# Improving Default Risk Information System with TensorFlow

Cláudio Augusto Silveira Lélis
*Scientific Initiation Program*
*IMES / Faculty ImesMercosur*
Juiz de Fora, Brazil
e-mail: claudioaugustolelis@imes.org.br

André Luiz Silveira Lopardi
*Master's Program in Business Administration*
*FUMEC University*
Belo Horizonte, Brazil
e-mail: andrelopardi@yahoo.com.br

***Abstract*—The decision process is essential in the granting of credit. The right decision can be critical in reducing financial losses. DeRis (Default Risk Information System) is an information system designed to support activities in the management of default risk. The main component is a predictive model of default based on indicators. Currently, the system has been improved allowing models of the TensorFlow tool. Based on real datasets, the default model and the predictive models of the TensorFlow tool was evaluated for different types of indicators. The results show that a model optimization was possible through the adjustment of the hyperparameters offered by TensorFlow, with 240 distinct combinations being tested between these hyperparameters. Although the results are associated with the data and the design of the experiment conducted, they were considered positive and promising for future work.**

***Keywords*–Indicators; Financial Management; Knowledge-based Decision-making; Default Prediction Model; TensorFlow.**

## I. INTRODUCTION

Over time, companies have been adapting to changes while remaining competitive and profitable in an increasingly crowded market. Applying constant investments in the area of Information Technology, financial institutions seek to offer products to their customers in a fast, safe and high technological value. Always attentive to high performance and information security, especially with the large volume of data. On the other hand, customers can count on the trust, performance and safety expected of a financial institution [1].

In this period, while efficiency has gained prominence, companies continue to analyze risks, reduce losses, and maximize efficiency. For instance, a financial institution should identify risks in lending situations, draw conclusions as to the borrower's ability to repay, and make recommendations regarding the best structuring and type of loan to be granted in the light of the applicant's financial needs [2]. In a scenario of uncertainties and incomplete information, risk analysis involves the ability to establish a decision rule to guide the granting of credit.

According to [3], credit risk is associated with the risk of a borrower or counterpart being defaulted. Thus, in the position of financial intermediaries, banks must act in a way that minimizes risk and enables fairer terms of credit acquisition. The difficulty of performing guarantees and recovering credit has led to uncertainty and instability in the market, making default the biggest cost of a bank's financial margin.

Although there are different concepts, default in the context of this research can be understood as a delay of more than 90 days in the liabilities assumed with a financial institution [4].

The use of default prediction models serves to measure, monitor and predict the financial situation of companies, reducing uncertainties and doubts in decision making [5]. The models are constructed with the support of statistical techniques and applied to analyze their dependent variables.

For the survival of financial institutions, the correct decision to grant credit is essential [6]. It is important to anticipate and reduce default [7], since the losses from unsuccessful credits should be covered by charging high interest rates on new concessions. Therefore, using a default risk forecasting model for a financial institution and linking management strategies to the reality of the borrower can be critical in assessing credit risk and reducing financial losses.

An information system, called DeRis [8], was developed aimed to support activities in the management of default risk. It encompasses a default prediction model based on conflict indicators, management, and financial indicators, a reasoner and visualization elements. Through the storage of decisions, a knowledge database is maintained. Thus, a significant amount of data must be collected, processed and stored over time, for proper monitoring of the indicators involved. DeRis offers this information through interactive visualizations, assisting the process of discovering knowledge through these data, aiming decision making.

Since its inception, the DeRis system has been validated, tested and applied to actual data, provided by a bank. This experience is reported in Lelis and Lopardi [8]. The bank that offered its data and became a partner of this research, will be called Zak bank for confidentiality issues.

Zak Bank focuses its activities on resource generation and credit analysis. It also seeks to meet the consumption and investment needs of individuals and companies. Considering the impact and risk of a customer becoming defaulted, it has become important for Zak Bank to monitor companies in the economic environment and manage a possible default.

There was a concern that the prediction model, used in DeRis system, specializes in the data provided by ZAK bank. As a consequence, the system could become static and fitted only to the reality of this context. To avoid this situation, the need to incorporate new model options into the system has increased. Moreover, it would also be necessary to submit the system to different datasets associated with different indicators.

