Applied Soft Computing Journal 91 (2020) 106263

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Statistical and machine learning models in credit scoring: A systematic literature survey

Xolani Dastile [a,*], Turgay Celik [a,b], Moshe Potsane [a]

[a] School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, South Africa
[b] Wits Institute of Data Science, University of the Witwatersrand, Johannesburg, South Africa

ARTICLE INFO

Article history:
Received 3 June 2019
Received in revised form 23 February 2020
Accepted 19 March 2020
Available online 25 March 2020

Keywords:
Credit scoring
Statistical learning
Machine learning
Deep learning
Systematic literature survey

ABSTRACT

In practice, as a well-known statistical method, the logistic regression model is used to evaluate the credit-worthiness of borrowers due to its simplicity and transparency in predictions. However, in literature, sophisticated machine learning models can be found that can replace the logistic regression model. Despite the advances and applications of machine learning models in credit scoring, there are still two major issues: the incapability of some of the machine learning models to explain predictions; and the issue of imbalanced datasets. As such, there is a need for a thorough survey of recent literature in credit scoring. This article employs a systematic literature survey approach to systematically review statistical and machine learning models in credit scoring, to identify limitations in literature, to propose a guiding machine learning framework, and to point to emerging directions. This literature survey is based on 74 primary studies, such as journal and conference articles, that were published between 2010 and 2018. According to the meta-analysis of this literature survey, we found that in general, an ensemble of classifiers performs better than single classifiers. Although deep learning models have not been applied extensively in credit scoring literature, they show promising results.

1. Introduction

The history of credit scoring dates back to the 1950s [1]. During those early years, credit decisions were made using a judgmental approach commonly known as 5C's approach [2]:

> Character: do you know the person or their family?
> Capital: how much is being asked for?
> Collateral: what is the borrower willing to put up from their resources?
> Capacity: what is their repayment ability?
> Condition: what are the conditions in the market?

The drawback of using the 5C's approach was the incapability of processing a huge number of applications daily. This resulted in the advent of scorecards, which make consistent nonjudgmental decisions that treat all borrowers fairly. The scorecards generate a score which quantifies the risk of lending money to borrowers.

When a borrower applies for a loan, the financial institution, without losing generality hereafter referred to as the bank, collects information from the borrower. This information is called application data and it consists of *demographic information*, e.g., the number of dependents, time at current address, time at current employment, etc. The *bureau information* is also collected from the local bureaus, and it includes the number of inquiries, judgments, number of delinquencies, etc. Once the accepted population, i.e. borrowers who have been granted loans, are identified, their loan repayment history is tracked for a period of time, e.g. 24-months. A *target flag* (i.e. good/bad flag) gets created based on loan repayment history of the accepted population. If the number of days past due (or missed payments) is less than a certain number of days, e.g. 90 days, then the borrower is flagged as a *good* borrower, otherwise a *bad* borrower. The known goods and bads are then used to develop a scorecard. The cut-off score is determined by the *Kolmogorov–Smirnov statistic*, and it measures the distance between the cumulative distribution of goods and bads. The score which gives the maximum distance between the distribution of goods and bads is regarded as the cut-off score and is used to predict the goods and the bads. If a score of a borrower is larger than or equal to the cut-off score, then the borrower is predicted as a good borrower otherwise a bad borrower. The scorecard is then applied to the rejected population to predict goods and bads and this is referred to as *reject inference* [3]. The final application scorecard is built on the accepted and rejected populations, i.e. Through-The-Door (TTD) population. Please see Fig. 1 for a schematic view of data flowchart for application scorecards. Traditionally, financial institutions use a logistic regression to score borrowers. The choice of using the logistic regression

* Corresponding author.
E-mail addresses: xdastile12@gmail.com (X. Dastile), celikturgay@gmail.com (T. Celik), Moshe.Potsane@wesbank.co.za (M. Potsane).

https://doi.org/10.1016/j.asoc.2020.106263
1568-4946/

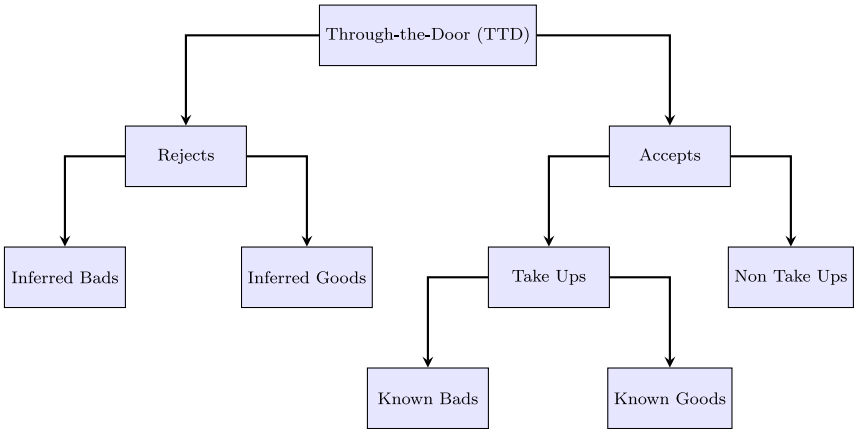

Fig. 1. Schematic view of data flowchart for application scorecards.

model is due to its simplicity and transparency. The scores from logistic regression are calculated using the following formula [3]

$$Score = \frac{pdo}{\ln(2)} \times \left(\beta_0 + \sum_{i=1}^{n} \beta_i \sum_{j=1}^{k_i} WOE_{i,j}\right), \tag{1}$$

where

$$WOE_{i,j} = \ln\left(\frac{F_{G_j}}{F_{B_j}}\right) \tag{2}$$

and n is the number of features, k_i is the number of groups or attributes in the ith feature, pdo is points double the odds and is used as a scaling factor and F_{G_j} and F_{B_j} are distribution of good borrowers and bad borrowers in the jth attribute in feature i, respectively. The β_i coefficients are estimated using *Maximum Likelihood Estimation* [4].

In literature, sophisticated machine learning (ML) models can be found that can replace the logistic regression model. In spite of high accuracies from ML models; ML models are generally unable to explain their predictions. Financial institutions are regulated entities and are required to be transparent in their decisions when using application scorecards. However, techniques that can deduce rules to mitigate the lack of transparency without even compromising accuracy are suggested in the literature, e.g. [5]. In this paper, we aim to present a systematic literature survey of statistical and machine learning models which are employed in credit scoring between 2010 and 2018 and propose a guiding ML framework for credit scoring. The remainder of this paper is organized as follows. In Section 2 we cover the methodology followed for conducting this systematic literature survey. In Section 3 we highlight feature selection/engineering methods. The learning models are covered in Section 4. In Section 5 we present evaluation metrics. The data imbalance is covered in Section 6. In Section 7 we cover model transparency. In Section 8 we discuss limitations and assumptions of different models. In Section 9 we discuss emerging trends. Section 10 discusses limitations in literature. In Section 11 we discuss results. In Section 12 we propose a guiding framework for machine learning in credit scoring and finally Section 13 provides conclusion and future work.

2. Survey methodology

This study employs a systematic literature survey approach to

(1) systematically review the most commonly used statistical and machine learning techniques in credit scoring;
(2) identify limitations in literature;
(3) propose a guiding machine learning framework to perform credit scoring;
(4) point to emerging directions.

In this survey, the statistical techniques considered include Linear Discriminant Analysis (LDA), Logistic Regression (LR) and Naïve Bayes (NB), and the machine learning include k-Nearest Neighbor (k-NN), Decision Trees (DTs), Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), Random Forests (RFs), Boosting, Extreme Gradient Boost (XGBoost), Bagging, Restricted Boltzmann Machines (RBMs), Deep Multi-Layer Perceptron (DMLP), Convolutional Neural Networks (CNNs) and Deep Belief Neural Networks (DBNs). Note that the list of these techniques for both statistical and machine learning is not exhaustive, the literature provides a myriad of techniques that are used to model credit scoring. However, this study only reviews the most commonly used techniques as it would be almost impossible to look at all techniques applied in credit scoring.

A process flowchart of the systematic literature survey methodology is presented in Fig. 2. Note that this study follows the methodology of the recently published systematic literature survey on bankruptcy prediction models [6]. The search words of the current study are "Statistical Learning in Credit Scoring", "Machine Learning in Credit Scoring" and "Deep Learning in Credit Scoring". The inclusion criterion for articles is based on the recently published work in credit scoring and the period considered is the year 2010 to the year 2018. The study selects peer-reviewed journal and conference articles as these are considered to be of high quality [7]. The articles are selected by reading the abstract and the conclusion, in some instances the entire article is read. This review is based on articles that are written in English only. This overlooks one of the requirements in systematic literature review where language constraints are discouraged. The following databases are used in searching papers: Google Scholar, Science Direct, IEEEXplore, ACM and SpringerLink. These databases include studies which are undertaken all over the world, hence geographical bias is removed.

All unpublished work and dissertations are not included in this current study. In the end, 74 primary studies were selected for this systematic literature survey. The primary studies include models in their hybrid form (i.e. where feature selection/feature engineering is combined with a classifier or ensemble classifiers) and in their standalone form. A meta-analysis of the results from the selected articles is done by producing summary tables, pie-charts and histograms. The German and Australian credit datasets are the most frequently used datasets in credit scoring, hence we used these two datasets for model performance comparisons.

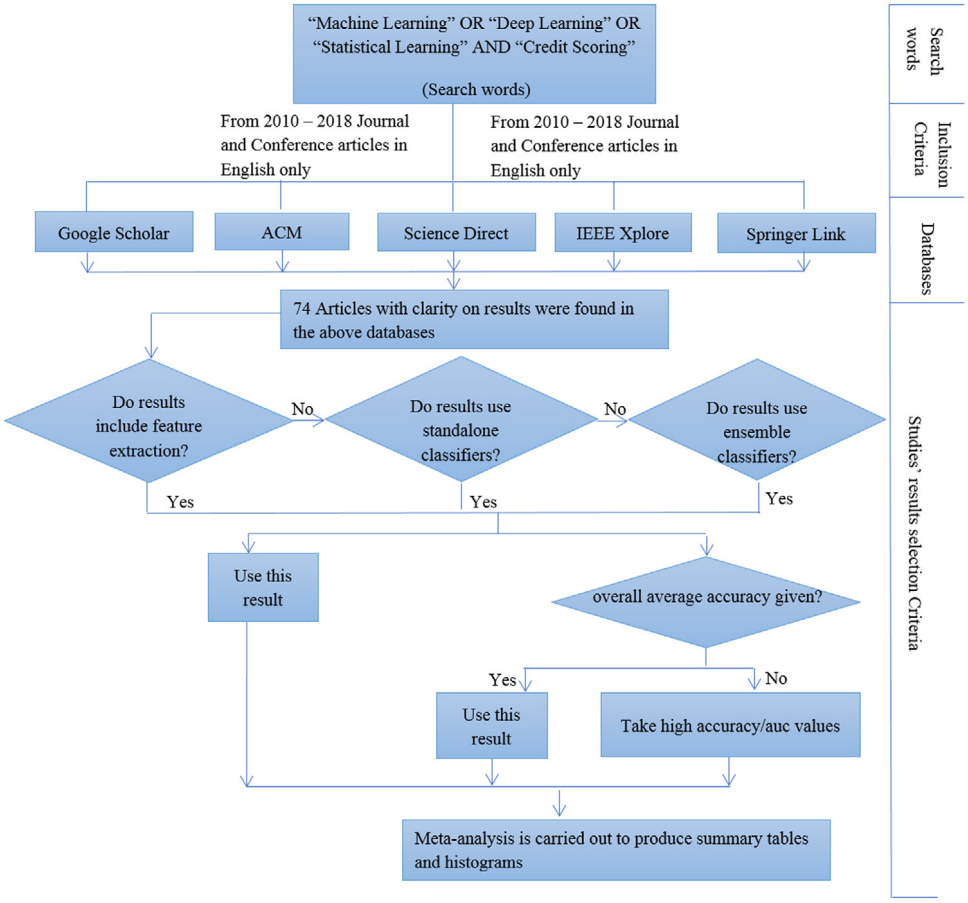

Fig. 2. Process flowchart of the systematic literature survey methodology.

2.1. Existing literature surveys on bankruptcy prediction or credit scoring

There are several literature surveys on bankruptcy prediction or credit scoring [6,8–13]. We briefly highlight the objectives of each literature survey and show where they differ with our study in Table 1. The literature surveys in Table 1 focus mostly on comparing models based on their performances. However, the listed literature surveys ignore key factors in credit scoring, such as model transparency and the nature of datasets. Furthermore, deep learning models are not considered in the listed literature surveys in Table 1.

Among the existing literature surveys shown in Table 1, [12] provide the most recent and the most comprehensive systematic literature survey in credit scoring. The main objective of [12] is "proposing a new method for rating, comparing traditional techniques, conceptual discussions, feature selection, literature review, performance measures studies and, at last, other issues". Their literature survey covers a period from January 1992 to December 2015 and focuses on 12 questions on the conceptual scenario over the techniques. Comparing [12] with our literature survey, there is an overlap from 2010 to 2015. However, our literature survey includes the most recent years. According to the best of our knowledge, there is no systematic literature survey for credit scoring that has been published which encompasses, statistical learning, traditional machine learning and deep learning models.

3. Feature selection and feature engineering

This section provides a review of most commonly used *feature selection* (FS) and *feature engineering* (FE) techniques used in credit scoring. The distinction between feature selection and feature engineering is that the feature selection selects a subset of features from the entire feature set whereas feature engineering creates a new set of features from the existing features. Albeit, Liang et al. [14] investigated the effect of feature selection and concluded that performing feature selection does not always improve the prediction performance, the studies in [15–19] showed that removing redundant features can improve model performance in credit scoring. There are also studies [20–22] which employed meta-heuristic approach such as *Genetic Algorithm* for feature selection. We divided the feature selection into *filter*, *wrapper* and *embedded* methods.

3.1. Filter methods

The filter methods perform feature selection based on the univariate analysis of the features without using a predictor. They compute a score for each feature and select a subset of features based on their scores. In the following, we present most commonly used filter methods in credit scoring.

3.1.1. F-score

F-score measures the discrimination of two sets of real numbers [17]. Given m training vectors $\mathbf{x}_k$, where $k = 1, 2, \ldots, m$, and if the number of positive and negative instances are $m^{(+1)}$ and $m^{(-1)}$, respectively, then the F-score of the ith feature $F(i)$ is defined as follows

$$F(i) = \frac{(\bar{x}_i^{(+1)} - \bar{x}_i)^2 + (\bar{x}_i^{(-1)} - \bar{x}_i)^2}{\frac{1}{m^{(+1)}-1}\sum_{k=1}^{m^{(+1)}}(x_{k,i}^{(+1)} - \bar{x}_i^{(+1)})^2 + \frac{1}{m^{(-1)}-1}\sum_{k=1}^{m^{(-1)}}(x_{k,i}^{(-1)} - \bar{x}_i^{(-1)})^2} \quad (3)$$

Table 1
Existing literature surveys on bankruptcy prediction or credit scoring and their differences from this survey paper.

