

Bankruptcy Prediction Using Machine Learning

Nanxi Wang

Shanghai Starriver Bilingual School, Shanghai, China

Email: nanxi117@outlook.com

How to cite this paper: Wang, N.X. (2017) Bankruptcy Prediction Using Machine Learning. *Journal of Mathematical Finance*, 7, 908-918.

<https://doi.org/10.4236/jmf.2017.74049>

Received: August 30, 2017

Accepted: November 14, 2017

Published: November 17, 2017

Copyright © 2017 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

With improved machine learning models, studies on bankruptcy prediction show improved accuracy. This paper proposes three relatively newly-developed methods for predicting bankruptcy based on real-life data. The result shows among the methods (support vector machine, neural network with dropout, autoencoder), neural network with added layers with dropout has the highest accuracy. And a comparison with the former methods (logistic regression, genetic algorithm, inductive learning) shows higher accuracy.

Keywords

Support Vector Machine, Autoencoder, Neural Network, Bankruptcy, Machine Learning

1. Introduction

Machine learning is a subfield of computer science. It allows computers to build analytical models of data and find hidden insights automatically, without being unequivocally coded. It has been applied to a variety of aspects in modern society, ranging from DNA sequences classification, credit card fraud detection, robot locomotion, to natural language processing. It can be used to solve many types of tasks such as classification. Bankruptcy prediction is a typical example of classification problems.

Machine learning was born from pattern recognition. Earlier works of the same topic (machine learning in bankruptcy) use models including logistic regression, genetic algorithm, and inductive learning.

Logistic regression is a statistical method allowing researchers to build predictive function based on a sample. This model is best used for understanding how several independent variables influence a single outcome variable [1]. Though useful in some ways, logistic regression is also limited.

Genetic algorithm is based on natural selection and evolution. It can be used

to extract rules in propositional and first-order logic, and to choose the appropriate sets of if-then rules for complicated classification problems [2].

Inductive learning's main category is decision tree algorithm. It identifies training data or earlier knowledge patterns and then extracts generalized rules which are then used in problem solving [2].

To see if the accuracy of bankruptcy prediction can be further improved, we propose three latest models—support vector machine (SVM), neural network, and autoencoder.

Support vector machine is a supervised learning method which is especially effective in cases of high dimensions, and is memory efficient because it uses a subset of training points in the decision function. Also, it specifies kernel functions according to the decision function [3]. Its nice math property guarantees a simple convex optimization problem to converge to a single global problem.

Neural networks, unlike conventional computers, are expressive models that learn by examples. They contain multiple hidden layers, thus are capable of learning very complicated relationships between inputs and outputs. And they operate significantly faster than conventional techniques. However, due to limited training data, overfitting will affect the ultimate accuracy. To prevent this, a technique called dropout—temporarily and randomly removes units (hidden and visible)—to the neural network [4].

Autoencoder, also known as Diabolo network, is an unsupervised learning algorithm that sets the target values to be equal to the inputs. By doing this, it suppresses the computation of representing a few functions, which improves accuracy. Also, the amount of training data required to learn these functions is reduced [5].

This paper is structured as follows. Section 2 describes the motivation for this idea. Section 3 describes relevant previous work. Section 4 formally describes the three models. In Section 5 we present our experimental results where we do a parallel comparison within the three models we choose and a longitudinal comparison with the three older models. Section 6 is the conclusion. Section 7 is the reference.

2. Motivation

The three models we choose (SVM, neural network, autoencoder) are relatively newly-developed but have already been applied to many fields.

SVM has been used successfully in many real-world problems such as text categorization, object tracking, and bioinformatics (Protein classification, Cancer classification). Text categorization is especially helpful in daily life—web searching and email filtering provide huge convenience and work efficiency.

Neural networks learn by examples instead of algorithms, thus, they have been widely applied to problems where it is hard or impossible to apply algorithmic methods [6]. For instance, finger print recognition is an exciting application. People can now use their unique fingerprints as keys to unlock their phones and payment accounts, free from the troubling, long passwords.

Autoencoders are especially successful in solving difficult tasks like natural language processing (NLP). They have been used to solve the previous seemingly intractable problems in NLP, including word embeddings, machine translation, document clustering, sentiment analysis, and paraphrase detection.