In this scenario, the incorporation of the TensorFlow tool [9] could be a good opportunity for improvement for the DeRis system. It is an open source software library for high-performance digital computing. Its flexible architecture enables the easy deployment of computing across multiple processing unit platforms and desktops to server clusters, for mobile devices and peripherals. It comes with strong support for machine learning and deep learning, and the flexible numerical computing core is used in many other scientific domains. Given this, TensorFlow can offer more dynamism and speed to the DeRis system from its implementation in Python.

This article presents the improvements in DeRis system allowing models of the TensorFlow tool, and is structured by this introduction and Section 2 shows the background in which the proposal is inserted and some related work. Section 3 focuses on the improved components of the DeRis system. In Section 4, the carried out experiment is presented and Section 5 includes the final considerations.

## II. Background

Through a survey of the specialized technical literature, it was possible to perceive that the researchers' interest in default risk models dates back to the 1930s [10][11]. Over the years, the pioneering work of Beaver [12] and especially Altman [13], boosted research in the 1970s with accounting indicators [14]–[18].

Changes in the world financial scenario since the 1990s, such as deregulation of interest rates and exchange rates, increased liquidity and increased competitiveness, especially in the banking sector, have increased the concern of financial institutions with the risk of default. Issues such as the emergence of new modeling techniques, the growing importance of credit risk management and the prevailing economic conditions, again aroused the interest in the area [19]–[22].

There are several techniques applied to credit risk forecasting models. They can be classified as discriminant analysis used in the model proposed by [13], neural networks as used by Lemos et al. [23] and replicated in the present paper.

Considering that a neural network has been replicated, it is necessary to present some basic concepts. Neural networks try to build internal representations of models or patterns detected in the data, which are generally not visible to the user. Neural Networks use a set of processing elements (nodes) analogous to neurons. These processing elements are interconnected in a network that can identify patterns in the data, that is, the network learns through experience, such as people. Existing Neural Networks models present one or more layers of neurons between the input and output layers, called hidden layers [24]. In these networks, each layer has a specific function. The output layer receives the results from the hidden layer and generates the final response. The network is formed by connecting the output of the neurons from the hidden layer to the input of the neurons of the output layer. The resulting structure is a weighted and directed graph. The weights as well as the functions that compute the internal state of a neuron (activation) can be modified by a process, called learning. This process is governed by a learning rule.

There are other techniques like multiple linear regression, linear programming, genetic algorithms, decision tree, logistic regression applied on the core model of DeRis system and, more recently, the analysis of survival.

Bonfim [25], for example, examined the determinants of corporate defaults in the banking sector in Portugal through the Logit or Probite Models of Survival Analysis. The study found that default is affected by specific characteristics of companies such as: capital structure, company size, profitability and liquidity, recent sales performance and investment policy. However, there was a significant improvement in the quality of the models, with the introduction of variables, especially the growth rate of all the riches produced in the country, the growth of lending, the average lending rate and the variation of stock market prices.

Bellovary et al. [26] investigated the main financial indicators used in studies to predict default and found the current liquidity present in 51 studies among those analyzed.

Years later, Jacobson et al. [27] presented a model based on macroeconomic factors. The nominal interest rate and the output gap are the two most important macroeconomic factors that affect corporate default. The authors also used the Earnings Before Interest, Taxes, Depreciation and Amortization (EBITDA) per Total Assets ratio, the interest coverage ratio, the leverage ratio, the total liability ratio and revenues, the ratio net assets and total liabilities and, finally, inventory turnover, as financial variables specific to each company.

Recently, Cunha et al. [28] carried out a national and an international survey where they investigate which features are required to enhance a credit scoring model in the context of a Brazilian retail enterprise, in order to find attributes that can improve the performance of classifier algorithms for credit granting. They conclude that additional financial and behavioral data increase defaulting prediction performance on credit granting.

From the identified models, as well as the concepts of credit and risk, it should be noted that the default forecast models, mostly, have financial indicators as explanatory variables. Therefore, creating a model without taking such indicators into account would put its effectiveness in question.

## III. DeRis System

The proposed DeRis system arose from the need to predict the financial situation of companies to avoid default, as well as support managers and financial institutions in making decisions regarding the granting of credit.

The details of the system, as well as the features and data flow between each of the components were detailed in [8]. For this reason, only an overview will be presented. Then we will focus on improvements made with TensorFlow.

Figure 1 shows an overview of the DeRis system architecture, after the improvement process with its main components.