Survey paper	Articles searched	Years	Objective	Difference from this survey
[8]	165	1965–2010	✓ Investigation of model type by decade ✓ Analysis of Number of features used by decade ✓ Compare model performance	✓ Transparency is not reported ✓ Deep learning models are not covered ✓ Nature of datasets (balanced vs imbalanced) is not reported
[9]	214	Not specified	✓ Comparing models based on performance	✓ Transparency is not reported ✓ Deep learning models are not covered ✓ Nature of datasets (balanced vs imbalanced) is not reported
[10]	130	1995–2010	✓ Models are compared based on design, datasets and baselines	✓ Transparency is not covered ✓ Deep learning models are not covered ✓ Nature of datasets (balanced vs imbalanced) is not reported
[11]	Not specified	Not specified	✓ Comparing models based on performance	✓ Transparency is not reported ✓ Deep learning models are not covered ✓ Nature of datasets (balanced vs imbalanced) is not reported
[12]	187	1992–2015	✓ Proposing a new method for scoring ✓ Compare traditional techniques ✓ Conceptual discussion	✓ Transparency is not reported ✓ Deep learning models are not covered ✓ Nature of datasets (balanced vs imbalanced) is not reported
[13]	6	Not specified	✓ Compare model performance	✓ Transparency is not reported ✓ Deep learning models are not covered ✓ Nature of datasets (balanced vs imbalanced) is not reported
[6]	49	2010–2015	✓ Compare models based on accuracy, transparency, data size capability, data dispersion and etc.	✓ Deep learning models are not covered ✓ Nature of datasets (balanced vs imbalanced) is not reported

where $x_{k,i}$ is the ith feature of the kth sample, $\bar{x}_i$, $\bar{x}_i^{(+1)}$ and $\bar{x}_i^{(-1)}$ are the averages of the ith feature of the whole, positive, and negative datasets, respectively [17]. The numerator indicates the discrimination between the positive and negative sets, and the denominator indicates the one within each of the two sets. The larger the F-score value is, the more discriminative power the feature has [17,23].

3.1.2. Rough set theory

In rough set theory, the sample dataset is called information system, denoted by $I = (U, A)$, where U is a non-empty finite set of observations called the universe and A is a non-empty finite set of features. Firstly, a notion of *indiscernibility relation* on a universe set is defined. With every subset $B \subseteq A$, an indiscernibility relation on U is defined as follows

$$I(B) = \{(\mathbf{x}, \mathbf{y}) \in U \times U : f_i(\mathbf{x}) = f_i(\mathbf{y}), \forall i \in B\}, \tag{4}$$

where $f_i : U \rightarrow V_i$ is the ith feature function and V_i is a set of values associated with feature i. Feature i is redundant in B if $I(B) = I(B - \{i\})$, otherwise feature i is important in B. If all of the features in B are important, then B is said to be a reduced set of features [24–26].

3.2. Wrapper methods

The wrapper methods select a subset of features based on an evaluation metric, such as accuracy, of a pre-determined predictor. Each subset of the features are scored according to their predictive power, thus, the wrapper methods are computationally expensive compared to the filter methods. However, since they select features based on the performance of the pre-determined predictor, the wrapper methods usually outperform filter methods in credit scoring. In the following, we present the most commonly used wrapper methods in credit scoring.

3.2.1. Stepwise selection

The stepwise feature selection has three components [3], namely, *forward feature selection*, *backward feature elimination* and a *stepwise feature selection* which is a combination of both forward feature selection and backward feature elimination. All three components use linear regression and a *p-value* for feature selection. The *forward feature selection* starts by regressing one feature and if the feature is significant according to the *p-value* then that feature is retained. This process is repeated by adding one feature at a time until the list of features is exhausted, and the significant features are retained and insignificant features are discarded. The *backward feature elimination* works the opposite way, you start with the entire set of features and you keep discarding features with *p-values* less than the chosen level of significance [27]. The *stepwise feature selection* adds and removes features.

3.2.2. Genetic algorithm

A genetic algorithm [28] uses genetic inspired operators to evolve an initial population into a new population. These operators are the *selection*, *crossover* and *mutation*. Each population comprises of chromosomes (a string of bits (0/1)) that represent genetically encoded individual solutions to a specific problem. Selection operator selects chromosomes in the population (a set of solutions) for reproduction. Crossover operator randomly chooses a position and exchanges the subsequent bits before and after that position between two chromosomes to create two offsprings. Mutation operator randomly flips some of the bits in a chromosome. Each individual has a fitness score assigned to them, which represents its ability of be selected. For feature selection, the individuals are subsets of features that are encoded as bits where the ith feature is selected if the corresponding bit is set to 0. The fitness value is some measure of model performance, such as classification accuracy. A new population is evolved by using operators of crossover, where selection is based on individual's fitness function and its ability to reproduce the next

generation [29]. Hence, genetic algorithm is an evolutionary algorithm problem that is solved by searching a space (in our case a space of features). However, the key elements of problem solving by search are *exploitation* and *exploration*. The exploitation uses the information from previously visited points to determine the prospect of finding profitable regions to be visited next and the exploration is the process of visiting new regions of a search space to uncover promising offsprings [30]. "How and when to control and balance exploration and exploitation in the search process to obtain even better fitness results and/or convergence faster are still on-going research" [31]. It is key that a diversified population is maintained during the whole search process. The measure of diversity is *entropy* and it represents the amount of population disorder, where an increase in entropy results in an increase in diversity [30].

3.3. Embedded methods

The embedded methods optimize an objective function or learning classifier with a goodness-of-fit term and a penalty for a large number of features [32]. The most commonly used embedded method in credit scoring is the least absolute shrinkage and selection operator (LASSO) [33]. Given a linear regression model to predict dependent variable y_i using independent variables $x_{i,j}$s for ith sample in the training dataset, i.e.

$$y_i = \sum_{i=1}^{n} x_{i,j}\beta_j + \beta_0, \tag{5}$$

where $j = 1, 2, \ldots, n$, the LASSO [33] solves the $L1$-penalized regression problem of finding the set of parameters $\beta = \{\beta_j\}_{j=1}^{n}$ to minimize

$$\sum_{i=1}^{m}\Big(y_i - \sum_{j} x_{i,j}\beta_j\Big)^2 + \lambda \sum_{j=1}^{n} |\beta_j|, \tag{6}$$

where m is the number of instances in the training dataset and $\lambda \sum_{j=1}^{n} |\beta_j|$ represents the penalty term. The penalty term reduces coefficients β_j and simplifies the linear regression model. Thus, the LASSO performs variable selection and model shrinkage based on the magnitude of β_js.

3.4. Feature engineering

The original features may be dependent on each other which may reduce the performance of a predictor. To tackle this problem, feature engineering is used to create a new set of engineered features from the original ones. The feature engineering methods can either decrease or increase the dimensionality of feature vectors [34]. In the following, we provide a brief review of most commonly used feature engineering methods in credit scoring.

3.4.1. Principal Component Analysis

Principal Component Analysis (PCA) aims to transform/compress the data from a higher dimensional space $\mathbb{R}^n$ to a lower dimensional space $\mathbb{R}^k$ [35], where $k \ll n$. Note that relevant feature information (variance) is not lost during this transformation. The reduced feature space consists of *principal components* which are orthogonal to each other. Each principal component represents the direction of maximum variance. The principal components are computed via *eigenvalue* decomposition of the *covariance matrix* of features. The resulting *eigenvectors* from the decomposition are used as principal components. The following steps are used when constructing principal components:

1. Standardize the n dimensional dataset. Each feature $x_{i,j}$ of the feature vector $\mathbf{x}_i$ is standardized according to
$$x_{i,j} \leftarrow \frac{x_{i,j} - \mu_j}{\sigma_j}, \tag{7}$$
where μ_j and σ_j are the mean and standard deviation of jth feature.
2. Construct the *covariance matrix* $\boldsymbol{\Sigma}$. The covariance matrix is computed as
$$\boldsymbol{\Sigma} = \frac{1}{m-1}\sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^T, \tag{8}$$
where $\mathbf{x}_i$ is assumed a column vector.
3. Apply eigenvalue decomposition on the covariance matrix to compute eigenvalues (λ_js) and eigenvectors ($\mathbf{e}_j$s).
4. Select k eigenvectors that correspond to the k largest eigenvalues, where k is the dimensionality of the new feature subspace. Generally, the *variance explained ratio* is used to select the number of principal components (eigenvectors). For a set of all sorted eigenvalues
$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n, \tag{9}$$
the variance explained ratio is defined as follows
$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{n} \lambda_j}. \tag{10}$$
The chosen k eigenvectors will have a high total explained variance.
5. Finally, use k eigenvectors and project each feature vector $\mathbf{x}$ to k-subspace, i.e.,
$$\hat{\mathbf{x}} = (\mathbf{x} - \boldsymbol{\mu})^T[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k], \tag{11}$$
where $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_n]^T$ is the mean vector and $\hat{\mathbf{x}} \in \mathbb{R}^k$ is the projection of $\mathbf{x}$ onto k-subspace.

3.4.2. Autoencoders

Fig. 3 shows an autoencoder which is a type of neural network with three layers, namely; input, hidden and output layers. For a typical autoencoder, the input variables are the same as the target variables. The autoencoder learns representations of its inputs at hidden layer and tries to reconstruct its inputs from the learned representations by optimizing a cost function on a training dataset, i.e.

$$E = \frac{1}{2}\sum_{i=1}^{m} \|a_2(a_1(\boldsymbol{\alpha}^{(1)}\mathbf{x}_i)\boldsymbol{\alpha}^{(2)}) - \mathbf{x}_i\|_2^2, \tag{12}$$

where $\boldsymbol{\alpha}^{(1)}$, $\boldsymbol{\alpha}^{(2)}$ are weights and a_1, a_2 are activation functions between input and hidden layer, hidden layer and output layer, respectively. Once $\boldsymbol{\alpha}^{(1)}$ and $\boldsymbol{\alpha}^{(2)}$ are learned, a feature vector $\mathbf{x}$ is mapped to $\hat{\mathbf{x}}$ as follows $\hat{\mathbf{x}} = \sigma_1(\boldsymbol{\alpha}^{(1)}\mathbf{x})$ which is further used as engineered feature vector. One can choose different number of hidden layers for the autoencoder to learn complex representations in the data. Furthermore, the autoencoder can both decrease or increase the dimensionality of the projected vectors $\hat{\mathbf{x}}$ with respect to the dimensionality of the input feature vector $\mathbf{x}$.

3.4.3. Linear discriminant analysis

Linear discriminant analysis (LDA) was first introduced by Fisher [36]. The following outlines the steps taken when using LDA for feature selection [37].

1. Calculate n-dimensional mean vectors $\boldsymbol{\mu}^{(+1)}$ and $\boldsymbol{\mu}^{(-1)}$ for "good" (or positive (+1)) and "bad" (or negative (−1))

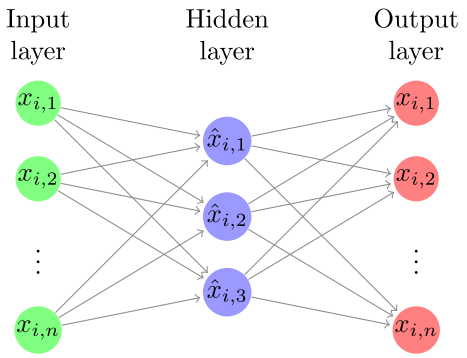

Fig. 3. An example of an autoencoder with one hidden layer. The green circles represent inputs, the blue circles represent engineered features, the red circles are reconstructed inputs.

classes, respectively:

$$\boldsymbol{\mu}^{(+1)} = \frac{1}{m^{(+1)}} \sum_{k=1}^{m^{(+1)}} \mathbf{x}_k^{(+1)}, \boldsymbol{\mu}^{(-1)} = \frac{1}{m^{(-1)}} \sum_{k=1}^{m^{(-1)}} \mathbf{x}_k^{(-1)}. \tag{13}$$

2. Construct the within-class scatter matrix $\mathbf{S}_w$ and the between-class scatter matrix $\mathbf{S}_b$:

$$\mathbf{S}_w = \sum_{\forall c \in \{+1,-1\}} \frac{m^{(c)}}{m^{(+1)} + m^{(-1)}} \sum_{k=1}^{m^{(c)}} \left(\mathbf{x}_k^{(c)} - \boldsymbol{\mu}^{(c)}\right) \left(\mathbf{x}_k^{(c)} - \boldsymbol{\mu}^{(c)}\right)^T \tag{14}$$

$$\mathbf{S}_b = \sum_{\forall c \in \{+1,-1\}} \frac{m^{(c)}}{m^{(+1)} + m^{(-1)}} \left(\boldsymbol{\mu}^{(c)} - \boldsymbol{\mu}\right) \left(\boldsymbol{\mu}^{(c)} - \boldsymbol{\mu}\right)^T \tag{15}$$

where $\boldsymbol{\mu} = \frac{m^{(+1)}}{m^{(+1)}+m^{(-1)}} \boldsymbol{\mu}^{(+1)} + \frac{m^{(-1)}}{m^{(+1)}+m^{(-1)}} \boldsymbol{\mu}^{(-1)}$ is the overall mean of the dataset.

3. Apply eigenvalue decomposition on the matrix $\mathbf{inv}(\mathbf{S}_w)\mathbf{S}_b$ to compute eigenvalues (λ_js) and eigenvectors ($\mathbf{e}_j$s), where $\mathbf{inv}(\cdot)$ is matrix inverse.
4. Similar to PCA, select k eigenvectors that correspond to the k largest eigenvalues.
5. Finally, use k eigenvectors and project each feature vector $\mathbf{x}$ to k-subspace, i.e.,

$$\hat{\mathbf{x}} = \mathbf{x}^T[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k], \tag{16}$$

where $\hat{\mathbf{x}} \in \mathbb{R}^k$ is the projection of $\mathbf{x}$ onto k-subspace.

4. Supervised learning

Credit scoring is a *supervised learning* problem, specifically it is a binary classification problem, where the aim is to classify good borrowers and bad borrowers. In this section we first define supervised learning problem and then present most commonly used statistical and machine learning methods in credit scoring.

4.1. Supervised learning problem

In the context of credit scoring, supervised learning problem can be defined as searching for a model $\phi : \mathbb{R}^n \mapsto \{+1, -1\}$ which maps a feature vector $\mathbf{x} \in \mathbb{R}^n$ to a predicted class label $\hat{y} \in \{+1, -1\}$, i.e

$$\phi_{\boldsymbol{\alpha}} : \mathbf{x} \rightarrow \hat{y}, \tag{17}$$

$\boldsymbol{\alpha}$ is the set of the model parameters. For the ease of reading $\phi_{\boldsymbol{\alpha}}$ and ϕ are used interchangeably.

In order to learn the model parameters, information from the previously processed applicants are collected to form a labeled dataset

$$\mathcal{D} = \{(\mathbf{x}_k, y_k)\}_{k=1}^m,$$

where $\mathbf{x}_k \in \mathbb{R}^n$ denotes kth applicant's feature vector and $y_k \in \{+1, -1\}$ is the corresponding label from the set of classes (good borrower $= +1$) and (bad borrower $= -1$). The dataset $\mathcal{D}$ is split into a training set $\mathcal{D}_{train} \subset \mathcal{D}$ and a test dataset $\mathcal{D}_{test} \subset \mathcal{D}$ where $\mathcal{D}_{train} \cap \mathcal{D}_{test} = \emptyset$ and $\mathcal{D}_{train} \cup \mathcal{D}_{test} = \mathcal{D}$.