However, the usage of the three models in economics or finance is comparatively hard to find. So, we aim to find out if they still work well in economical field by running them with real-life data in a predicting bankruptcy task.

Another motivation is finding out if the accuracy of this particular problem (bankruptcy prediction) can be improved after reading previous works—The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms [2], and Predicting Bankruptcy with Robust Logistic Regression [1]—which uses older models. Thus, a comparison of the models and results is included in this paper.

3. Related Work

Machine learning enables computers to find insights from data automatically. The idea of using machine learning to predict bankruptcy has previously been used in the context of Predicting Bankruptcy with Robust Logistic Regression by Richard P. Hauser and David Booth [1]. This paper uses robust logistic regression which finds the maximum trimmed correlation between the samples remained after removing the overly large samples and the estimated model using logistic regression [1]. This model has its limitation. The value of this technique relies heavily on researchers' abilities to include the correct independent variables. In other words, if researchers fail to identify all the relevant independent variables, logistic regression will have little predictive value [7]. Its overall accuracy is 75.69% in the training set and 69.44% in testing set.

Another work, the discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms, in 2003 by Myoung-Jong Kim and Ingoo Han uses the same dataset as we do. They apply older models—inductive learning algorithms (decision tree), genetic algorithms, and neural networks without dropout. Since the length of genomes in GA is fixed, a given problem cannot easily be encoded. And GA gives no guarantee of finding the global maxima. The problem of inductive learning is with the one-step-ahead node splitting without backtracking, which may generate a suboptimal tree. Also, decision trees can be unstable because small variations in the data might result in a completely different tree being generated [3]. And the absence of dropout in the neural network model increases the possibility of overfitting which affects accuracy. The overall accuracies are 89.7%, 94.0%, and 90.3% respectively.

The models we choose either contain a newly developed technique, like dropout, or completely new models that have hardly been utilized in bankruptcy prediction.

4. Model Description

This section describes the proposed three models.

4.1. Support Vector Machine

Specifically, we use support vector classify (SVC), a subcategory of SVM, in this task. It constructs a hyper-plane, as shown in **Figure 1**, in a high dimensional space which is used for classification. Generally, a good separation represented by the solid line in **Figure 1** means the distance (the space between the dotted lines) to the nearest training data points (the red and blue dots) of any class (represented by the color red and blue) is the largest. This is also known as functional margin [3].

With training vectors in two classes and a vector,

$$x_i \in \mathbb{R}^p, i = 1, \dots, n, y \in \{1, -1\}^n$$

respectively, SVM aims at solving the problem:

$$\min_{\omega, b, \zeta} \frac{1}{2} \omega^T \omega + C \sum_{i=1}^n \zeta_i$$

subject to

$$y_i (\omega^T \phi(x_i) + b) \geq 1 - \zeta_i$$

Its dual is

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$

subject to

$$y^T \alpha = 0, 0 \leq \alpha_i \leq C, i = 1, \dots, n$$

where e is a common vector, $C > 0$ is upper bound, Q is n by n positive semi-definite matrix, $Q_{ij} \equiv y_i y_j k(x_i \cdot x_j)$, and $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the kernel.