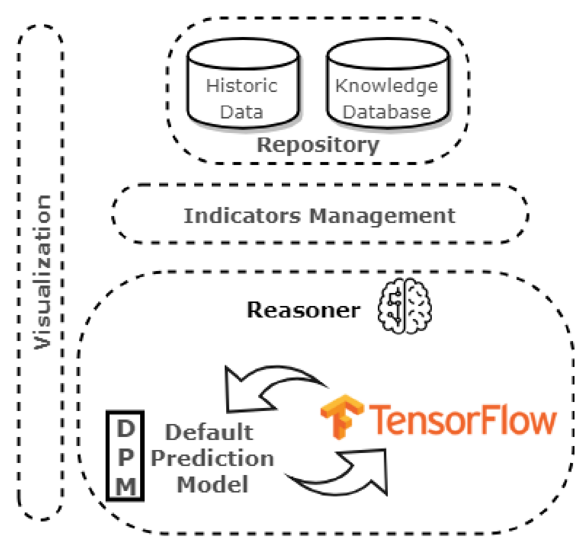


Figure 1. Improved DeRis Archtecture.

The information flow, while using the system, starts in the repository that stores the historical data of the indicators. Additional information, such as a brief description and the indicator classification are also stored. Indicator management is performed to determine which indicators will be used in the selected prediction model. This choice is made considering the possibility of calculating the indicator with the available data.

The machine learning module is triggered and the selection of the default forecast model is enabled by the system. Reasoner gives managers the gain of information between raw data and model results. Moreover, Reasoner helps managers interpret the information generated during the process. Finally, the visualizations show the results and analyzes made by Reasoner and, through interaction elements, managers can associate control measures with the indicators.

The decisions taken during the process of using the system, as well as the information generated, are stored in a knowledge database, in order to feed the system and support future decisions.

The DeRis system presented a single model, at its core, based on Logistic Regression. In the construction and training phase of the model, the insertion or removal of some indicator can generate different results. Even the logistic regression model may present changes in the result. A new feature added to the system allows the storage of the decision, the indicators with their values, as well as the generated model. This measure aims to adjust the choice of indicators with the models provided by the TensorFlow tool, achieving better results.

### A. Indicators Management

This is an important component of the system. It identifies the indicators whose data is stored, its additional information and mainly the classification that assists the Reasoner in the analyzes. Through these data, a feature of this component is to relate the indicator to the way of collecting, or calculating, its value.

All stored indicators are candidates to participate in the model. It is necessary to make the selection of the indicators when starting the monitoring of a company, or through previous decisions, the own component is capable of selects the indicators.

### B. Visualization

The views offered by the DeRis system can now be triggered from any component of the system, and support attributes that can be measured in real time.

Through interaction elements, it is possible to select a point in the line graph and obtain contextual information, such as: the future trend of the default probability, the values of each indicator, the model used and the decision taken at the time, if any. Moreover, the percentage of each indicator, colored according to the variation to the previous occurrence. This feature allows the analysis of the variability in the influence of indicators.

### C. Machine Learning Module

The components responsible for intelligence, knowledge discovery and information interpretation have been aggregated and now integrate the Machine Learning Module along with the TensorFlow tool.

*1) Reasoner Phase:* Assuming that there are indicators A, B and C. However, only the data for indicators A and C are available. Given this, *would it be possible to replace B? Which indicator could replace it?* Questions like these that Reasoner tries to answer with their analysis.

These questions can be answered by the Reasoner due to information such as: the class to which the indicator belongs, the unit of indicator measure, its degree of influence on the default and decisions taken previously after the exchange of indicators.

Another important function is to relate a moment of the past with a description of the decision made and what were the critical indicators for default, based on historical data and the knowledge base.

Although the review process is transparent to the user, the decision to replace an indicator is performed by the manager, when necessary, whenever a company's monitoring begins.

With the new version of the system, Reasoner has acquired a new feature. Before the data is stored in the knowledge base, it is responsible for mapping between the attributes with their values and the chosen model with its execution parameters.

*2) Default Prediction Model:* This is the legacy core model based on conflict, management and financial indicators, classified in the indicators management stage.

The technique applied by the model is logistic regression [29] that allows analyzing the effect of one or more independent variables on a dichotomous dependent variable, representing the presence or absence of a characteristic. In this way, it describes the relationship between several independent variables. According to this theory, the model calculates the probability of default, given by (1):

$$ProbDefault(yes) = \frac{e^{\eta}}{1 + e^{\eta}} \tag{1}$$

Where $\eta$ depends on the indicators and data available for the logistic regression calculation.