Once the training and testing datasets are formed, the parameters $\boldsymbol{\alpha}$ of the model ϕ are learned on the training dataset $\mathcal{D}_{train}$ by minimizing a cost function C

$$C(\mathcal{D}_{train}, \phi) = \sum_{\forall \mathbf{x}_k \in \mathcal{D}_{train}} d(y_k, \phi(\mathbf{x}_k)), \tag{18}$$

$d(y_k, \phi(\mathbf{x}_k))$ is a distance measure, such as the Mean Squared Error (MSE) or Cross-Entropy, between y_k and its prediction $\hat{y}_k = \phi(\mathbf{x}_k)$ from the model. The optimal parameters $\boldsymbol{\alpha}^*$ of the model ϕ are obtained according to

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \; C(\mathcal{D}_{train}, \phi_{\boldsymbol{\alpha}}). \tag{19}$$

The performance of the learned model $\phi_{\boldsymbol{\alpha}^*}$ is assessed on the test dataset $\mathcal{D}_{test}$.

4.2. Statistical learning models

This section discusses popular statistical learning techniques in credit scoring.

4.2.1. Linear discriminant analysis

The *Linear Discriminant Analysis (LDA)* develops a linear combination of independent features which yield the largest mean difference between classes. Under the assumption of equal class covariances [38], we have the following solution in Eq. (20) for a binary classification problem, such as credit scoring. For a given feature vector $\mathbf{x}$ we decide for class $\hat{y} = +1$ if the expression on the right hand side is greater than the expression on the left hand side, otherwise we decide for class $\hat{y} = -1$

$$(\boldsymbol{\mu}^{(+)} - \boldsymbol{\mu}^{(-1)})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \gtrless \log(P(y = -1)) - \log(P(y = +1)), \tag{20}$$

where $P(\cdot)$ denotes probability.

4.2.2. Logistic regression

The Logistic Regression (LR) is the most commonly used statistical model in credit scoring due to interpretability of its decisions. The LR model is defined as

$$P(y = +1|\mathbf{x}) = \frac{1}{1 + \exp(\alpha_0 + \boldsymbol{\alpha}^T \mathbf{x})} \tag{21}$$

and

$$P(y = -1|\mathbf{x}) = 1 - P(y = +1|\mathbf{x}) = \frac{\exp(\alpha_0 + \boldsymbol{\alpha}^T \mathbf{x})}{1 + \exp(\alpha_0 + \boldsymbol{\alpha}^T \mathbf{x})}, \tag{22}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the feature vector, $P(y = +1|\mathbf{x})$ is the probability of classifying $\mathbf{x}$ as a good borrower, $P(y = -1|\mathbf{x})$ is the probability of classifying $\mathbf{x}$ as a bad borrower and $\{\alpha_0, \boldsymbol{\alpha}\}$ are the model parameters estimated by using, e.g., maximum likelihood estimation on the training dataset [4]. Once the model parameters are estimated, the decision on an input feature vector $\mathbf{x}$ is made in favor of $\hat{y} = +1$ if

$$P(y = +1|\mathbf{x}) \geq P(y = -1|\mathbf{x}), \tag{23}$$

which is equivalent to the following decision rule

$$\hat{y} = \begin{cases} +1 & \text{for } 1 \geq \exp(\alpha_0 + \boldsymbol{\alpha}^T \mathbf{x}); \\ -1 & \text{otherwise.} \end{cases} \tag{24}$$

4.2.3. Naïve Bayes

The Naïve Bayes (NB) classifier is based on Bayes decision rule [39]. The adjective "naive" comes from the assumption that the features in a dataset are mutually independent. The Naïve Bayes classifier decides for class $y = +1$ over $y = -1$ if

$$P(y = +1|\mathbf{x}) \geq P(y = -1|\mathbf{x}), \tag{25}$$

where

$$P(y = +1|\mathbf{x}) = \frac{p(\mathbf{x}|y = +1)P(y = +1)}{p(\mathbf{x})} \tag{26}$$

and

$$P(y = -1|\mathbf{x}) = \frac{p(\mathbf{x}|y = -1)P(y = -1)}{p(\mathbf{x})}. \tag{27}$$

The probability density $p(\mathbf{x})$ of observing $\mathbf{x}$ is defined according to

$$p(\mathbf{x}) = p(\mathbf{x}|y = +1)P(y = +1) + p(\mathbf{x}|y = -1)P(y = -1) \tag{28}$$

and the conditional density functions $p(\mathbf{x}|y = -1)$ and $p(\mathbf{x}|y = +1)$ are modeled according to

$$p(\mathbf{x}|y = -1) = \frac{1}{(2\pi)^{\frac{m}{2}} |\boldsymbol{\Sigma}^{(-)}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}^{(-)})^T \mathbf{inv}\left(\boldsymbol{\Sigma}^{(-)}\right)(\mathbf{x} - \boldsymbol{\mu}^{(-)})\} \tag{29}$$

and

$$p(\mathbf{x}|y = +1) = \frac{1}{(2\pi)^{\frac{m}{2}} |\boldsymbol{\Sigma}^{(+)}|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}^{(+)})^T \mathbf{inv}\left(\boldsymbol{\Sigma}^{(+)}\right)(\mathbf{x} - \boldsymbol{\mu}^{(+)})\}, \tag{30}$$

where $\boldsymbol{\Sigma}^{(-)}$ and $\boldsymbol{\Sigma}^{(+)}$ are covariance matrixes computed on negative and positive instances from the training dataset. For the Naïve Bayes classifier, $\boldsymbol{\Sigma}^{(-)}$ and $\boldsymbol{\Sigma}^{(+)}$ are diagonal matrices.

Note that $p(\mathbf{x}|y = -1)$ and $p(\mathbf{x}|y = +1)$ are uni-modal multivariate Gaussian distribution density functions which may not adequately model multi-modal data. In this case, one can employ multi-variate *Gaussian Mixture Model* [40] and Expectation Maximization (EM) [41] to learn the model parameters on the training dataset.

4.3. Machine learning models

This section discusses popular machine learning techniques in credit scoring.

4.3.1. k-Nearest neighbor

A k-Nearest Neighbor (k-NN) [42] assigns to an input feature vector $\mathbf{x}$ the class of the majority of its k nearest neighbors in the training dataset. The nearest neighbors are determined by calculating the *Euclidean distance* or *Mahalanobis distance* between the input feature vector $\mathbf{x}$ and the training dataset $\{\mathbf{x}_k\}_{k=1}^m$. Thus the class for the new data point is

$$y = \underset{y \in \{+1,-1\}}{majority}\left[\underset{k}{argmin} \left\|\{\mathbf{x}_k\}_{k=1}^m - \mathbf{x}\right\|\right]. \tag{31}$$

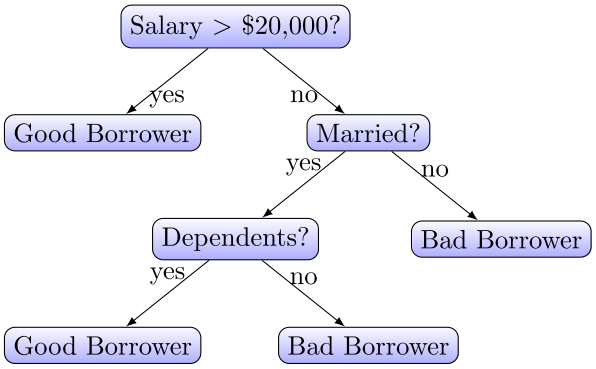

Fig. 4. An example of a decision tree in credit scoring.

4.3.2. Decision trees

As shown in Fig. 4 a Decision Tree (DT) [39] asks a series of questions in order to arrive to an answer (which is a class label). The root of the Decision Tree is called the *root node* and it is the most discriminating feature. The *leaf nodes* denote the classes.

4.3.3. Support vector machines

A Support Vector Machine (SVM) [43] uses an idea of a hyperplane (which is a decision boundary) that separates classes in a high dimensional feature space. The linear SVM focuses on maximizing the margin $\|\boldsymbol{\alpha}\| = \sqrt{\sum_{i=1}^m \alpha_i^2}$ between the negative and positive hyperplanes. The correct class is assigned by using the following equation:

$$y = \begin{cases} +1, \ if \ b + \boldsymbol{\alpha}^T \mathbf{x} \geq +1 \\ -1, \ if \ b + \boldsymbol{\alpha}^T \mathbf{x} \leq -1. \end{cases} \tag{32}$$

where b is the bias. For non-linear cases, a kernel trick [44] is used to project features into a high dimensional space.

4.3.4. Artificial neural networks

An Artificial Neural Network (ANN) [45] is a system which is motivated by a biological neural network system. An ANN emulates the way in which a biological neural network of the brain processes information by means of interconnected neurons [2,46,47]. Typically, a neural network consists of three layers, namely an input, hidden and output layers [48]. Essentially, training a neural network involves the process of finding optimal weights that map the input and output layers by means of *back-propagation*.

For a given input feature vector $\mathbf{x}$, a three-layer ANN computes the output $\hat{\mathbf{y}}$ according to

$$\hat{\mathbf{y}} = a_2(a_1(\boldsymbol{\alpha}^{(1)}\mathbf{x} + \alpha_0^{(1)})\boldsymbol{\alpha}^{(2)} + \alpha_0^{(2)}), \tag{33}$$

where $(\alpha_0^{(1)}, \boldsymbol{\alpha}^{(1)})$, $(\alpha_0^{(2)}, \boldsymbol{\alpha}^{(2)})$ are weights and a_1, a_2 are activation functions between input and hidden layer, hidden layer and output layer, respectively. The parameters are learned on a training set. The ANN performs final decision by applying a decision function, such as soft-max, on $\hat{\mathbf{y}}$. The ANN was first applied in credit scoring by Odom and Sharda [49].

4.3.5. Random forests

A Random Forest (RF) is an ensemble of decision trees [50], i.e. K decision trees are built on bootstrapped samples with m observations. Each decision tree is developed using a subset of randomly chosen k features. Each decision tree will give a class of a new feature vector. Thereafter, for overall classification, a RF assigns the class of the new feature vector by using majority vote based on the outputs from the decision trees.

4.3.6. Boosting

The Boosting works by estimating multiple models iteratively and assigning weights to data instances [51]. Boosting starts by developing a weak model (i.e. shallow decision tree). Thereafter a better model that will address errors of the previous model is developed. The instances which were incorrectly classified by the previous model will be assigned higher weights. The popular boosting technique is *Adaptive Boosting* (Ada-boost). The Ada-boost assigns a class to an input feature vector $\mathbf{x}$ in the following way

$$\hat{y} = sign\left(\sum_{t=1}^{T} \alpha_t \phi_t(\mathbf{x})\right), \tag{34}$$

where α_t is the weight of classifier $\phi_t(\mathbf{x})$ and T is the total number of classifiers. The parameters α_ts are learned on the training dataset.

4.3.7. Extreme gradient boost

The XGBoost [52] is short for extreme gradient boosting. The XGBoost is famously known for its processing speed and performance. It works similar to gradient boosting but builds the decision trees in parallel instead of building the decision trees in a series [53]. The optimization function to minimize is

$$\mathcal{L}^{(t)} = \sum_{k=1}^{n} l(y_k, \hat{y}_k^{(t-1)} + \phi_t(\mathbf{x}_k)) + \Omega(\phi_t) \tag{35}$$

where $l(\cdot)$ is a loss function and $\Omega(\phi_t)$ is a regularization term that penalizes complexity of the model. The goal of XGBoost is to find the ϕ_t which minimizes the objective function $\mathcal{L}^{(t)}$ [54].

4.3.8. Bagging

The Bagging classifier is also known as the *bootstrap aggregation* [55]. The Bagging technique takes K bootstraps from the underlying datasets and builds a classifier for every bootstrap. Then a class label is assigned by using a majority vote from votes of the K classifiers

$$y = \underset{y\in\{+1,-1\}}{argmax} \sum_{i=1}^{K} 1\left(y = \phi_i(\mathbf{x})\right), \tag{36}$$

where

$$1\left(y = \phi_i(\mathbf{x})\right) = \begin{cases} 1, & if\ y = \phi_i(\mathbf{x}); \\ 0, & if\ y \neq \phi_i(\mathbf{x}). \end{cases} \tag{37}$$

4.4. Deep learning models

Deep learning algorithms have been successfully applied in literature since the 1980s in an attempt to improve classification accuracy. Moreover, deep learning models with optimal hidden layers have been developed to reveal information not easily detectable with traditional statistical and machine learning models. The following subsections highlight the deep learning models that are used in credit risk datasets.

4.4.1. Restricted Boltzmann machines

As shown in Fig. 5, a Restricted Boltzmann Machine (RBM) can be perceived as an undirected neural network with two layers. The two layers can be called the hidden and the visible layers. The hidden layer is used as a feature detector while the visible layer is used to train the input data [56]. Given n visible layers $\mathbf{v}$, and m hidden layers $\mathbf{h}$, the energy function is

$$E(v, h) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_{ij} v_i h_j, \tag{38}$$

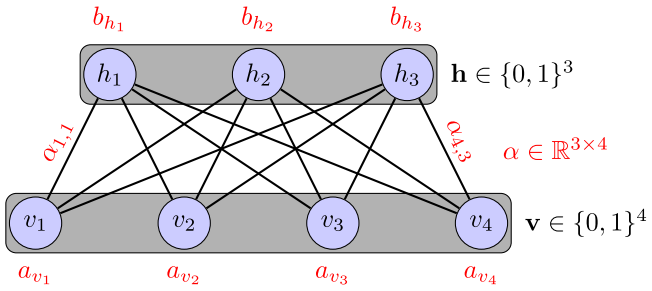

Fig. 5. A restricted Boltzmann machine architecture with a visible layer $\mathbf{v}$ and a hidden layer $\mathbf{h}$.

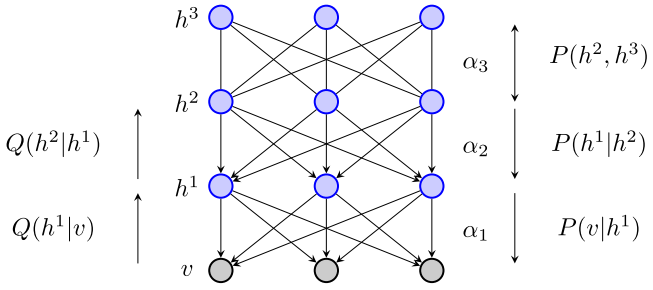

Fig. 6. A deep belief neural network architecture which is composed of several hidden layers of restricted Boltzmann machines.

where a_i and b_j are biases for binary variables v_i and h_j respectively, while α_{ij} are the weights between unit i and j.

