for predicting oil-gas drilling cost, in *2011 International Conference on Business Management and Electronic Information* (Guangzhou, China, 2011)