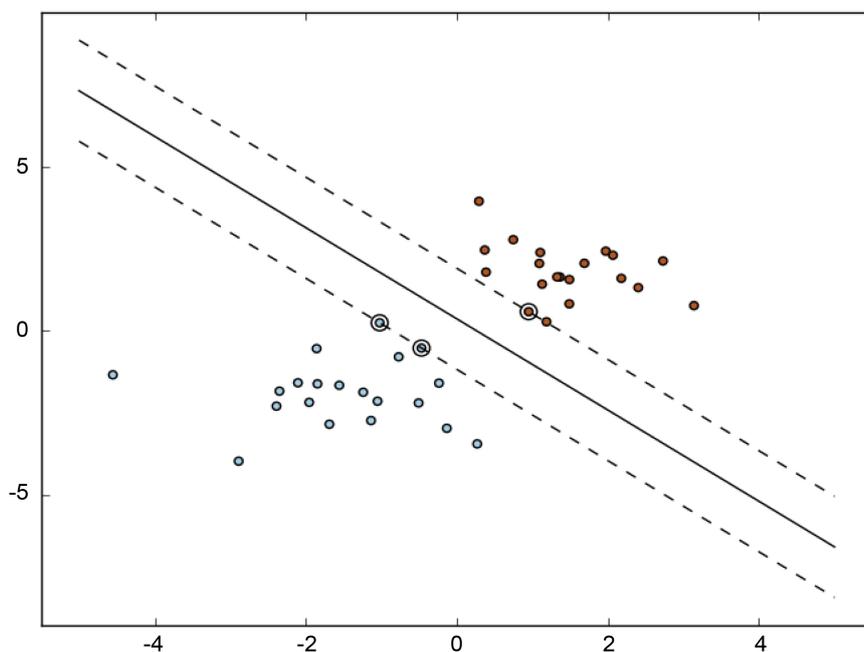


Figure 1. SVM model [3].

Here the function implicitly maps the training vectors into a higher dimensional space.

The decision function is:

$$\text{sgn}\left(\sum_{i=1}^n y_i \alpha_i K(x_i, x) + \rho\right) \quad [3]$$

4.2. Neural Network with Dropout

Neural networks' inputs are modelled as layers of neurons. Its structure is shown in the following figure.

As shown in **Figure 1**, the formal neuron uses n inputs x_1, x_2, \dots, x_n to classify the signals coming from dendrites, and are then synoptically weighted correspondingly with w_1, w_2, \dots, w_n that measure their permeabilities. Then, the excitation level of the neuron is calculated as the weighted sum of input values:

$$\xi = \sum_{i=1}^n w_i x_i$$

f in **Figure 2** represents activation function.

When the value of excitation level x reaches the threshold h , the output y (state) of the neuron is induced. This simulates the electric impulse generated by axon [8].

Dropout is a technique that further improves neural network's accuracy. In **Figure 3**, let L be the number of hidden layers, $l \in \{1, \dots, L\}$ the hidden layers of the neural network, $z(l)$ and $y(l)$ the vectors of inputs and outputs of layer l , respectively. $W(l)$ and $b(l)$ are the weights and biases at layer l . For $l \in \{0, \dots, L-1\}$ and any hidden unit i , the network then can be described as:

$$z^{(l+1)} = w^{(l+1)} y^l + b^{(l+1)}, \text{ iii}$$

$$y^{(l+1)} = f(z^{(l+1)}), \text{ ii}$$

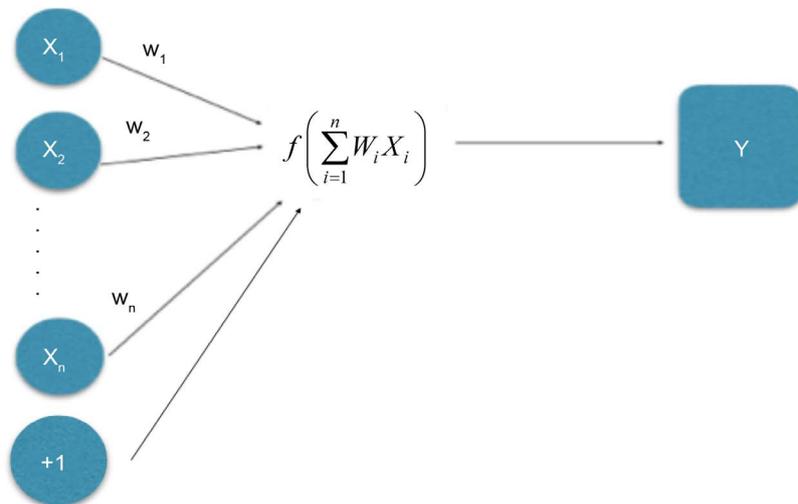


Figure 2. Neural network model.

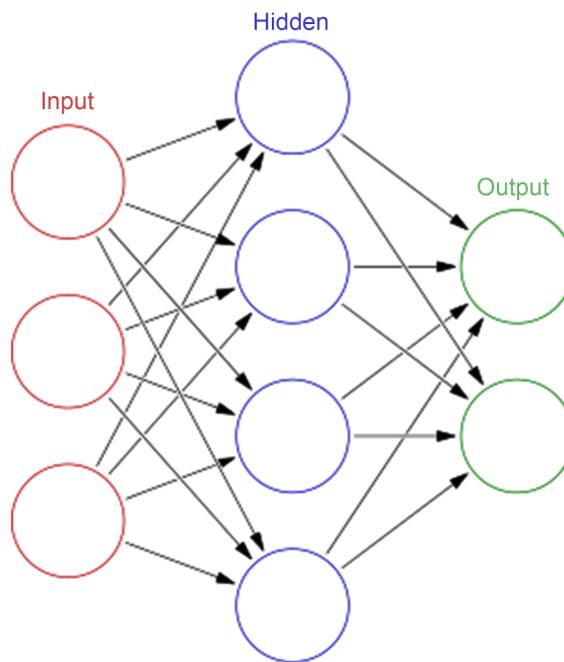


Figure 3. Artificial neural network.

where f is any activation function.