4.4.2. Deep belief neural networks

Deep Belief Networks (DBNs), introduced by Hinton et al. [57], is a class of deep neural networks. As shown in Fig. 6 a typical DBN is composed of several hidden layers of RBM. Essentially, an output of a lower level RBM can be perceived as input of the higher level RBM. Fig. 6 shows a graphical view of the DBN layers.

4.4.3. Convolutional neural networks

Convolutional Neural Networks (CNNs) were first introduced by LeCun et al. [58] and have been mainly used in image processing [59–61], and on time series data [60,62,63]. The convolutional neural network consists of an *input layer*, *convolutional layers*, *pooling layers* and *fully connected layers* (see Fig. 7). The convolutional and pooling layers are responsible for extracting data representations [64].

Input. A convolutional neural network takes inputs as *tensors* of shape (*height*, *width*, *channel*). In image classification, the height and width values represent the height and width of an image. The channel represents the depth/color of an image (e.g. 1 represents a gray-scale image and 3 represents an image with color). A tensor is a multidimensional array that contains numerical values or in some instances non-numeric values. The input shape is normally changed into a shape that the convolutional neural network anticipates and the input is scaled so that all values are in the [0, 1] interval [64]. Since credit scoring data is not an image data, a 1*D* convolutional neural network is normally used for non-image data with the exception of speech/voice data. Hence, for credit scoring, the CNN architecture consists of an input layer which is a tensor of shape $(m, 1, n)$, where m is the number of instances and n is the number of features. Each instance is a feature vector with three channels that has a shape $(1, 1, n)$. All inputs are scaled into [0, 1] interval.

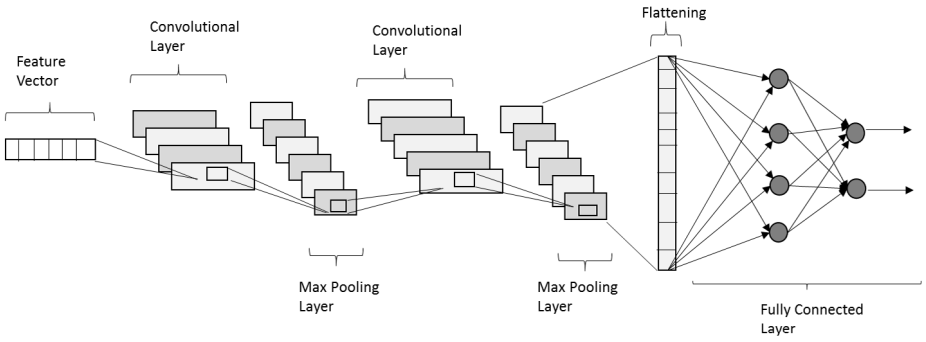

Fig. 7. A 1*D* convolutional neural network architecture which consists of two convolutional layers, two pooling layers and a fully connected layer.

Convolutional layer. The word "convolution" refers to a mathematical operation which is a specialized kind of linear operation [65]. Below is a convolution function

$$(f * g)(t) = \int_{-\infty}^{\infty} f(x)g(t-x)dx. \tag{39}$$

A convolutional layer learns local patterns [64], for example, in image classification, a convolutional layer breaks down an image into edges and textures. These local patterns are *feature maps* and are also known as *activation maps*. A convolution operation is applied on the input tensor and the *feature detector*. The feature detector also known as the *filter* or *kernel* is a tensor of shape (*height*, *width*). The feature detector slides along different locations of an input tensor to form a feature map. The feature map is a reduced and transformed input tensor. A non-linear activation function such as *ReLU* (Rectified Linear Unit) or a *Sigmoid* function is applied on activation maps to introduce non-linearity.

Pooling layer. A pooling layer is responsible for *downsampling* feature maps. At pooling layer, either a *max pooling* or an *average pooling* is used. For max pooling, each local input location is transformed by taking the maximum value of each channel over the location, whereas for average pooling, an average value of each channel is used. A convolutional neural network has a property called *spatial invariance*, which occurs at a pooling layer. The spatial invariance assures that the network is not influenced by input distortions or variations. Thus, the pooling layer is capable of preserving important features.

Flattening. The pooled feature maps are then converted into a single vector by a process called *flattening*. This single vector becomes an input to a fully connected artificial neural network.

Fully connected layer. A Fully Connected Layer is a feed forward artificial neural network. It consists of an input layer, hidden layer/s and an output layer. At output layer, a *loss function* or *cost function* is defined as follows

$$C(y, \hat{y}) = \sum_{\forall i} d(y_i, \hat{y}_i), \tag{40}$$

where $y_i, \hat{y}_i \in \{+1, -1\}$ and $d(y_i, \hat{y}_i)$ is a measure between y_i and $\hat{y}^{(i)}$ such as the Mean Squared Error(MSE) or Cross-Entropy. The mean squared error is

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2, \tag{41}$$

and the cross-entropy is

$$H(y, \hat{y}) = -\sum_{i=0}^{1} y_i \log \hat{y}_i. \tag{42}$$

Table 2
A confusion matrix with correct predictions, True Positives (TP) and True Negatives (TN), and incorrect predictions, False Positives (FP) and False Negatives (FN).

		Predicted	
		Positives	Negatives
Actual	Positives	TP	FN
	Negatives	FP	TN

A neural network tries to minimize the cost function by learning and updating the connection weights using *back propagation* [66].

4.4.4. Deep Multi-Layer Perceptron

A Multi-Layer Perceptron with multiple hidden layers is called Deep Multi-Layer Perceptron. Deep Multi-Layer Perceptron is a directed neural network and training is performed by back-propagation. The cost or loss function for Deep Multi-Layer Perceptron uses *Softmax* and Cross-Entropy to update the weights. The Softmax function is

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{43}$$

and $f_j(z) \in [0, 1]$ and z_j is a probability output from the network.

5. Evaluation metrics

There are many evaluation metrics which are used in literature and the following metrics are the most popular metrics for assessing the performance of the models in credit scoring. The *Percentage Correctly Classified* (PCC), *Type I Error*, *Type II Error*, *Kolmogorov–Smirnov Statistic* (K-S), *Sensitivity/Recall*, *Specificity*, *Geometric-Mean*(G-mean), *F-measure* and *Area Under Receiver Operating Characteristics Curve* (AUC). Each of these metrics is shown in the sequel.

A *confusion matrix* (see Table 2) consists of *True Positives (TP)*, *True Negatives (TN)*, *False Positives (FP)* and *False Negatives (FN)* and is used for calculating the metrics which are discussed in this section. Based on credit scoring classification, TN is the number of borrowers who are correctly classified as non-defaults, FP is the number of non-defaulted borrowers who are incorrectly classified as defaults, FN is the number of defaulted borrowers who are incorrectly classified as non-defaults and TP is the number of borrowers who are correctly classified as defaults.

From the confusion matrix the performance metrics can be derived, such as

$$\text{PCC} = \frac{(TP + TN)}{(TP + FP + FN + TN)}, \tag{44}$$

$$\text{Type I} = \frac{(FP)}{(FP + TN)}, \tag{45}$$

$$\text{Type II} = \frac{(FN)}{(TP + FN)}, \tag{46}$$

$$\text{Sensitivity/Recall} = \frac{TP}{(TP + FN)}, \tag{47}$$

$$\text{Specificity} = \frac{TN}{(FP + TN)}, \tag{48}$$

$$\text{G–mean} = \sqrt{\frac{TN}{(FP + TN)} \times \frac{TP}{(TP + FN)}}, \tag{49}$$

$$\text{F–measure} = \frac{2 \times Recall \times Precision}{Precision + Recall}. \tag{50}$$

The Kolmogorov–Smirnov Statistic

$$\text{K–S} = max|P(s|G) - P(s|B)| \tag{51}$$

calculates the maximum distance between the cumulative score distribution of bads $P(s|B)$ and goods $P(s|G)$

$$P(s|B) = \sum_{t \leq s} p(t|B) \tag{52}$$

and

$$P(s|G) = \sum_{t \leq s} p(t|G) \tag{53}$$

respectively and where $s \in \mathbb{Z}^+$ denotes a score in $500 \leq s \leq 1000$.

The AUC [67] is

$$\text{AUC} = \frac{1}{2}\left(1 + \frac{TP}{TP + FN} - \frac{FP}{FP + TN}\right) \tag{54}$$

and measures the discriminative power of a model between classes.

6. Data imbalance

Imbalanced datasets occur as the number of observations in one class (referred to as a minority class) in a dataset is usually much lower than the number of observations in the other class (referred to as a majority class). There are studies that have dealt with imbalanced datasets. For example, Brown and Mues [15] showed that the random forest and gradient boosting classifiers perform very well in a credit scoring context and are able to cope comparatively well with pronounced class imbalances in the datasets. On the other hand, when faced with a large class imbalance, the C4.5 decision tree algorithm, quadratic discriminant analysis and k-nearest neighbors perform significantly worse than the best performing classifiers. Douzas and Bacao [68] tackle the problem of imbalanced datasets by using a novel oversampling method, Self-Organizing Map-based Oversampling (SOMO).

There are a number of over-sampling (applied on minority class) or under-sampling techniques (applied on majority class) that can be found in literature. For example, Chawla et al. [69] proposed Synthetic Minority Over-sampling Technique (SMOTE). The SMOTE over-samples the minority class by taking each minority class sample and creating synthetic examples (along the line segments joining any/all of the k minority class nearest neighbors). Thereafter, neighbors from the k nearest neighbors are randomly chosen, depending on the amount of over-sampling required. For instance, if the amount of over-sampling needed is 300%, only three neighbors are chosen and one sample is generated in the direction of each. Synthetic samples are generated by taking the difference between the feature vector (sample) under consideration and its nearest neighbor. Thereafter this difference is multiplied by a random number between 0 and 1, and is added to the feature vector under consideration to form a synthetic feature vector.

There are other techniques that neither do over-sampling nor under-sampling to deal with class imbalance, such as *wavelet data transformation* and *linear dependence approach*. Saia et al. [70] proposed a discrete wavelet transformation to deal with imbalanced data in credit scoring. Wavelets are small waves and wavelet transform captures both the time and frequency domains [71]. The discrete wavelet transform disintegrates a signal wave into a set of wavelets that is mutually orthogonal [71]. Saia et al. [70] approach outperformed the random forest model regardless of data distributions. Saia and Carta [72] proposed a linear dependence approach for imbalanced data. The idea was to exploit one class (the majority class) to overcome imbalanced class distribution issue. The linear dependence is determined by calculating the determinant of a square and non-square matrices. The determinant is a real number that is calculated from a matrix. When vectors are dependent, their determinant is zero. Saia and Carta [72] used an average of sub-matrix determinants and reliability band of the majority class to classify new instances. The proposed approach performed very similarly to the random forest model.

7. Model transparency

This section discusses different model transparency techniques. The aim of model transparency is to make non-transparent models explainable. In a case of credit scoring, this will help in explaining why a borrower was not granted a loan. In the following, we present the most commonly used model transparency techniques in credit scoring.

7.1. NeuroRule

The NeuroRule was first introduced by Setiono and Liu [73]. The NeuroRule uses three-layer feed-forward neural network. The NeuroRule consists of six steps:

(Step-1) Build and train a neural network;
(Step-2) Prune the neural network to remove irrelevant connections;
(Step-3) Discretize the hidden unit activation values of the pruned neural network by clustering;
(Step-4) Extract rules that describe the network outputs in terms of the discretized hidden activation values;
(Step-5) Extract rules that describe the discretized hidden unit activation values in terms of the network inputs;
(Step-6) Combine the two sets of rules extracted in steps 4 and 5 to obtain a final set of rules that relate the inputs and outputs of the network.

7.2. Trepan

Trepan was first introduced by Craven and Shavlik [74]. Trepan is an hybrid intelligent system. It [Trepan] induces a tree to approximate any classifier's predictions. Hence, Trepan is not restricted to neural networks. Trepan learns queries as opposed to normal decision trees which learn from data. At each node, Trepan stores (i) a subset of training observations, (ii) a set of *query instances* and (iii) a set of constraints (the conditions that observations must satisfy in order to reach the node).

7.3. LIME

LIME stands for Local Interpretable Model-Agnostic Explanations. LIME is capable of explaining a prediction of any classifier

by learning an interpretable model (e.g. a Decision Tree or a linear model) around a prediction [75]. The logic behind explaining predictions is for human subjects to have trust in the predictions if actions need to be taken. For example, if a doctor depends on a model to predict presence/absence of any disease, he/she will need to trust the model predictions in order for him/her to prescribe medication for patients. The doctor has prior knowledge in the field of medicine, he/she will either accept or reject the explanation based on his/her expertise. Hence, human prior knowledge plays an important role for accepting or rejecting explanations for predictions. LIME is designed to provide *trust* to human subjects.

7.3.1. Interpretable data representation

Before acquiring explanations, a data set needs to be in a format that is understandable to humans. This is applicable, e.g., in image classification or text classification, where features in the input space need to be transformed to a vector of ones and zeros to indicate the presence or absence of feature components. An observation $\mathbf{x} \in \mathbb{R}^n$ is transformed into $\hat{\mathbf{x}} \in \{0,1\}^{n'}$ for interpretability.

7.3.2. Local fidelity

The behavior of a classifier in the vicinity of an individual prediction determines the local faithfulness of an explanation. Let an explanation be denoted by $e \in E$ where E is a space of interpretable models, e.g., decision trees or linear models. The complexity of an explanation is given by $\Omega(e)$, e.g., for decision trees the complexity is the depth of a tree. Let $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ be a classifier which needs to be explained and let $\pi_{\mathbf{x}}(\mathbf{y})$ be a proximity measure between observation $\mathbf{y}$ to observation $\mathbf{x}$ which defines the locality around $\mathbf{x}$. The measure of how unfaithful is e in explaining $\phi(\mathbf{x})$ in the locality given by $\pi_{\mathbf{x}}(\mathbf{y})$ is

$$\psi\{\phi, e, \pi_{\mathbf{x}}\}, \tag{55}$$

where

$$\pi_{\mathbf{x}}(\mathbf{y}) = exp\left(-D(\mathbf{x},\mathbf{y})^2/\sigma^2\right) \tag{56}$$

is an exponential kernel defined on some distance D (e.g. Mahalanobis distance).

To obtain interpretability and local fidelity, an optimization is performed to minimize equation Eq. (55) and $\Omega(e)$ is kept low. Hence, the prediction explanation for observation $\mathbf{x}$ is

$$\zeta(x) = \underset{e \in E}{\operatorname{argmin}} \psi\{\phi, e, \pi_{\mathbf{x}}\} + \Omega(e). \tag{57}$$

7.3.3. Model agnostic

For explanations to be model-agnostic, we do not need to make assumptions about ϕ. This will ensure that any prediction of *any* black-box model can be explained.

7.3.4. Sampling

The Eq. (55) is approximated by drawing samples which are weighted by $\pi_{\mathbf{x}}$ around an observation of interest $\mathbf{x}$. The sampling is done by drawing non-zero elements of $\mathbf{x}$ uniformly at random [75]. Then $\hat{\mathbf{x}} \in \{0,1\}^{n'}$ is a perturbed sample which contains non-zero elements of $\mathbf{x}$. The perturbed data set is denoted by $\Re$. The outcome/label prediction for $\hat{\mathbf{x}}$ is given by $\phi(\mathbf{x})$.

