Deep Learning in Risk Assessment

Janayna M. Fernandes[1], Lucas Z. Bissaro[1], Fernanda M. C. Santos[1] and Murillo G. Carneiro[1]

[1]Faculdade de Computação – Universidade Federal de Uberlândia (UFU)
Uberlândia, MG – Brazil

{jmfernandes,lucas.bissaro,fmcsantos,mgcarneiro}@ufu.br

Abstract. *Credit evaluation models have been largely studied in the accounting and finance literature. With the support of such models, usually developed as part of a data mining process, it is possible to classify the credit applicants more accurately into "good" or "bad" risk groups. Despite many machine learning techniques have been extensively evaluated to this problem, deep learning models have been barely explored yet, although they have provided state-of-the-art results for a myriad of applications. In this paper, we propose deep learning models for the credit evaluation problem. To be specific, we investigate the abilities of deep neural networks (DNN) and convolutional neural networks (CNN) for such a problem and systematically compare their classification accuracy against five commonly adopted techniques on three real-world credit evaluation datasets. The results show that random forest, which is a state-of-the-art technique for such a problem, presented the most consistent performance, although CNN demonstrated a high potential to outperform it in bigger datasets.*

1. Introduction

A large number of institutions such as commercial banks have realized the importance of their databases, which generally cover transactions carried out over several years. [Wei et al. 2015]. These databases can lead to a better understanding of its customers profiles and thus come to collaborate with the quality in offering new products or services. In the same way on financial decision environments substantial amounts of information are drawn from a variety of sources.

The increasing volume of data stored and managed by companies has risen in such way that their treatment overcomes the human ability to understand and efficiently handle it, causing a critical need for tools and techniques that are able to automatically perform efficient data analyses to support companies and individuals in strategic planning and decision making [Zhang and Zhou 2004]. The general process that has this purpose is mainly referenced in the literature as KDD: Knowledge Discovery in Databases.

The reasons for the difficulties in the exploration and analysis of stored data are the large volume of data to be examined and the nature of the relationships themselves that are not trivial. A knowledge discovery tool is needed to assist the decision maker in relation to loan applications. The KDD process provides a variety of useful methods for discovering such relationships in historical data, while ensuring that discovered relationships can be generalized to new (future) data [Tan 2018]. It can be described in three big phases: preprocessing, data mining, and post-processing.

Credit risk assessment is basically a classifier system that studies income and its security. The decision about to offer credit to a customer should be judicious. Providing credit to a prospective customer is determined by a set of features which usually includes personal information such as age, income, solvency, schooling, etc., credit information such as type of credit, maturity, loan value and other points inherent to financial transactions [Chen and Huang 2003]. The purpose of credit risk assessment models is to classify customers as good (accept) or bad (reject) [Hand and Henley 1997]. Furthermore, the knowledge acquired through of KDD can be used by credit managers to help them reject or accept customers. By using these techniques, there would be less risk to financial companies when predicting which customers will succeed in their payments. Consequently, more people could have access to credit loans.

Despite many machine learning techniques have been extensively evaluated in such a knowledge discovery process, e.g., decision tree, support vector machines, neural networks and random forest, just to name a few, investigations about recent techniques, like deep learning models, have been barely explored yet in the literature [Sun and Vasarhelyi 2018]. Due to deep learning models salient features like the abilities to learn high level representations from low ones and to learn from huge amount of data, they have provided state-of-the-art results for a myriad of applications.

In this paper, we propose deep learning models for credit risk assessment problems. To be specific, we investigate the abilities of deep neural networks (DNN) and convolutional neural networks (CNN) for such a problem. Experiments were conducted on three real-world credit evaluation datasets. In order to give a frank account about the predictive performance of the deep learning models, they are systematically compared in terms of predictive performance against five appropriately tuned techniques, including random forest which is a state-of-the-art for credit risk evaluation [Lessmann et al. 2015].

The remainder of the paper is organized as follows. Sect. 2 presents a brief discussion about related works. Sect. 3 describes the data as well as the techniques adopted in this study. Sect. 4 discuss the experimental results and Sect. 5 concludes the paper.

2. Related Work

The literature contains several researches that address the usage of machine learning techniques to provide credit risk evaluation to financial data [Yeh and Lien 2009, Lessmann et al. 2015]. Most of them are strictly related to traditional machine learning techniques, like decision trees, nearest neighbors, neural networks, support vector machines, ensemble methods, and so on. Among such traditional techniques, random forest have been considered the state-of-the-art technique for the credit risk assessment problem [Baesens et al. 2003].

Regarding the usage of deep learning methods, they have been succesfuly applied for problems of negotiation processes, commercial area and financial data in general investment terms. The authors of [Deng et al. 2016] introduced a recurrent deep neural network novel structure consisting of simultaneous environment sensing and recurrent decision making for the problem of online financial assert trading. The technique is composed of two parts: DNN for feature learning and recurrent neural network for reinforcement learning. This is the first paper to implement and demonstrate the effectivenes of deep learning in designing a real trading system for financial signal representation and

self-taught reinforcement trading.