With dropout, the feed-forward operation becomes:

$r^{(l)}$ -Bernoulli(p), j

$$y^{(l)} = r^{(l)} y^{(l)},$$

$$z^{(l+1)} = w^{(l+1)} y^{(l)} + b^{(l+1)}, \text{ iii [4].}$$

4.3. Autoencoder

Consider an $n/p/n$ autoencoder.

In Figure 4, let F and G denote sets, n and p be positive integers where $0 < p < n$, and B be a class of functions from F^n to G^p .

Define $X = \{x_1, \dots, x_m\}$ as a set of training vectors in F^n . When there are external targets, let $Y = \{y_1, \dots, y_m\}$ denote the corresponding set of target vectors in F^n . And Δ is a distortion function (e.g. Lp norm, Hamming distance) defined over F^n .

For any $A \in A$ and $B \in B$, the input vector $x \in F^n$ becomes output vector $A \circ B(x) \in F^n$ through the autoencoder. The goal is to find $A \in A$ and $B \in B$ that minimize the overall distortion function:

$$\min E(A, B) = \min E(x_i) = \min \Delta(A \circ B(x_i), x_i) \text{ [10].}$$

4.4. Decision Tree

Given training vectors $x_i \in R^n$, $i = 1, \dots, l$ and a label vector $y \in R^l$, a decision tree groups the sample according to the same labels.

Let Q represents the data at node m . The tree partitions the data $\theta = (j, t_m)$

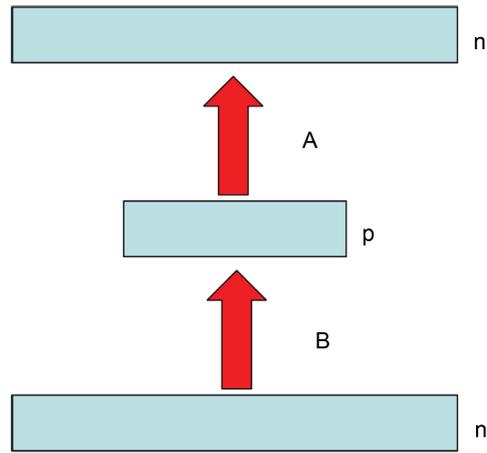


Figure 4. An n/p/n Autoencoder Architecture [Pierre Baldi, 2012].

(feature j and threshold t_m) into $Q_{\text{left}}(\theta)$ and $Q_{\text{right}}(\theta)$ subsets:

$$Q_{\text{left}}(\theta) = (x, y) \mid x_j \leq t_m$$

$$Q_{\text{right}}(\theta) = Q \setminus Q_{\text{left}}(\theta)$$

The impurity function $H(\cdot)$ is used to calculate the impurity at m , the choice of which depends on the task being solved (classification or regression)

$$G(Q, \theta) = \frac{n_{\text{left}}}{N_m} H(Q_{\text{left}}(\theta)) + \frac{n_{\text{right}}}{N_m} H(Q_{\text{right}}(\theta))$$

Choose the parameters that minimises the impurity

$$\theta^* = \arg \min_{\theta} G(Q, \theta)$$

Then recur for subsets $Q_{\text{left}}(\theta^*)$ and $Q_{\text{right}}(\theta^*)$ until reaching the maximum possible depth, $N_m < \min_{\text{samples}}$ or $N_m = 1$ [3].

5. Experimental Result

The data we used shown in **Table 1**, called Qualitative Bankruptcy database, is created by Martin. A, Uthayakumar. j, and Nadarajan. m in February 2014 [10]. The attributes include industrial risk, management risk, financial flexibility, credibility, competitiveness, and operating risk.

5.1. Parallel Comparison

5.1.1. SVM (Linear Kernel)

As shown in **Table 2**, the accuracy increases when truncate increases in a SVM model.