7.3.5. Sparse linear explanations

This is a part where predictions are explained. In this stage an *explainer* is a linear model. Hence, $e(\hat{\mathbf{x}})$ is a linear model

$$e(\hat{\mathbf{x}}) = \boldsymbol{\alpha}^T \hat{\mathbf{x}}. \tag{58}$$

Let $\mathbf{x}$ be the original representation of $\hat{\mathbf{x}}$. Then the Eq. (55) becomes a locally weighted square loss

$$\psi\{\phi, e, \pi_{\mathbf{x}}\} = \sum_{\mathbf{x}, \hat{\mathbf{x}} \in \Re} \pi_{\mathbf{x}}(\Phi(\mathbf{x}) - e(\hat{\mathbf{x}}))^2. \tag{59}$$

8. Model limitations and imposed assumptions

The Linear Discriminant Analysis (LDA) technique is one of the two popular statistical techniques applied in credit scoring, and is subject to a parametric assumption which is defined as the underlying multivariate normal distribution of the features. Eisenbeis [76] argues that categorical features violate this parametric assumption. The other model which uses two strong statistical assumptions (i.e. i. the assumption that the features are conditionally independent ii. the values of numeric features are normally distributed) is the Naïve Bayes Classifier. The normally distributed assumption in Naïve Bayes classifier does not always hold in other domains, hence a need to estimate other continuous distributions is required [77].

The most popular statistical technique in credit scoring is Logistic Regression (LR). The Logistic Regression assumes a linear relationship between the inputs and the log odds. However, the linearity assumption does not always hold, there are other cases where the relationship between the independent variables and the log odds is non-linear. In such cases kernel Support Vector Machines (SVM) serves as one of the non-linear classification techniques to be used. However, Yu et al. [78] argues that the performance of SVM model is sensitive not only to the algorithm for solving the Quadratic Programming problem but also to the parameters setting in learning (i.e. regularized parameter balancing the classification margin and tolerable misclassification errors, and the kernel parameter). Yu et al. [78] propose Least Squares SVM (LSSVM) to solve the quadratic programming problem and the design of the experiment for parameter selection in SVM modeling.

Henley and Hand [42] propose the use of a non-parametric technique, k-nearest neighbor method in credit scoring. However, the k-nearest neighbor method is costly because it requires more computations (i.e. it calculates a metric distance for each data record stored during classification). Jiang [79] proposes the use of a decision tree (C4.5) in conjunction with an approach called Simulating Annealing Algorithm (SAA) which performs global optimization. In this study, Jiang highlights the shortcoming of the Decision Tree approach which is the inability to perform global optimization because of its local search strategy, hence the need to use SAA. Albeit the ANNs better performance when compared to other techniques in terms of accuracy, it lacks interpretability [5]. Baesens et al. [5] propose Decompositional and Pedagogical approaches to extract rules (without compromising the accuracy) from the ANN for interpretability.

Deep Learning models have not been applied extensively in credit scoring. However, the problem with deep learning models is interpretability. Since banks are governed by regulators, banks are required to be transparent in their credit scoring process. A bank needs to tell a borrower why his/her loan application has been rejected.

9. Emerging trends

Most banks have developed an interest in applying machine learning models (including deep learning modes) in their credit scoring systems. However, the regulators still maintain that the models which are used for credit scoring should be transparent. Despite their non-transparency, machine learning models are gaining traction in credit scoring. One way of mitigating the non-transparency issue of machine learning models is to rationalize

Table 3
Datasets that are used in literature for credit scoring. Each dataset has a number of good borrowers and a number of bad borrowers.

Dataset	Sample size	#Goods	#Bads	# of features
European Credit Bureau	186,574	179,544	7,030	324
UK	30,000	28,800	1,200	14
Barbados	21,117	20,614	503	20
Indonesia	14,700	14,290	4,410	31
Benelux 2 (Belgium, Netherlands, and Luxembourg)	7,190	5,033	2,157	28
Brazilian	4,504	4,144	360	5
Benelux 1 (Belgium, Netherlands, and Luxembourg)	3,123	1,040	2,083	27
University of California, San Diego	2,435	1,836	599	38
China	1,057	552	505	10
German	1,000	700	300	20
Iranian	1,000	950	50	27
Australian	690	307	383	14
Japanese	653	296	357	15
Polish	240	128	112	30
Texas Banks	162	81	81	19

(i.e. give justification as to why a certain decision has been made) the predictions.

9.1. Transparency

Methods that involve *rule extractions* for opening-up non-transparent models have been suggested in the literature. Baesens et al. [5] evaluated and contrasted three neural network rule extraction techniques, namely, Neurorule, Trepan, and Nefclass for credit-risk evaluation. They concluded that neural network rule extraction is an effective and powerful management tool which allows the development of advanced and user-friendly decision-support systems for credit-risk evaluation. Setiono and Liu [80] proposed a NeuroLinear method for extracting oblique decision rules from neural networks. The experimental results showed that NeuroLinear is effective in extracting compact and comprehensible rules that have high predictive accuracy from neural networks.

The techniques highlighted above are based on rules, and recently techniques such as *SHAP values*, *Partial Dependence* and *Explainable Boosting Machines* that do not require rules have been applied in domains (other than credit scoring). In their empirical study [81] proposed intelligible model that can easily explain its predictions. The study applied generalized additive models with feature interactions (also referred to as explainable boosting machines) on real health care problems. Their results showed that the proposed generalized additive model can perform comparably with best performing machine learning models such as random forest and logistic regression. Lundberg and Lee [82] used SHAP (SHapley Additive exPlanations) as a unified framework to interpret predictions. The SHAP method assesses contribution of each feature towards a prediction. On the other hand, the partial dependence checks how the prediction changes when different feature values are used. The partial dependence plots show the marginal impact one or two features have on the prediction of a machine learning model [83].

9.2. Deep learning in credit scoring

This section covers deep learning techniques in credit scoring. There is an emerging trend to replace statistical and classical machine learning techniques with deep learning techniques in credit scoring. For instance, Luo et al. [84] used Corporate Default Swaps (CDS) data to compare performances of deep belief networks with well-known credit scoring models such as logistic regression, multi-layer perceptron and support vector machine. Deep belief networks showed better performance. In their study, Ramasamy and Rajaraman [85] compared meta-cognitive restricted Boltzmann machine with extreme learning machines, support vector machines and multi-layer perceptron using Australian, German and Kaggle credit datasets. The meta-cognitive restricted Boltzmann machine showed superior performance. Tomczak and Zieba [67] assessed and compared performances of classification restricted Boltzmann machine with several traditional statistical and machine learning models, such as logistic regression, decision trees, adaboost, random forest etc., using German, Australian, Kaggle and Short-Term Loans data. The classification restricted Boltzmann machine showed comparable results on German, Kaggle and Short-Term Loan datasets, and better results on Australian dataset. Tran et al. [86] proposed a hybrid genetic programming and stacked auto-encoder network model. The proposed hybrid model was compared to logistic regression, $k-$NN, support vector machines, artificial neural networks and decision trees using German and Australian credit datasets. The hybrid model showed a better accuracy rate. Yeh et al. [87] used daily stock returns to predict defaults and non-defaults using deep belief network and support vector machines. The proposed deep belief network outperformed support vector machines. In their empirical study, Neagoe et al. [88] compared deep convolutional neural networks with multi-layer perceptron using German and Australian datasets. The results showed a superior overall accuracy rate for deep convolutional neural networks. These studies have shown the superiority of deep learning models in credit scoring. However, Hamori et al. [89] argue that the performance of deep learning models is dependent on the choice of activation function, the number of hidden layers and the dropout rate. The results in [89] showed a better performance for ensemble methods, such as boosting and bagging, when compared with deep neural networks using Taiwan credit dataset. These studies highlight and reiterate the applicability of deep learning algorithms to credit scoring data.

9.2.1. Data augmentation

Deep learning models require more data for training to avoid overfitting [90]. Data augmentation is normally performed to increase the number of training data points. This is done by applying several distortions to the original training images, such as changing the brightness and rotation of images and the distortions should not change the spatial pattern of target classes [91]. In [91], several data augmentations such as *random shift*, *random zoom*, *random horizontal flip* and *random rotation* were applied on image data. The random shift randomly shifts the images by a factor, the random zoom randomly zooms the images by a certain range, the random rotation randomly rotates the images by a certain angle and the random horizontal flip randomly flips the images horizontally to produce additional images. Their results [91] showed that data augmentation did not significantly improve the accuracy (i.e. accuracy increased by 1%). Kvamme

et al. [92] argue that data augmentation has been mostly done in image processing, and does not necessarily generalize well to other data sources.

On the other hand, Krizhevsky et al. [93] achieved significant improvements on error rates when data augmentation was applied on *ImageNet* dataset. A huge deep neural network (60 million parameters and 650,000 neurons) was used for this task, and for this they needed to increase the dataset size. Perez and Wang [94] used data augmentation on *gold fish vs. dogs* and *cats vs. dogs* datasets. The results showed better improvements in terms of accuracy on both datasets. Salamon and Bello [95] combined deep learning neural network with data augmentation to classify *audio* data. They used four different data augmentations and the proposed combination significantly outperformed deep neural network without augmentation. Frid-Adar et al. [96] proposed a combination of classic data augmentation with *Generative Adversarial Network (GAN)* data augmentation. The GAN was used to synthesize new *images of liver lesion*. This combination resulted in significant accuracy improvement. All of these studies used image datasets and credit scoring data is not in the form of images. However, the credit scoring data can be converted into images [97].

The above authors have contrasting views on the effectiveness of data augmentation on model accuracy. The conclusion from Krizhevsky et al. [93] is that when data augmentation is used in conjunction with large deep neural networks, the accuracy increases significantly compared to using data augmentations with "shallow" neural networks. Also Salamon and Bello [95] found that class accuracy is influenced differently by each augmentation.

10. Limitations in credit scoring literature

Below is the list of limitations which have been partially or completely ignored in credit scoring literature:

- ✓ No inclusion of macro-economic variables;
- ✓ The time it takes for borrowers to default is not determined;
- ✓ Exploratory Data Analysis: detection of outliers and distribution of variables (checking zero-variance for variables) is not performed;
- ✓ Time/Model Complexity is not factored;
- ✓ Creation of new features instead of performing feature space reduction techniques/feature selection is not taken into account;
- ✓ Correlation between dependent variable (or target variable) and independent variables is not assessed.
- ✓ Most studies in literature focus on homogeneous base classifiers and ignore heterogeneous base classifiers for ensemble methods;
- ✓ Using different cut-offs (instead of 0.5) to classify borrowers as either non-defaults or defaults is not covered in literature;
- ✓ Few studies incorporate Type II error. The type II error is more costly in credit scoring, since Type II (False Negative ratio) predicts a borrower as a good borrower but in actual fact he or she is a bad borrower.

The increases in macro-economic variables such as interest rate, inflation rate and unemployment rate may increase the risk of a borrower defaulting. Hence, it is key to incorporate macro-economic variables in credit scoring. Since macro-economic variables are time-varying, it is key to develop forward-looking credit scoring techniques. The survival analysis can cater for forward-looking credit scoring techniques. This can help not only to classify a borrower as default but to know when will a borrower default. The univariate analysis can help in identifying non-predictive variables and also detect outliers. Model/time complexity stems from models with too many parameters. Hence, it is key to keep a few parameters during model development. Although feature reduction/selection improves model accuracy, the literature needs to look at other feature engineering techniques such as taking the sum/product of two features to create a new feature. In cases where there are few independent features, a correlation between the target variable and the independent features can be performed to determine the high predictive features. The literature needs to focus on other ensemble techniques where base classifiers/learners are heterogeneous (i.e. models coming from different model classes). The default cut-off for classifying borrowers is 0.5, e.g. in logistic regression, the literature needs to assess the impact of using different cut-offs. The error which is of interest in credit scoring community is Type II error and the aim is to minimize this error. The literature needs to report more on this error when developing credit scoring models.

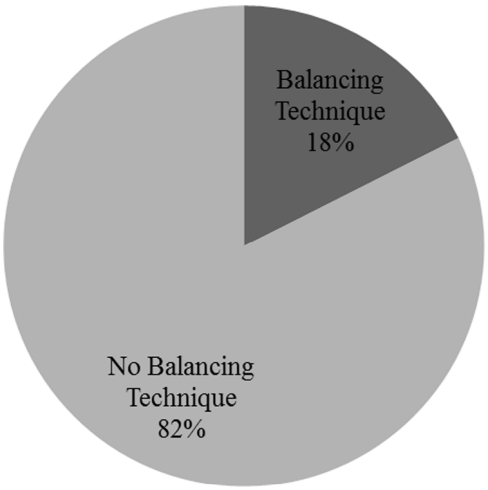

Fig. 8. A pie-chart that shows a proportion of literature papers that used data balancing techniques and a proportion of papers that did not balance their datasets.

11. Results

This section focuses on the meta-analysis of the results from primary studies. The frequency distributions of feature extraction techniques, the models and evaluation metrics from all primary studies are analyzed. The pie-charts for distribution of data balancing techniques and transparency techniques are also shown. The German and Australian credit datasets are selected for model performance comparison, as they are the most frequently used datasets in credit scoring.

11.1. Data balancing

Credit risk data is generally imbalanced (see Table 3). The non-default borrowers are usually more than the defaulted borrowers. In cases where there is a high imbalance in data, the models turn to be biased towards the majority class. To mitigate this bias, techniques such as over-sampling or under-sampling are usually performed. However, this study shows that only 18% of the primary studies in this literature survey have balanced their datasets (see Fig. 8). The most used technique is under-sampling the majority class (see Fig. 9).

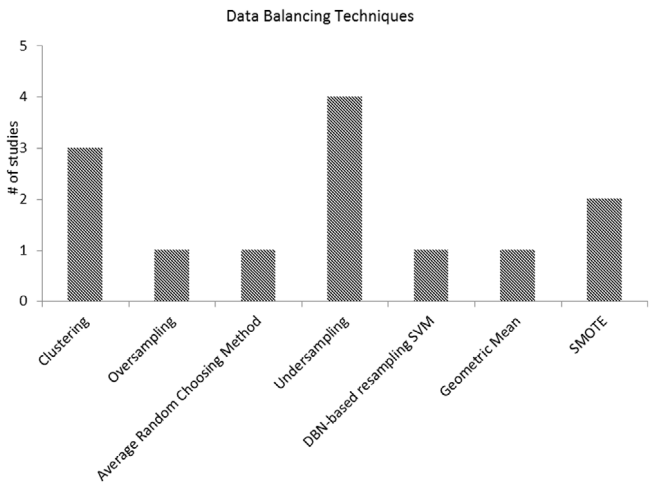

Fig. 9. A frequency distribution of data balancing techniques that are used in primary studies of this literature survey.