*3) TensorFlow:* TensorFlow is a tool for machine learning with vast majority built in Python. It contains a wide range of functionality and provides many APIs. An important one is the Estimator API which provide scalable, high-performance models. Working as an interface for creating and executing machine learning algorithms.

Considering that TensorFlow is designed primarily for deep neural network models, it is necessary to analyze the hyperparameters that can be used in the training process to tune the model.

During training, the train method usually processes the examples several times. In addition, training works best if the training examples are in random order. Therefore, it is good practice to ensure that the data will be well scrambled.

The train method processes a batch of examples at a time and sets the default batch size to 100, which means that the batch method will concatenate groups of 100 samples. The ideal lot size depends on the problem. As a general rule, smaller batch sizes often allow the method to train the model more quickly at (or sometimes even) expense of accuracy.

The steps argument tells train to stop training after the specified number of iterations. Increasing steps increases the amount of time the model will train. Counter-intuitively, training a model longer does not guarantee a better model. The

 ISBN: 978-1-61208-651-4 

number of steps to train is a hyperparameter that can be tuned. Choosing the right number of steps usually requires both experience and experimentation.

In the next section, there will be shown the evaluation in which these parameters could be tested in different configurations.

## IV. EVALUATION

This Section presents the experimental study conducted. According to the Goal/Question/Metric approach (GQM) [30] the goal can be stated as: **Analyze** the DeRis system **in order to** verify the feasibility of using the improvements made **with respect to** the implementation of the TensorFlow tool offering option to the default forecast model **from the point of view of** managers and professionals of financial institutions **in the context of** credit analysis and default risk.

In this sense, the metrics defined to verify the fit quality of the models were the mean error and mean accuracy, like used by [23].

The experiment was proposed based on a set of real data collected from documentary sources by [23]. The Dataset was chosen because it presents attributes different from those previously studied in the DeRis system. In addition, because it is available for access and have the application parameters and methodology explained in order to facilitate the replication attempt.

Therefore, in this work, the historical data of 339 corporate clients were used, including micro, small and medium-sized enterprises, of which 73 are defaulters and 266 debt free companies. From each of them, 24 information (between cadastral and company accounting) were extracted, which will be specified below.

- **Restrictions on behalf of the company:** Represents the existence of restrictions and can be categorized between YES or NO.
- **Restrictions lowered in the last five years on behalf of the company:** Represents the existence of lowered restrictions and can be categorized between YES or NO.
- **Time of account:** Defined as a numeric value in Months.
- **Sector of activity:** Defined as a category between between Trade (1), Industry (2) or Services (3).
- **Uptime:** Categorized in sets of years where: More than 9 years (1), From 6 to 9 years (2), From 3 to 5 years (3), From 1 to 2 years (4) and Less than 1 year (5).
- **Number of employees:** Defined as a numeric value.
- **Company headquarters (property):** Categorized between Own (1), Rented (2) or Provided (3)
- **Neighborhood:** Categorized between Downtown (1) or Other (2).
- **Main customers:** Categorized between Individuals (1), Companies (2) or Mixed (3).
- **Annual gross sales:** represented by a numeric value.
- **Customer in another bank:** Indicates if the company is also customer in another borrower and can be categorized between YES or NO.
- **Real estate:** Defined as a numeric value.
- **Movable property:** Defined as a numerical value.
- **Business insurance:** can be categorized between YES or NO.
- **Financial applications:** can be categorized as Greater than 8,000 (1), From 8,000 to 4,000 (2), From 4,000 to 2,000 (3), Less than 2,000 (4) and no applications (5)
- **Term sales:** can be categorized as Less than 20% (1) or More than 20% (2).
- **Credit experience:** can be categorized as Greater than 2 years (1), Less than 2 years (2) or No experience (3)
- **Account history:** can be categorized as Normal (1), Checks returned (2), New customer (3), Small frequent delays (4)
- **Company members have restrictions:** it can be categorized between YES or NO.
- **Members of the company had restrictions lowered in the last five years:** it can be categorized between YES or NO.
- **Partnership between spouses:** It indicates the existence of society and can be categorized between YES or NO.
- **Real estate on behalf of partners:** Defined as a numerical value.
- **Movable property on behalf of the partners:** Defined as a numerical value.
- **Assigned risk:** a concept defined by the borrower, which stipulates the minimum guarantees required in credit operations. Categorized into levels by a scale where, 1 is the best and 5 is the worst concept.