5.1.2. Neural Network (Activation = Softmax, Num_Classes = 2, Optimiser = Adam, Loss = Categorical_Crossentropy, Metrics = Accuracy)

As shown in **Table 3**, when other things in the model hold the same, dropout rate of 0.5 yields the highest accuracy.

Table 1. Dataset Description.

Data set	Dimensionality	Instances	Training Set	Test Set	Validation
Bankruptcy	6 times1	250	80%	10%	10%

Table 2. Accuracy of Neural Network Model with Truncate 50 or 100.

variation	accuracy
truncate = 50	0.9899
truncate = 100	0.9933

Table 3. Accuracy of Neural Network Model with and without Dropout.

variation	accuracy
without dropout	0.9867 with loss 0.0462
with dropout (dropout rate = 0.1)	0.9867 with loss 0.0292
with dropout (dropout rate = 0.3)	0.9933 with loss 0.0300
with dropout (dropout rate = 0.4)	0.9933 with loss 0.0401
with dropout (dropout rate = 0.5)	0.9933 with loss 0.0278
with dropout (dropout rate = 0.7)	0.9933 with loss 0.0428
with dropout (dropout rate = 0.8)	0.9867 with loss 0.0318

As shown in **Table 4** and **Table 5**, we can conclude that adding layers increases accuracy. **Figure 5** and **Figure 6** depict **Table 5**.

5.1.3. Autoencoder (Encoding Dim = 2, Activation = "Relu", Optimizer = "Adam", Lose = "Mse")

As shown in **Table 6**, autoencoder with decision tree yields higher accuracy.

5.2. Longitudinal Comparison

As shown in **Table 7**, neural network with truncate = 100 with added layers with dropout has the highest accuracy. And all the new models have higher accuracy than the old ones.

6. Conclusions

Support vector machine, neural network with dropout, and autoencoder are three relatively new models applied in bankruptcy prediction problems. Their accuracies outperform those of the three older models (robust logistic regression, inductive learning algorithms, genetic algorithms). The improved aspects include the control for overfitting, the improved probability of finding the global maxima, and the ability to handle large feature spaces. This paper compared and concluded the progress of machine leaning models regarding bankruptcy prediction, and checked to see the performance of relatively new models in the context of bankruptcy prediction that have rarely been applied in that field.

However, the three models also have drawbacks. SVM does not directly give probability estimates, but uses an expensive five-fold cross-validation instead.

Table 4. Accuracy of Neural Network Model with Two, Three, and Four Layer.

variation	accuracy
two layer with dropout (dropout rate = 0.5)	0.9933 with loss 0.0278
three layer (added layer with dense 200) with dropout (dropout rate = 0.5)	0.9933 with loss 0.0221
four layer (added layer with dense 16) with dropout (dropout rate = 0.5)	1.0000 with loss 0.0004

Table 5. Accuracy of Neural Network Model with Truncate 50 or 100 and With Four Layers.

variation	accuracy
truncate = 50 with four layers (added layer dense 16,200) with dropout rate 0.5	0.9950 with loss 0.0389
truncate = 100 with four layers (added layer dense 16,200) with dropout rate 0.5	1.0000 with loss 0.0004

Table 6. Accuracy of Neural Network Model with SVM or With Decision Tree.

variation	accuracy
with SVM	0.9867
with decision tree	0.9933

Table 7. Accuracy of Neural Network Model with Different models.

model	accuracy
Robust logistic regression	0.6944
inductive learning algorithms (decision tree)	0.897
genetic algorithms	0.94
neural networks without dropout	0.903
SVM truncate = 100	0.9933
Truncate = 100 with four layers (added layer dense 16,200) with dropout rate 0.5	1.0000 with loss 0.0004
autoencoder (with decision tree)	0.9933

Also, if the data sample is not big enough, especially when outnumbered by the number of features, SVM is likely to give bad performance [4]. With dropout, the time to train the neural network will be 2 to 3 times longer than training a standard neural network. An autoencoder captures as much information as possible, not necessarily the relevant information. And this can be a problem