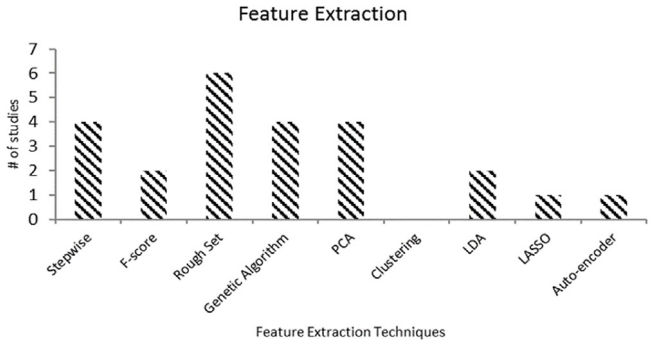

Fig. 10. A frequency distribution of feature extraction and engineering techniques that are used in primary studies of this literature survey.

11.2. Feature extraction

The distribution of feature extraction techniques, such as feature selection and feature engineering for this literature survey is shown in Fig. 10. This shows that out of 17 studies (see Table 4 for details) which used feature extraction, Rough Set technique was the most frequently used feature extraction technique, followed by Stepwise, Genetic Algorithm and PCA.

11.3. Evaluation metrics

Based on this current literature survey, Fig. 11 shows commonly used evaluation metrics in credit scoring. The most frequently used evaluation metrics are PCC and AUC (see Table 5 for details).

11.4. Models

This literature survey focuses on statistical, traditional and state-of-the-art machine learning models. Based on the results of this literature survey, Fig. 12 shows that LR, SVM and ANN were the most frequently used single classifiers (see Table 6 for details). For ensemble of classifiers, Boosting was the most frequently used ensemble classifier. The deep learning classifiers are among the least frequently used classifiers and this is attributable to the fact that deep learning classifiers are not applied extensively in credit scoring.

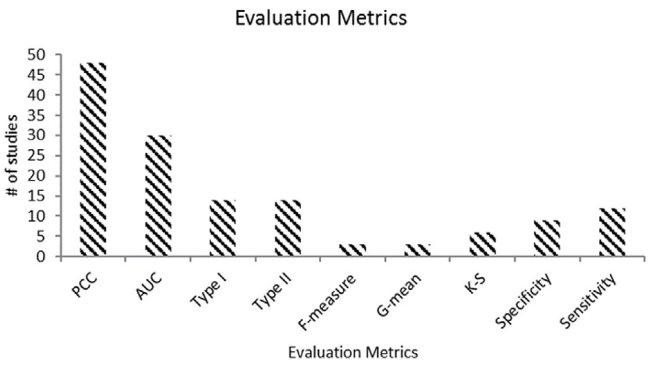

Fig. 11. A frequency distribution of evaluation metrics that are used in primary studies of this literature survey.

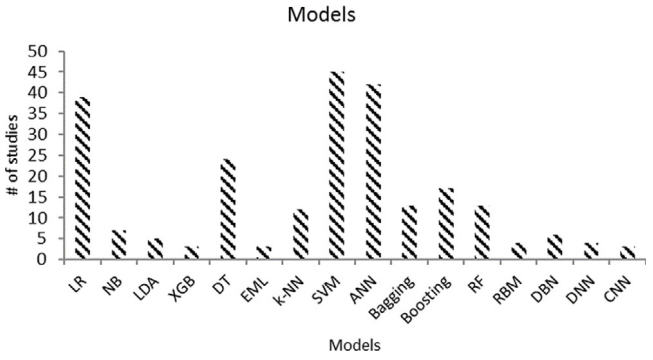

Fig. 12. A frequency distribution plot of most frequently used statistical and machine learning models in credit scoring.

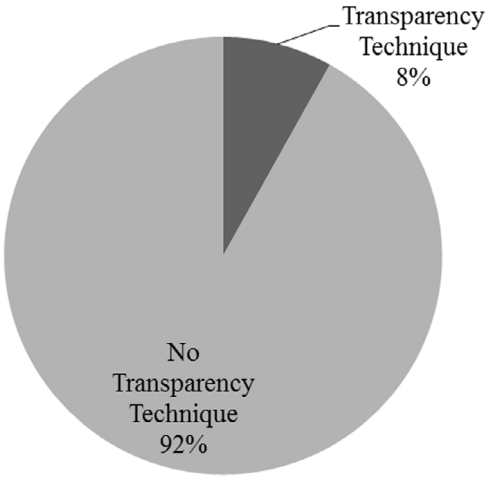

Fig. 13. A pie-chart that shows a proportion of studies that used transparency techniques and a proportion of studies that did not use transparency techniques.

11.5. Transparency

Under Basel II Accord [149], credit scoring models are required to be transparent (i.e. to explain their predictions). The majority of machine learning models are not transparent in nature. Hence, a need to make the models transparent in credit scoring is paramount. However, as can be seen in Fig. 13, this survey shows that only 8% of the primary studies have looked into techniques which make models transparent.

Table 4
Feature extraction methods used in primary studies. Legend: ST (Stepwise), FS (F-score), RS (Rough Set), GA (Genetic Algorithm), PCA (Principal Component Analysis), AE (AutoEncoder), LDA (Linear Discriminant Analysis), and LASSO (Least Absolute Shrinkage and Selection Operator).

Source	ST	FS	RS	GA	PCA	AE	LDA	LASSO
[17]		✓	✓					
[98]	✓						✓	
[25]			✓					
[99]			✓					
[100]					✓			
[101]	✓		✓					
[20]				✓				
[15]	✓							
[102]			✓					
[16]	✓							
[103]					✓			
[104]			✓					
[18]		✓						
[105]					✓			
[22]				✓				
[86]				✓				
[106]								✓
[107]								
[108]					✓	✓		
[109]							✓	
[110]				✓				

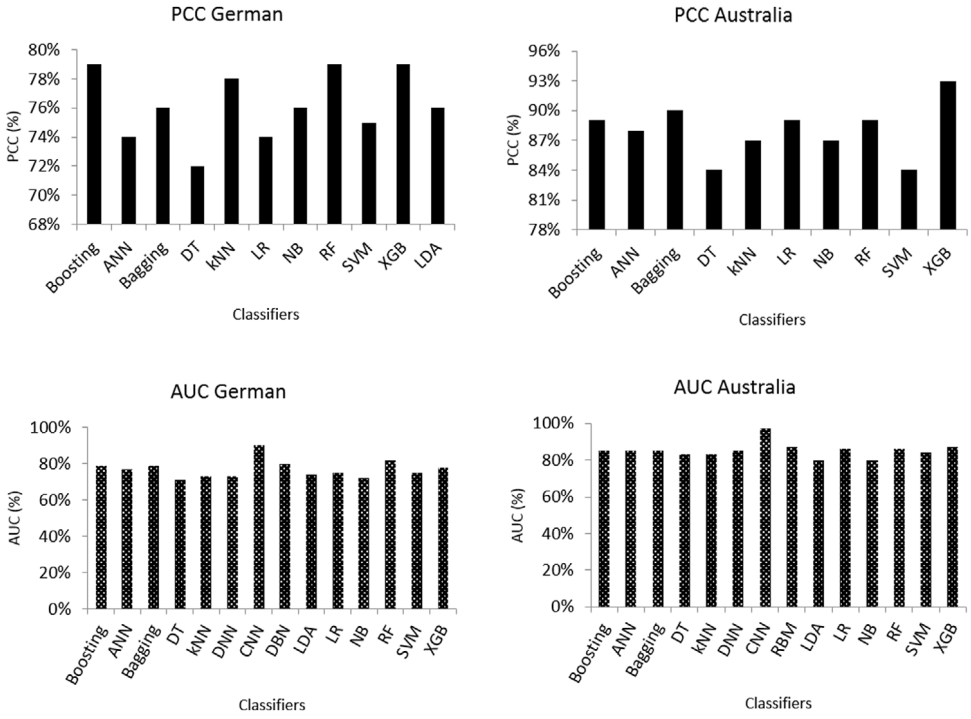

Fig. 14. Averages of pcc and auc that are calculated from the results reported on the papers considered for this literature survey that used German and Australian credit datasets.

11.6. Model performance on German and Australian datasets

Out of 74 primary studies, 39 primary studies either used German credit dataset, Australian credit dataset or both. The results for both PCC and AUC on both datasets were collected from the primary studies. The results (see Fig. 14) show that on average on both datasets, the Convolutional Neural Networks performed better in terms of PCC, followed by ensemble classifiers such as Random Forests, Bagging and Boosting. The high performance of CNN can be attributed to the fact that they can detect features that are more discriminative between borrowers. In terms of AUC, the ensemble of classifiers performed better than all single classifiers. We noted that different studies reported differently on PCC and AUC for each of the classifiers. This could be attributed to data pre-processing of each study.

12. Guiding machine learning framework for credit scoring

Based on the results of this literature survey, we propose a framework which is shown in Fig. 15, which serves as a guideline for credit scoring analysts. The nature of the data in credit scoring is either balanced or imbalanced. For both types of datasets, this framework suggests exploratory data analysis, data pre-processing, feature extraction techniques, the sampling methodology, the models to use as benchmarks, the best performing

Table 5
Evaluation metrics from primary studies. Legend: PCC (Percentage Correctly Classified), AUC (Area Under ROC Curve), T1 (Type I Error), T2 (Type II Error), F-M (F-measure), G-M (G-measure), K-S (Kolmogorov–Smirnov Statistic), SP (Specificity), SE (Sensitivity)

Source	PCC	AUC	T1	T2	F-M	G-M	K-S	SP	SE
[111]	✓								
[25]		✓			✓			✓	✓
[112]	✓								
[113]	✓								
[98]	✓								
[17]		✓							
[25]		✓			✓			✓	✓
[99]	✓								
[114]	✓		✓	✓					
[100]	✓								
[115]	✓		✓	✓					
[116]	✓							✓	✓
[117]	✓		✓	✓					
[118]			✓	✓				✓	✓
[114]		✓							✓
[101]	✓								
[119]		✓							
[20]		✓					✓		
[120]	✓		✓	✓					
[121]		✓	✓	✓					
[15]		✓							
[122]	✓								
[102]					✓			✓	✓
[16]		✓					✓		
[123]	✓							✓	✓
[104]	✓								
[18]	✓								
[103]	✓							✓	✓
[47]									
[124]		✓							
[22]	✓								
[105]	✓								
[125]		✓							
[126]	✓								
[127]		✓							
[67]		✓				✓		✓	✓
[128]	✓								
[87]	✓								
[129]									
[130]	✓	✓	✓	✓					
[131]		✓							
[132]	✓		✓	✓					
[133]		✓							
[86]	✓		✓	✓					
[134]	✓		✓	✓					
[135]	✓								
[46]	✓	✓	✓	✓					
[136]	✓	✓							
[137]	✓								
[107]		✓	✓	✓			✓		
[84]	✓	✓							
[138]	✓					✓			
[136]	✓		✓	✓					
[19]	✓								✓
[139]	✓							✓	✓
[85]						✓			✓
[140]	✓								
[141]	✓		✓	✓					
[106]		✓							
[142]	✓	✓							
[110]	✓	✓							
[143]							✓		
[92]	✓	✓						✓	✓
[108]		✓					✓		
[97]	✓	✓					✓		
[144]		✓							
[88]	✓								
[89]	✓	✓			✓				
[109]	✓								
[145]	✓	✓							
[56]	✓								
[146]	✓	✓							
[147]		✓							

Table 6
Models used in credit scoring. Legend: LR (Logistic Regression), NB (Naïve Bayes), LDA (Linear Discriminant Analysis), XGB (XGBoost), EML (Extreme Learning Machines), *k*-NN (*k*-Nearest Neighbor), SVM (Support Vector Machine), ANN (Artificial Neural Network), BA (Bagging), BO (Boosting), RF (Random Forest), RBM (Restricted Boltzmann Machine), DBN (Deep Belief Network), DMLP (Deep Multi-Layer Perceptron), and CNN (Convolutional Neural Network).

Source	LR	NB	LDA	XGB	DT	EML	*k*-NN	SVM	ANN	BA	BO	RF	RBM	DBN	DMLP	CNN
[111]	✓															
[98]	✓								✓							
[99]					✓					✓						
[112]	✓	✓			✓		✓		✓	✓	✓					
[113]								✓	✓	✓						
[25]	✓							✓	✓							
[17]								✓								
[114]								✓								
[100]								✓								
[115]	✓				✓			✓	✓	✓	✓					
[116]								✓								
[117]								✓					✓	✓		
[119]	✓				✓											
[114]	✓	✓	✓		✓		✓	✓								
[101]								✓								
[118]	✓				✓											
[20]	✓															
[120]	✓	✓			✓		✓	✓	✓	✓	✓					
[15]	✓		✓		✓		✓	✓	✓		✓	✓				
[121]	✓				✓		✓	✓	✓	✓	✓	✓				
[16]	✓				✓											
[123]	✓															
[122]								✓	✓							
[102]	✓				✓				✓							
[103]	✓							✓								
[18]								✓								
[104]								✓								
[18]								✓								
[47]					✓			✓	✓	✓	✓					
[124]					✓											
[105]								✓					✓			
[22]									✓							
[125]	✓							✓								
[126]								✓								
[127]	✓		✓					✓	✓		✓	✓				
[129]									✓							
[131]	✓		✓		✓		✓	✓	✓		✓	✓				
[128]								✓								
[87]	✓							✓						✓		
[67]	✓				✓			✓		✓	✓	✓	✓			
[132]	✓							✓	✓							
[130]	✓		✓		✓		✓	✓	✓		✓	✓				
[135]										✓						
[86]	✓				✓		✓	✓	✓		✓				✓	
[134]	✓				✓			✓	✓					✓		
[133]		✓			✓			✓	✓			✓				
[142]	✓				✓			✓	✓	✓	✓					
[141]	✓								✓							
[46]	✓							✓	✓	✓	✓	✓				
[136]	✓				✓	✓	✓	✓	✓	✓	✓					
[137]	✓										✓					
[140]		✓														
[106]								✓								
[107]	✓															
[84]	✓							✓	✓					✓		
[138]	✓							✓	✓						✓	
[136]	✓				✓	✓	✓	✓	✓							
[19]	✓															
[139]								✓						✓		
[85]						✓		✓	✓				✓			
[97]	✓											✓				✓
[144]	✓								✓		✓	✓			✓	
[88]									✓							✓
[109]		✓			✓		✓	✓	✓			✓				
[145]	✓			✓	✓			✓				✓				
[143]	✓															
[92]																✓
[108]																
[146]				✓												
[147]				✓												
[148]	✓		✓	✓	✓			✓	✓			✓				
[110]		✓					✓	✓								
[56]								✓						✓		
[89]									✓	✓	✓	✓			✓	