### A. Analysis

In this section, we present the replication made from the data and the methodology adopted. As well as an optimization in relation to the parameters that allow to tune the model. And, finally, a test with a balanced data set. A comparison between them with regard to their performances in obtaining the results are discussed and presented at the end.

*1) Replication:* Eight sets of tests were performed in total. The first test included information from all 339 companies. In the others, the data were separated into two sets: a training with data from 306 companies, composed of 241 non-defaulting companies and 65 defaulters, and another set for testing with information from 33 companies, composed of 25 non-defaulting companies and 8 defaulters. Except in the first case, in each of the tests performed, the sets were randomly generated in order to avoid any kind of induction of results. The difference between each test was the number of times the samples were shuffled. For the first test were shuffled once to the second they were shuffled twice each set of data and so on until the eighth test.

The training was carried out through a network of multiple layers, varying the following parameters:

- **Number of iterations (steps):** in each set of tests, the Neural Network was trained with 100, 1,000, 2,000, 4,000, 6,000, 8,000 and 10,000 iterations;
- **Number of intermediate neurons in the network:** in each test performed, the Neural Network was trained first without the intermediate layer and then using 2, 4, 6, 8 and 10 neurons in the intermediate layer. For


ISBN: 978-1-61208-651-4

each test performed, a random set of initial weights was used;

- **Learning rate:** constant equal to 0.01;
- **Momentum rate:** It was decided not to use it.

Considering in this replication the random characteristic in the selection of the datasets this indicates that it is not possible to say if the same samples are in the same sets used by Lemos et al. [23]. For this reason, the results found in each set of tests were aggregated by the mean of the accuracy and are depicted in Table I.

TABLE I. REPLICATION RESULTS

| Mean accuracy of | Lemos et al. [23] | Replication |
|---|---|---|
| Train set | 95.91% | 96.58% |
| Test set | 90.04% | 83.12% |

The analysis shows an accuracy gain of 0.67 in the replicate model training set. In relation to the test set, the replicated model presented a difference in the accuracy of 6.92 of the results presented by Lemos et al. [23]. After such observations, it was necessary to analyze if there is statistical relevance in the variations found by the means and, furthermore, a hypothesis test was conducted.

Initially, the Shapiro-Wilk normality test was performed on the results obtained by Lemos et al. [23] and then, on the replication results. Considering that the data had no normal distribution, a non-parametric test was applied. The Mann-Whitney test, at a significance level of 5%, obtained a p-value of 0.533. Thus, the null hypothesis was accepted that there are no statistically significant differences for the samples. Therefore, the means are statistically equivalent.

*2) Optimization:* In order to verify the influence of the hyperparameters in the model, an optimization was proposed from the same sets generated for the replication and the same shuffle strategy. The training was carried out varying the following parameters:

- **Number of iterations (steps):** in each set of tests, the Neural Network was trained up to 15,000 iterations;
- **Number of hidden layers:** in each test performed, the Neural Network was trained first without a hidden layer and then using 1, 2 and 3 hidden layers, with the same number of neurons;
- **Number of intermediate neurons in the network:** the same as used for Replication;
- **Learning rate:** the same as used for Replication;
- **Momentum rate:** the same as used for Replication.

The results found in each set of tests were aggregated by the mean of the accuracy and are shown in Table II.

TABLE II. OPTIMIZATION RESULTS

| Mean accuracy of | Replication | Optimization |
|---|---|---|
| Train set | 96.58% | 99.71% |
| Test set | 83.12% | 85.71% |

*3) Balanced:* It is worth noting that there was a concern about the validity of the results found due to the fact that the sample was unbalanced, with 73 companies in default and 266 companies free of debt.

Therefore, it was decided to repeat the process with a new sample that was balanced and thus obtain a revalidation of the model. For the construction of this sample, all the defaulters were selected (73) and a random sample was made in the 266 so that 73 were selected. In order to guarantee the external and internal validity of the results found, a 8-steps of shuffle was used.

The results found in each set of tests were aggregated by the mean of the accuracy, compared with Replication and Optimization results and they are shown in Table III.

TABLE III. BALANCED SAMPLE RESULTS

| Mean accuracy of | Replication | Optimization | Balanced |
|---|---|---|---|
| Train set | 96.58% | 99.71% | 99.27% |
| Test set | 83.12% | 85.71% | 87.50% |

### B. Lessons Learned

Considering the improvement process applied to the DeRis system, the process of preparing the indicators and the implementation of the evaluations, some lessons learned should be highlighted.