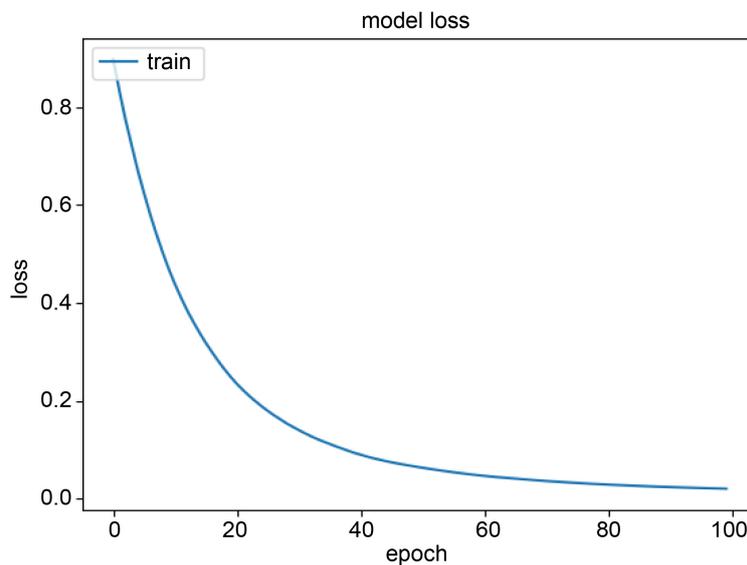


Figure 5. Neural network-loss.

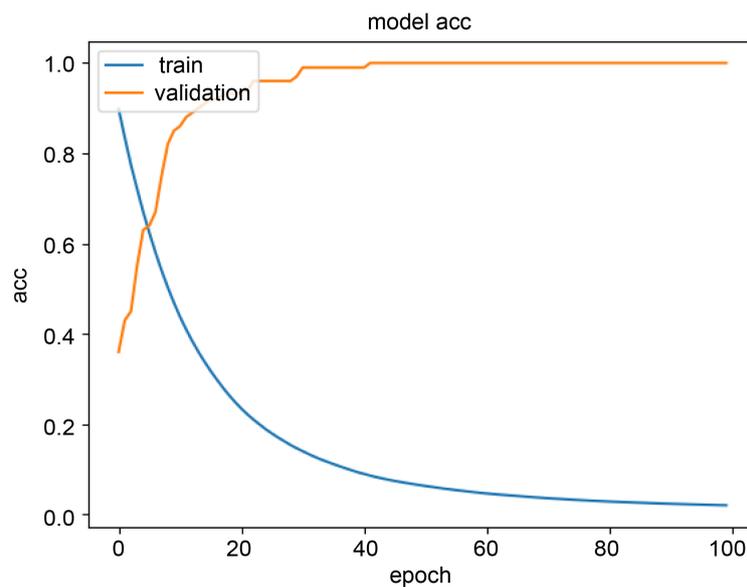


Figure 6. Neural network-accuracy.

when the most relevant information only makes up a small percent of the input. The solutions to overcome these drawbacks are yet to be found.

References

- [1] Hauser, R.P. and Booth, D. (2011) Predicting Bankruptcy with Robust Logistic Regression. *Journal of Data Science*, **9**, 565-584.
- [2] Kim, M.-J. and Han, I. (2003) The Discovery of Experts' Decision Rules from Qualitative Bankruptcy Data Using Genetic Algorithms. *Expert Systems with Application*, **25**, 637-646.
- [3] Pedregosa, et al. (2011) Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**, 2825-2830.

- [4] Sirvastava, N., *et al.* (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, **15**, 1929-1958.
- [5] Dev, D. (2017) Deep Learning with Hadoop. Packet Publishing, Birmingham, 52.
- [6] Nielsen, F. (2001) Neural Networks—Algorithms and Applications. <https://www.mendeley.com/research-papers/neural-networks-algorithms-applications-5/>
- [7] Robinson, N. (n.d.) The Disadvantages of Logistic Regression. <http://classroom.synonym.com/disadvantages-logistic-regression-8574447.html>
- [8] Sima, J. (1998) Introduction to Neural Networks. Technical Report No. 755.
- [9] Baldi, P. (2012) Autoencoders, Unsupervised Learning, and Deep Architectures. *Journal of Machine Learning Research*, **27**, 37-50.
- [10] Martin, A., Uthayakumar, J. and Nadarajan, M. (2014) Qualitative Bankruptcy Data Set, UCI. https://archive.ics.uci.edu/ml/datasets/qualitative_bankruptcy