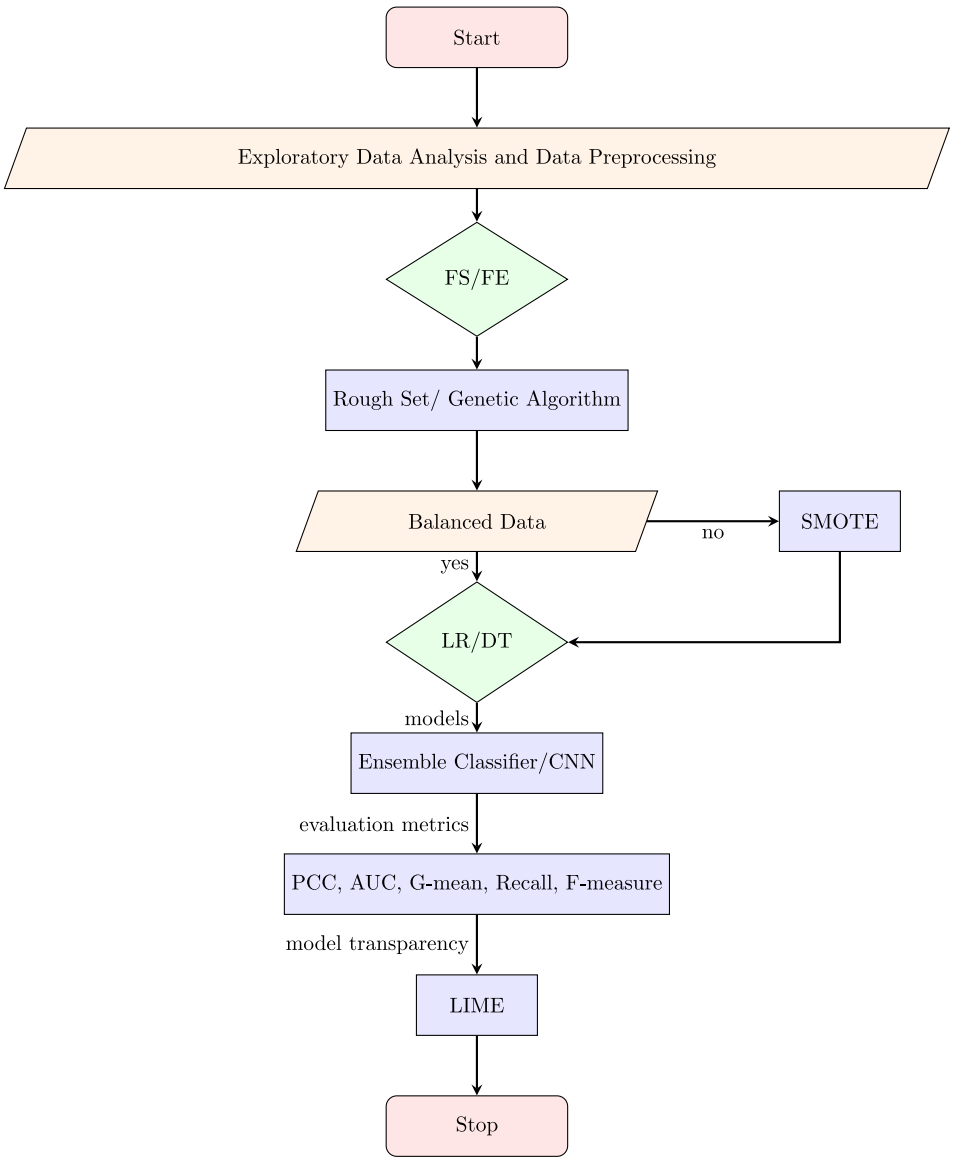

Fig. 15. The guiding machine learning credit scoring framework that is proposed in this literature survey.

models in credit scoring, the evaluation metrics and the technique which explains model predictions. This literature survey showed that the most frequently used techniques for feature extraction are Rough Set and Genetic Algorithm. The sampling methodology that we are suggesting for imbalanced datasets is SMOTE. The sampling is only applied on the training set. The models which can be used as benchmarks is Logistic Regression (LR) and Decision Trees (DT). The LR performs similarly to most traditional Machine learning models and the choice of selecting DT is its capability of explaining predictions. This literature survey shows that ensemble classifiers and CNN performed better when compared to other models, hence we suggest these two models for use in credit scoring. For imbalanced datasets, evaluation metrics which can be applied are G-mean, Recall and F-measure. Since predictions from ensemble classifiers and CNN cannot be explained, we suggest LIME for predictions' explanations.

13. Conclusion and future work

Many studies over the years have evaluated and contrasted the performances of different statistical and classical machine learning models in credit scoring. However, no consensus has been reached in identifying the best performing model. This literature survey systematically reviewed the most commonly used statistical, classical machine learning and deep learning models in credit scoring. The performances (on German and Australian credit datasets) of statistical, classical machine learning and deep learning models that were reported in literature were compared in this literature survey. The literature results showed that an ensemble of classifiers generally outperform single classifiers. Despite the minimal application of deep learning models in credit scoring literature, deep learning models such as convolutional neural networks showed better results compared to statistical and classical machine learning models. This literature survey also highlighted limitations in credit scoring literature. The credit scoring literature often ignores exploratory data analysis, omits inclusion of macro-economic variables and does not determine correlation between the target variable and the independent features, to mention a few. Furthermore, in this survey we proposed a guiding machine learning framework for analysts to perform credit scoring. This framework includes feature selection methods, data balancing technique, models to

consider for bench-marking, best performing models in credit scoring, relevant evaluation metrics and a method for making models transparent. This survey pointed to emerging directions in credit scoring such as the use of techniques that make model predictions explainable and the applications of deep learning models in credit scoring.

Future research should focus more on balancing classes of datasets in credit scoring. Balancing techniques that neither do over-sampling nor under-sampling such as linear dependence approach and wavelet data transformation should be explored in credit scoring. Future research should incorporate macro-economic variables such as interest rates, unemployment rates and inflation rates. An increase in any of the mentioned macro-economic variables may increase the risk of a borrower defaulting. Future studies should consider the time it takes for borrowers to default and this will allow commercial banks to be forward-looking and proactive. The literature does not take into consideration the exploratory data analysis, hence future research should focus on this aspect to better understand for example the distributions of features. Future studies should focus more on time/model complexity to allow efficiency in model development in credit scoring. Future research should determine the correlation between the target variable and the independent variables/features in-order to identify predictive variables/features. There are few studies that focused on using heterogeneous base classifiers for ensemble methods [113,145,147]. However, future studies should focus more on using heterogeneous instead of homogeneous base classifiers in ensemble methods to allow diversity of base classifiers.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the Bankseta, South Africa (The Sector Education and Training Authority (SETA) for the banking industry) for making funds available to Xolani Dastile's PhD study. The authors also would like to thank the anonymous reviewers for providing valuable feedback on initial versions of this paper.

References

[1] L.C. Thomas, J. Crook, D. Edelman, Credit Scoring and Its Applications, Society for Industrial and Applied Mathematics, 2002.
[2] L.C. Thomas, A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers, Int. J. Forecast. 16 (2) (2000) 149–172.
[3] N. Siddiqi, Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring, SAS Publishing, 2005.
[4] I.J. Myung, Tutorial on maximum likelihood estimation, J. Math. Psych. 47 (1) (2003) 90–100.
[5] B. Baesens, R. Setiono, C. Mues, J. Vanthienen, Using neural network rule extraction and decision tables for credit-risk evaluation, Manage. Sci. 49 (3) (2003) 312–329.
[6] H.A. Alaka, L.O. Oyedele, H.A. Owolabi, V. Kumar, S.O. Ajayi, O.O. Akinade, M. Bilal, Systematic review of bankruptcy prediction models: Towards a framework for tool selection, Expert Syst. Appl. 94 (2018) 164–184.
[7] R. Schlosser, Appraising the Quality of Systematic Reviews, Focus: Technical Briefs 17, 2007, pp. 1–8.
[8] J.L. Bellovary, D.E. Giacomino, M.D. Akers, A review of bankruptcy prediction studies: 1930 to present, J. Financial Educ. 33 (2007) 1–42.
[9] H.A. Abdou, J. Pointon, Credit scoring, statistical techniques and evaluation criteria: A review of the literature, Int. J. Intell. Syst. Account. Financ. Manage. 18 (2–3) (2011) 59–88.
[10] W. Lin, Y. Hu, C. Tsai, Machine learning in financial crisis prediction: A survey, IEEE Trans. Syst. Man Cybern. C 42 (4) (2012) 421–436.
[11] X. Wang, M. Xu, Ö.T. Pusatli, A survey of applying machine learning techniques for credit rating: existing models and open issues, in: S. Arik, T. Huang, W.K. Lai, Q. Liu (Eds.), Neural Information Processing, Springer International Publishing, 2015, pp. 122–132.
[12] F. Louzada, A. Ara, G.B. Fernandes, Classification methods applied to credit scoring: Systematic review and overall comparison, Surv. Oper. Res. Manag. Sci. 21 (2) (2016) 117–134.
[13] S.S. Devi, Y. Radhika, A Survey on Machine Learning and Statistical Techniques in Bankruptcy Prediction, 2018.
[14] D. Liang, C.-F. Tsai, H.-T. Wu, The effect of feature selection on financial distress prediction, Knowl.-Based Syst. 73 (2015) 289–297.
[15] I. Brown, C. Mues, An experimental comparison of classification algorithms for imbalanced credit scoring data sets, Expert Syst. Appl. 39 (3) (2012) 3446–3453.
[16] K. Bijak, L.C. Thomas, Does segmentation always improve model performance in credit scoring? Expert Syst. Appl. 39 (3) (2012) 2433–2442.
[17] F.-L. Chen, F.-C. Li, Combination of feature selection approaches with SVM in credit scoring, Expert Syst. Appl. 37 (7) (2010) 4902–4909.
[18] W. Chen, L. Shi, Credit scoring with F-score based on support vector machine, in: Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer, MEC, 2013, pp. 1512–1516.
[19] H. Chen, Y. Xiang, The study of credit scoring model based on group lasso, Procedia Comput. Sci. 122 (2017) 677–684, 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.
[20] B.-W. Chi, C.-C. Hsu, A hybrid approach to integrate genetic algorithm into dual scoring model in enhancing the performance of credit scoring model, Expert Syst. Appl. 39 (3) (2012) 2650–2661.
[21] B. Back, T. Laitinen, K. Sere, Neural networks and genetic algorithms for bankruptcy predictions, Expert Syst. Appl. 11 (4) (1996) 407–413.
[22] S. Oreski, G. Oreski, Genetic algorithm-based heuristic for feature selection in credit risk assessment, Expert Syst. Appl. 41 (4, Part 2) (2014) 2052–2064.
[23] Q. Song, H. Jiang, J. Liu, Feature selection based on FDA and F-score for multi-class classification, Expert Syst. Appl. 81 (2017) 22–27.
[24] Z. Pawlak, Rough set approach to knowledge-based decision support, European J. Oper. Res. 99 (1) (1997) 48–57.
[25] J. Wang, K. Guo, S. Wang, Rough set and tabu search based feature selection for credit scoring, Procedia Comput. Sci. 1 (1) (2010) 2425–2432, ICCS 2010.
[26] Q. Zhang, Q. Xie, G. Wang, A survey on rough set theory and its applications, CAAI Trans. Intell. Technol. 1 (4) (2016) 323–333.
[27] C.-F. Tsai, Feature selection in bankruptcy prediction, Knowl.-Based Syst. 22 (2) (2009) 120–127.
[28] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1996.
[29] V. Kozeny, Genetic algorithms for credit scoring: Alternative fitness function performance comparison, Expert Syst. Appl. 42 (6) (2015) 2998–3004.
[30] M. Crepinsek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: A survey, ACM Comput. Surv. 45 (2013) 35:1–35:33.
[31] S.-h. Liu, M. Mernik, B. Bryant, To explore or to exploit: An entropy-driven approach for evolutionary algorithms, KES J. 13 (2009) 185–206.
[32] J.M. Cadenas, M.C. Garrido, R. Martínez, Feature subset selection filter–wrapper based on low quality data, Expert Syst. Appl. 40 (16) (2013) 6241–6252.
[33] R. Tibshirani, Regression shrinkage and selection via the lasso: a retrospective, J. R. Stat. Soc. Ser. B Stat. Methodol. 73 (3) (2011) 273–282.
[34] A. Zheng, A. Casari, Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists, first ed., O'Reilly Media, Inc., 2018.
[35] S. Sehgal, H. Singh, M. Agarwal, V. Bhasker, Shantanu, Data analysis using principal component analysis, in: 2014 International Conference on Medical Imaging, M-Health and Emerging Communication Systems, MedCom, 2014, pp. 45–48.
[36] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (2) (1936) 179–188.
[37] A.M. Martinez, A.C. Kak, PCA versus LDA, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 228–233.
[38] C. Rao, The utilization of multiple measurements in problems of biological classification, J. R. Stat. Soc. (1948) 159–203.
[39] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley, 2001.
[40] D. Reynolds, Gaussian mixture models, in: Encyclopedia of Biometrics, Springer US, Boston, MA, 2015, pp. 827–832.
[41] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B Stat. Methodol. 39 (1) (1977) 1–38.
[42] W.E. Henley, D.J. Hand, A *k*-nearest-neighbour classifier for assessing consumer credit risk, J. R. Stat. Soc. 45 (1) (1996) 77–95.