Although evaluating the visualizations was not the goal of the experiment, it is worth noting that they fit the new indicators and data with success.

Data Interoperability: The DeRis system was originally implemented in Java. However, the operation of TensorFlow, based on Python language, was transparent to the end user.

During the evaluation process, the possibility of setting up and managing indicators of different types from those shown in [8] was verified through the Indicator Manager.

The manipulation and representation of the data was facilitated. Avoid changing the data type to do the representation. Just load the data into the model.

Optimizing models is possible through configuration variables. Such optimization reduces the error in accuracy. However, the task of finding a good setup is not trivial. It is worth remembering that during the evaluation, between the replication and optimization phase, 240 combinations were tested, from the variation of 4 parameters.

Although the goal is not to replace the decision maker, the improvement process conducted in the DeRis system is an important tool in the management of the generated knowledge. The decisions taken generate information that re-feed the entire system. At the same time it adds value by allowing different perceptions of the data through each model.

## V. CONCLUSION AND FUTURE WORK

The task of granting or not credit is and will always be difficult. Data Mining Techniques such as Neural Networks have proved to be very valuable tools for bank credit analysts. They become essential, combined with systems like DeRis and credit analyst experience.

An experimental study was conducted. The evaluation process was divided into three stages. Initially, the replication of the test applied by Lemos et al. [23] was successfully performed. In the following steps, only the results obtained by TensorFlow, through the DeRis system, were used for the comparison. Thus, the second step was to find the best values for the hyperparameters and obtain an optimized model with more accurate results. In the third step, the construction of a model trained from a balanced data set took place.

 

A threat to validity of the study is associated to the universe of variables previously defined and the particularities of the data under study. Even with a sign that is totally favorable to the granting of credit to a new customer, it may become a defaulting customer. Other factors, such as an accident (fire, theft or other) can interfere with the company's behavior in relation to commitments. An example like this is difficult to predict considering the analyzed data.

Although the implementation of the neural network was in a programming language different from that adopted by Lemos et al. [23], some precautions were taken to reduce the possibility of variation in the results obtained by replication and, as a consequence, reduce the effect of this possible threat to validity. Precautions such as the use of the same total data set, the same (random) selection strategy for the training set and the test set, as well as the same values of the network configuration parameters, considering the replication phase.

The results showed that the optimization was achieved with the adjustment of the hyperparameters offered by TensorFlow, with 240 different combinations being tested between these hyperparameters. Moreover, the model trained from a balanced set between classes obtained a better result even with a smaller number of instances. This provides opportunities for future research, for example, slicing the dataset to avoid models being trained with less representative instances. Although the results are associated with the data and the design of the experiment conducted, they were considered positive and promising. Therefore, the feasibility to apply TensorFlow in the context of DeRis was verified and, furthermore, it was possible to show that DeRis was prepared to deal with distinct indicators from those studied in [8].

As a contribution of this work, it was presented an application of TensorFlow tool in DeRis system that offers financial institutions an approach to encourage the use of different types of data and indicators, in the search for continuous improvement. Considering the experiment process, the replication carried out, in the first stage of the evaluation, can be considered a contribution of the present study since it increases the external validity of the tests conducted by Lemos et al. [23].

As future work we intend to use the system in a larger set of data to analyze its efficiency in the context of Big Data. In addition, expand the application of the system to other types of financial problems such as business bankruptcy, profit forecasting and value to be granted on credit.

## REFERENCES


[1] C. G. Bernardo, "Loan system in brazilian financial institution-a soa application," in Information Technology: New Generations (ITNG), 2012 Ninth International Conference on. IEEE, 2012, pp. 293–298.

[2] D. Duffie and K. J. Singleton, Credit risk: pricing, measurement, and management. Princeton University Press, 2012.

[3] S. Westgaard and N. Van der Wijst, "Default probabilities in a corporate bank portfolio: A logistic model approach," European journal of operational research, vol. 135, no. 2, 2001, pp. 338–349.

[4] J. Bessis, Risk management in banking. John Wiley & Sons, 2011.

[5] M. T. Dugan and C. V. Zavgren, "How a bankruptcy model could be incorporated as an analytic," The CPA Journal, vol. 59, no. 5, 1989, p. 64.