[43] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[44] B. Schölkopf, The kernel trick for distances, in: Proceedings of the 13th International Conference on Neural Information Processing Systems, MIT Press, 2000, pp. 283–289.
[45] T.M. Mitchell, Machine Learning, first ed., McGraw-Hill, Inc., New York, NY, USA, 1997.
[46] F. Barboza, H. Kimura, E. Altman, Machine learning models and bankruptcy prediction, Expert Syst. Appl. 83 (2017) 405–417.
[47] C.-F. Tsai, Y.-F. Hsu, D.C. Yen, A comparative study of classifier ensembles for bankruptcy prediction, Appl. Soft Comput. 24 (2014) 977–984.
[48] C.-F. Tsai, J.-W. Wu, Using neural network ensembles for bankruptcy prediction and credit scoring, Expert Syst. Appl. 34 (4) (2008) 2639–2649.
[49] M.D. Odom, R. Sharda, A neural network model for bankruptcy prediction, in: 1990 IJCNN International Joint Conference on Neural Networks, vol. 2, 1990, pp. 163–168.
[50] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[51] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. System Sci. 55 (1) (1997) 119–139.
[52] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–794.
[53] J. Nobre, R. Neves, Combining principal component analysis, discrete wavelet transform and xgboost to trade in the financial markets, Expert Syst. Appl. 125 (2019).
[54] Y. Xia, C. Liu, Y. Li, N. Liu, A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring, Expert Syst. Appl. 78 (2017).
[55] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[56] L. Yu, R. Zhou, L. Tang, R. Chen, A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data, Appl. Soft Comput. 69 (2018) 192–202.
[57] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[58] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[59] Y. Seo, K. shik Shin, Hierarchical convolutional neural networks for fashion image classification, Expert Syst. Appl. 116 (2019) 328–339.
[60] Y. Lecun, Y. Bengio, The Handbook of Brain Theory and Neural Networks, MIT Press, 1995, chapter Convolutional networks for images, speech, and time-series.
[61] F.F. Ting, Y.J. Tan, K.S. Sim, Convolutional neural network improvement for breast cancer classification, Expert Syst. Appl. 120 (2019) 103–115.
[62] O.B. Sezer, A.M. Ozbayoglu, Algorithmic financial trading with deep convolutional neural networks: time series to image conversion approach, Appl. Soft Comput. 70 (2018) 525–538.
[63] B. Zhao, H. Lu, S. Chen, J. Liu, D. Wu, Convolutional neural networks for time series classification, J. Syst. Eng. Electron. 28 (2017) 162–169.
[64] F. Chollet, Deep Learning with Python, first ed., Manning Publications Co., Greenwich, CT, USA, 2017.
[65] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[66] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Inc., New York, NY, USA, 1995.
[67] J.M. Tomczak, M. Zieba, Classification restricted Boltzmann machine for comprehensible credit scoring model, Expert Syst. Appl. 42 (4) (2015) 1789–1796.
[68] G. Douzas, F. Bacao, Self-organizing map oversampling (SOMO) for imbalanced data set learning, Expert Syst. Appl. 82 (2017) 40–52.
[69] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.
[70] R. Saia, S. Carta, G. Fenu, A Wavelet-Based Data Analysis to Credit Scoring, 2018.
[71] J. Nobre, R. Neves, Combining principal component analysis, discrete wavelet transform and xgboost to trade in the financial markets, Expert Syst. Appl. 125 (2019).
[72] R. Saia, S. Carta, A Linear-Dependence-Based Approach to Design Proactive Credit Scoring Models, 2016.
[73] R. Setiono, H. Liu, Symbolic representation of neural networks, Computer 29 (3) (1996) 71–77.
[74] M.W. Craven, J.W. Shavlik, Extracting tree-structured representations of trained networks, in: Proceedings of the 8th International Conference on Neural Information Processing Systems, MIT Press, 1995, pp. 24–30.
[75] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": Explaining the predictions of any classifier, CoRR abs/1602.04938 (2016).
[76] R.A. Eisenbeis, Problems in applying discriminant analysis in credit scoring models, J. Bank. Financ. 2 (3) (1978) 205–219.
[77] G.H. John, P. Langley, Estimating continuous distributions in Bayesian classifiers, in: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.
[78] L. Yu, X. Yao, S. Wang, K. Lai, Credit risk evaluation using a weighted least squares SVM classifier with design of experiment for parameter selection, Expert Syst. Appl. 38 (12) (2011) 15392–15399.
[79] Y. Jiang, Credit scoring model based on the decision tree and the simulated annealing algorithm, in: 2009 WRI World Congress on Computer Science and Information Engineering, Vol. 4, 2009, pp. 18–22.
[80] R. Setiono, H. Liu, Neurolinear: From neural networks to oblique decision rules, Neurocomputing 17 (1) (1997) 1–24.
[81] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, N. Elhadad, Intelligible models for healthCare: Predicting pneumonia risk and hospital 30-day readmission, in: KDD '15, 2015.
[82] S. Lundberg, S. Lee, A unified approach to interpreting model predictions, CoRR abs/1705.07874 (2017).
[83] J.H. Friedman, Greedy function approximation: A gradient boosting machine, Ann. Statist. 29 (2000) 1189–1232.
[84] C. Luo, D. Wu, D. Wu, A deep learning approach for credit scoring using credit default swaps, Eng. Appl. Artif. Intell. 65 (2017) 465–470.
[85] S. Ramasamy, K. Rajaraman, A hybrid meta-cognitive restricted Boltzmann machine classifier for credit scoring, in: TENCON 2017 - 2017 IEEE Region 10 Conference, 2017, pp. 2313–2318.
[86] K. Tran, T. Duong, Q. Ho, Credit scoring model: A combination of genetic programming and deep learning, in: 2016 Future Technologies Conference, FTC, 2016, pp. 145–149.
[87] S.H. Yeh, C.J. Wang, M.F. Tsai, Deep belief networks for predicting corporate defaults, in: 2015 24th Wireless and Optical Communication Conference, WOCC, 2015, pp. 159–163.
[88] V. Neagoe, A. Ciotec, G. Cucu, Deep convolutional neural networks versus multilayer perceptron for financial prediction, in: 2018 International Conference on Communications, COMM, 2018, pp. 201–206.
[89] S. Hamori, M. Kawai, T. Kume, Y. Murakami, C. Watanabe, Ensemble learning or deep learning? Application to default risk analysis, J. Risk Financial Manag. 11 (1) (2018).
[90] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 60.
[91] A. Gómez-Ríos, S. Tabik, J. Luengo, A.S.M. Shihavuddin, B. Krawczyk, F. Herrera, Towards highly accurate coral texture images classification using deep convolutional neural networks and data augmentation, CoRR abs/1804.00516 (2018).
[92] H. Kvamme, N. Sellereite, K. Aas, S. Sjursen, Predicting mortgage default using convolutional neural networks, Expert Syst. Appl. 102 (2018).
[93] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, Neural Inf. Process. Syst. 25 (2012).
[94] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, CoRR (2017).
[95] J. Salamon, J.P. Bello, Deep convolutional neural networks and data augmentation for environmental sound classification, CoRR (2016) http://arxiv.org/abs/1608.04363.
[96] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, Synthetic data augmentation using GAN for improved liver lesion classification, CoRR (2018).
[97] B. Zhu, W. Yang, H. Wang, Y. Yuan, A hybrid deep learning model for consumer credit scoring, in: 2018 International Conference on Artificial Intelligence and Big Data, ICAIBD, 2018, pp. 205–208.
[98] M.F. Kiani, F. Mahmoudi, A new hybrid method for credit scoring based on clustering and support vector machine (ClsSVM), in: 2010 2nd IEEE International Conference on Information and Financial Engineering, 2010, pp. 585–589.
[99] D. Zhang, X. Zhou, S.C. Leung, J. Zheng, Vertical bagging decision trees model for credit scoring, Expert Syst. Appl. 37 (12) (2010) 7838–7843.
[100] M.A.H. Farquad, V. Ravi, Sriramjee, G. Praveen, Credit scoring using PCA-SVM hybrid model, in: V.V. Das, J. Stephen, Y. Chaba (Eds.), Computer Networks and Information Technologies, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 249–253.
[101] Y. Ping, L. Yongheng, Neighborhood rough set and SVM based hybrid credit scoring classifier, Expert Syst. Appl. 38 (9) (2011) 11300–11304.
[102] J. Wang, A.-R. Hedar, S. Wang, J. Ma, Rough set and scatter search metaheuristic based feature selection for credit scoring, Expert Syst. Appl. 39 (6) (2012) 6123–6128.
[103] L. Han, L. Han, H. Zhao, Orthogonal support vector machine for credit scoring, Eng. Appl. Artif. Intell. 26 (2) (2013) 848–862.
[104] J. Shi, S.-y. Zhang, L.-m. Qiu, Credit scoring by feature-weighted support vector machines, J. Zhejiang Univ. Sci. C 14 (3) (2013) 197–204.
[105] Q. Li, J. Zhang, Y. Wang, K. Kang, Credit risk classification using discriminative restricted boltzmann machines, in: 2014 IEEE 17th International Conference on Computational Science and Engineering, 2014, pp. 1697–1700.

[106] S. Maldonado, J. Pérez, C. Bravo, Cost-based feature selection for support vector machines: An application in credit scoring, European J. Oper. Res. 261 (2) (2017) 656–665.
[107] H. Sutrisno, S. Halim, Credit scoring refinement using optimized logistic regression, in: 2017 International Conference on Soft Computing, Intelligent System and Information Technology, ICSIIT, 2017, pp. 26–31.
[108] R.A. Mancisidor, M. Kampffmeyer, K. Aas, R. Jenssen, Segment-based credit scoring using latent clusters in the variational autoencoder, 2018, arXiv:1806.02538.
[109] X. Zhang, Y. Yang, Z. Zhou, A novel credit scoring model based on optimized random forest, in: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference, CCWC, 2018, pp. 60–65.
[110] S. Jadhav, H. He, K. Jenkins, Information gain directed genetic algorithm wrapper feature selection for credit rating, Appl. Soft Comput. 69 (2018) 541–553.
[111] G. Dong, K.K. Lai, J. Yen, Credit scorecard based on logistic regression with random coefficients, Procedia Comput. Sci. 1 (1) (2010) 2463–2468, ICCS 2010.
[112] B. Twala, Multiple classifier application to credit risk assessment, Expert Syst. Appl. 37 (4) (2010) 3326–3336.
[113] N.-C. Hsieh, L.-P. Hung, A data driven ensemble classifier for credit scoring analysis, Expert Syst. Appl. 37 (1) (2010) 534–545.
[114] L. Yu, X. Yao, S. Wang, K. Lai, Credit risk evaluation using a weighted least squares SVM classifier with design of experiment for parameter selection, Expert Syst. Appl. 38 (12) (2011) 15392–15399.
[115] G. Wang, J. Hao, J. Ma, H. Jiang, A comparative assessment of ensemble learning for credit scoring, Expert Syst. Appl. 38 (1) (2011) 223–230.
[116] Q. Wang, K.K. Lai, D. Niu, Green credit scoring system and its risk assessemt model with support vector machine, in: 2011 Fourth International Joint Conference on Computational Sciences and Optimization, 2011, pp. 284–287.
[117] B. Ribeiro, N. Lopes, Deep belief networks for financial prediction, in: B.-L. Lu, L. Zhang, J. Kwok (Eds.), Neural Information Processing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 766–773.
[118] B.W. Yap, S.H. Ong, N.H.M. Husain, Using data mining to improve assessment of credit worthiness via credit scoring models, Expert Syst. Appl. 38 (10) (2011) 13274–13283.
[119] F. Louzada, O. Anacleto-Junior, C. Candolo, J. Mazucheli, Poly-bagging predictors for classification modelling for credit scoring, Expert Syst. Appl. 38 (10) (2011) 12717–12720.
[120] A. Marqués, V. García, J. Sánchez, Exploring the behaviour of base classifiers in credit scoring ensembles, Expert Syst. Appl. 39 (11) (2012) 10244–10250.
[121] A. Marqués, V. García, J. Sánchez, Two-level classifier ensembles for credit risk assessment, Expert Syst. Appl. 39 (12) (2012) 10916–10922.
[122] B. Tang, S. Qiu, A new credit scoring method based on improved fuzzy support vector machine, in: 2012 IEEE International Conference on Computer Science and Automation Engineering, CSAE, Vol. 3, 2012, pp. 73–75.
[123] F. Louzada, P.H. Ferreira-Silva, C.A. Diniz, On the impact of disproportional samples in credit scoring models: An application to a Brazilian bank data, Expert Syst. Appl. 39 (9) (2012) 8071–8078.
[124] J. Abellán, C.J. Mantas, Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring, Expert Syst. Appl. 41 (8) (2014) 3825–3830.
[125] T. Harris, Credit scoring using the clustered support vector machine, Expert Syst. Appl. 42 (2) (2015) 741–750.
[126] B. Yi, J. Zhu, Credit scoring with an improved fuzzy support vector machine based on grey incidence analysis, in: 2015 IEEE International Conference on Grey Systems and Intelligent Services, GSIS, 2015, pp. 173–178.
[127] S. Jones, D. Johnstone, R. Wilson, An empirical evaluation of the performance of binary classifiers in the prediction of credit ratings changes, J. Bank. I Finance 56 (2015) 72–85.
[128] J. Chen, L. Xu, A method of improving credit evaluation with support vector machines, in: 2015 11th International Conference on Natural Computation, ICNC, 2015, pp. 615–619.
[129] Z. Zhao, S. Xu, B.H. Kang, M.M.J. Kabir, Y. Liu, R. Wasinger, Investigation and improvement of multi-layer perceptron neural networks for credit scoring, Expert Syst. Appl. 42 (7) (2015) 3508–3516.
[130] R. Florez-Lopez, J.M. Ramon-Jeronimo, Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal, Expert Syst. Appl. 42 (13) (2015) 5737–5753.
[131] R. Florez-Lopez, J.M. Ramon-Jeronimo, Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal, Expert Syst. Appl. 42 (13) (2015) 5737–5753.
[132] M. Aláraj, M. Abbod, A systematic credit scoring model based on heterogeneous classifier ensembles, in: 2015 International Symposium on Innovations in Intelligent SysTems and Applications, INISTA, 2015, pp. 1–7.
[133] M. Aláraj, M.F. Abbod, Classifiers consensus system approach for credit scoring, Knowl.-Based Syst. 104 (2016) 89–105.
[134] L. Yu, Z. Yang, L. Tang, A novel multistage deep belief network based extreme learning machine ensemble learning paradigm for credit risk assessment, Flex. Serv. Manuf. J. 28 (4) (2016) 576–592.
[135] H. Xiao, Z. Xiao, Y. Wang, Ensemble classification based on supervised clustering for credit scoring, Appl. Soft Comput. 43 (2016) 73–86.
[136] A. Bequé, S. Lessmann, Extreme learning machines for credit scoring: An empirical evaluation, Expert Syst. Appl. 86 (2017) 42–53.
[137] A. Lawi, F. Aziz, S. Syarif, Ensemble gradientboost for increasing classification accuracy of credit scoring, in: 2017 4th International Conference on Computer Applications and Information Processing Technology, CAIPT, 2017, pp. 1–4.
[138] Y. Li, X. Lin, X. Wang, F. Shen, Z. Gong, Credit risk assessment algorithm using deep neural networks with clustering and merging, in: 2017 13th International Conference on Computational Intelligence and Security, CIS, 2017, pp. 173–176.
[139] Z. Li, Y. Tian, K. Li, F. Zhou, W. Yang, Reject inference in credit scoring using semi-supervised support vector machines, Expert Syst. Appl. 74 (2017) 105–114.
[140] O.J. Okesola, K.O. Okokpujie, A.A. Adewale, S.N. John, O. Omoruyi, An improved bank credit scoring model: A Naïve Bayesian approach, in: 2017 International Conference on Computational Science and Computational Intelligence, CSCI, 2017, pp. 228–233.
[141] H. Chen, M. Jiang, X. Wang, Bayesian ensemble assessment for credit scoring, in: 2017 4th International Conference on Industrial Economics System and Industrial Security Engineering, IEIS, 2017, pp. 1–5.
[142] J. Abellán, J.G. Castellano, A comparative study on base classifiers in ensemble methods for credit scoring, Expert Syst. Appl. 73 (2017) 1–10.
[143] B. Vanderheyden, J. Priestley, Logistic ensemble models, 2018.
[144] P. Martey Addo, D. Guegan, B. Hassani, Credit risk analysis using machine and deep learning models, Risks 6 (2018) 38.
[145] Y. Xia, C. Liu, B. Da, F. Xie, A novel heterogeneous ensemble credit scoring model based on bstacking approach, Expert Syst. Appl. 93 (C) (2018) 182–199.
[146] Y.-C. Chang, K.-H. Chang, G.-J. Wu, Application of extreme gradient boosting trees in the construction of credit risk assessment models for financial institutions, Appl. Soft Comput. 73 (2018).
[147] W. Li, S. Ding, Y. Chen, S. Yang, Heterogeneous ensemble for default prediction of peer-to-peer lending in China, IEEE Access 6 (2018) 54396–54406.
[148] A. Cao, H. He, Z. Chen, W. Zhang, Performance evaluation of machine learning approaches for credit scoring, Int. J. Econ. Finance Manag. Sci. 6 (2018) 255–260.
[149] Basel Committee on Banking Supervision, Basel II: International convergence of capital measurement and capital standards: A revised framework - comprehensive version, bank for international settlements, BIS (2006).