[6] L. J. Gitman, R. Juchau, and J. Flanagan, Principles of managerial finance. Pearson Higher Education AU, 2015.

[7] M. Steiner, C. Carnieri, B. Kopittke, and P. S. Neto, "Probabilistic expert systems and neural networks in bank credit analysis," International Journal Of Operations And Quantitative Management, vol. 6, no. 4, 2000, pp. 235–250.

[8] C. A. S. Lélis and A. L. S. Lopardi, "Deris: Information system supporting the prediction of default risk in companies," in Information Technology-New Generations. Springer, 2018, pp. 325–332.

[9] M. Abadi et al., "Tensorflow: A system for large-scale machine learning." in OSDI, vol. 16, 2016, pp. 265–283.

[10] P. J. FitzPatrick, "A comparison of the ratios of successful industrial enterprises with those of failed companies," Certified Public Accountant, vol. 12, 1932, pp. 598–605.

[11] R. Smith and A. Winakor, Changes in the financial structure of unsuccessful industrial corporations. University of Illinois: Bureau of Business Research, 1935.

[12] W. H. Beaver, "Financial ratios as predictors of failure," Journal of accounting research, 1966, pp. 71–111.

[13] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," The journal of finance, vol. 23, no. 4, 1968, pp. 589–609.

[14] R. O. Edmister, "An empirical test of financial ratio analysis for small business failure prediction," Journal of Financial and Quantitative analysis, vol. 7, no. 2, 1972, pp. 1477–1493.

[15] E. B. Deakin, "A discriminant analysis of predictors of business failure," Journal of accounting research, 1972, pp. 167–179.

[16] M. Blum, "Failing company discriminant analysis," Journal of accounting research, 1974, pp. 1–25.

[17] R. A. Eisenbeis, "Pitfalls in the application of discriminant analysis in business, finance, and economics," The Journal of Finance, vol. 32, no. 3, 1977, pp. 875–900.

[18] R. C. Moyer, "Forecasting financial failure: a re-examination," Financial Management, vol. 6, no. 1, 1977, p. 11.

[19] E. I. Altman, G. Marco, and F. Varetto, "Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the italian experience)," Journal of banking & finance, vol. 18, no. 3, 1994, pp. 505–529.

[20] P. Brockett, W. Cooper, L. Golden, and X. Xia, "A case study in applying neural networks to predicting insolvency for property and casualty insurers," Journal of the Operational Research Society, 1997, pp. 1153–1162.

[21] C. Y. Shirata, "Financial ratios as predictors of bankruptcy in japan: an empirical research," Tsukuba College of Technology Japan, vol. 1, no. 1, 1998, pp. 1–17.

[22] C. Lennox, "Identifying failing companies: a re-evaluation of the logit, probit and da approaches," Journal of economics and Business, vol. 51, no. 4, 1999, pp. 347–364.

[23] E. Prezepiorski Lemos, M. T. Arns Steiner, and J. C. Nievola, "Análise de crédito bancário por meio de redes neurais e árvores de decisão: uma aplicação simples de data mining. alternative title: Analysis of bank credit through neural networks and decision trees: a simple application of data mining," RAUSP, vol. 40, no. 3, 2005, pp. 225–234.

[24] J. A. Anderson, An introduction to neural networks. MIT press, 1995.

[25] D. Bonfim, "Credit risk drivers: Evaluating the contribution of firm level information and of macroeconomic dynamics," Journal of Banking & Finance, vol. 33, no. 2, 2009, pp. 281–299.

[26] J. L. Bellovary, D. E. Giacomino, and M. D. Akers, "A review of bankruptcy prediction studies: 1930 to present," Journal of Financial education, 2007, pp. 1–42.

[27] T. Jacobson, J. Lindé, and K. Roszbach, "Firm default and aggregate fluctuations," Journal of the European Economic Association, vol. 11, no. 4, 2013, pp. 945–972.

[28] S. C. Cunha, E. M. Carneiro, L. F. S. Mialaret, L. A. V. Dias, and A. M. da Cunha, "Investigating attribute assessment for credit granting on a brazilian retail enterprise," in Information Technology-New Generations. Springer, 2018, pp. 301–309.

[29] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, Applied logistic regression. John Wiley & Sons, 2014, vol. 398.

[30] V. R. Basili and D. M. Weiss, "A methodology for collecting valid software engineering data," IEEE Transactions on Software Engineering, vol. SE-10, no. 6, 1984, pp. 728–738.