In [Chen et al. 2016], the authors developed a deep learning framework based on CNN to analyze trading time series data. The experimental results showed that the proposed learning system was better than the traditional rule-based trading. The proposed system was implemented and benchmarked in the historical datasets of Taiwan Stock Index Futures.

The authors of [Sun and Vasarhelyi 2018] demonstrate the effectiveness of deep learning in predicting credit card delinquencies. They used real-life credit card data from a Brazil bank and developed a DNN to predict severe delinquencies based on clients' personal information and spending characteristics. They compared the predictive performance of the DNN against neural networks, logistic regression, naive Bayes, and decision tree models. The experimental results showed that DNN generally works better than other models in terms of predictive performance.

The authors [Li et al. 2017] propose a credit risk assessment algorithm using deep neural networks with clustering and merging. As in credit data the classes are extremely imbalanced, the majority class samples are divided into several subgroups by k-means clustering algorithm, each subgroup is merged with the minority class samples to produce several balanced subgroups, and these balanced subgroups are classified using deep neural networks respectively. The experimental results show that the proposed algorithm has a higher prediction accuracy in credit risk assessment.

In summary, the current investigation is also motivated by the works discussed before, which has achieved succesfuly results for domain-related problems through of deep learning techniques.

3. Material and Methods

This paper aims to investigate the usage of deep neural networks to the problem of credit assessment. In the following, we present the data sets under study and the deep neural networks evaluated in this paper.

3.1. Datasets for Credit Assessment

Here we discuss in detail the three credit assessment datasets, available publicly at UCI Machine Learning data repository [Asuncion and Newman 2007], as well as the preparation and pre-processing steps conducted over each one of them.

3.1.1. Australian credit data

This dataset deals with credit card applications. All attribute names and values have been changed by the data owner to meaningless symbols to protect confidentiality of the data. It contains 14 attributes, where six are continuous attributes and eight are categorical attributes. There were also a few missing values, 37 cases (5%) had one or more missing values. The missing values were replaced by the mode of the attribute, if it was a categorical attribute and if by the mean in case of a numerical attribute.

Table 1. Brief description of the attributes in the Australian credit dataset.

Attribute	Description	Type
1	X1	Categorical
2	X2	Numerical
3	X3	Numerical
4	X4	Categorical
5	X5	Categorical
6	X6	Categorical
7	X7	Numerical
8	X8	Categorical
9	X9	Categorical
10	X10	Numerical
11	X11	Categorical
12	X12	Categorical
13	X13	Numerical
14	X14	Numerical

Table 2. Brief description of the attributes in the German credit dataset.

Attribute	Description	Type
1	Status of existing checking account	Categorical
2	Duration in month	Numerical
3	Credit history	Categorical
4	Purpose	Categorical
5	Credit account	Numerical
6	Savings account/bonds	Categorical
7	Present employment since	Categorical
8	Installment rate in percentage of disposable income	Numerical
9	Personal status and sex	Categorical
10	Other debtors/guarantors	Categorical
11	Present residence since	Numerical
12	Property	Categorical
13	Age	Numerical
14	Other installment plans	Categorical
15	Housing	Categorical
16	Number of existing credits at this bank	Numerical
17	Job	Categorical
18	Number of people being liable to provide maintenance for	Numerical
19	Telephone (yes/no)	Categorical
20	Foreign worker	Categorical

3.1.2. German Credit Data

The German dataset was provided by Prof. Hofmann in Hamburg. The original data has a mix of 20 categorical and numerical attributes (see Table 2). It documents several financial and demographic information about the applicants. For algorithms that need numerical attributes, Strathclyde University produced a numeric version of this dataset which is also available at UCI where the categorical attributes were converted into numerical ones, increasing the dimension to 24 input numerical values. The data instances are labeled as classes 1 (good) and 2 (bad).

3.1.3. Taiwanese Credit Data

This dataset relates to a credit card issuer in Taiwan and the targets were credit card holders. It contains 23 attributes, where three are categorical attributes and all others continuous. For the target variable was employed a binary variable, default payment (Yes = 1, No = 0).The attributes description is shown in the Table 3.

Table 3. Brief description of the attributes in the Taiwanese credit dataset.

Attribute	Description	Type
1	Limit Balance	Numerical
2	Sex	Categorical
3	Education	Categorical
4	Marital status	Categorical
5	Age	Numerical
6	The repayment status in September, 2005	Numerical
7	The repayment status in August, 2005	Numerical
8	The repayment status in July, 2005	Numerical
9	The repayment status in June, 2005	Numerical
10	The repayment status in May, 2005	Numerical
11	The repayment status in April, 2005	Numerical
12	Amount of bill statement in September, 2005	Numerical
13	Amount of bill statement in August, 2005	Numerical
14	Amount of bill statement in July, 2005	Numerical
15	Amount of bill statement in June, 2005	Numerical
16	Amount of bill statement in May, 2005	Numerical
17	Amount of bill statement in April, 2005	Numerical
18	Amount paid in September, 2005	Numerical
19	Amount paid in August, 2005	Numerical
20	Amount paid in July, 2005	Numerical
21	Amount paid in June, 2005	Numerical
22	Amount paid in May, 2005	Numerical
23	Amount paid in April, 2005	Numerical

3.2. Deep Learning

Deep learning (DL) methods are representation learning methods with multiple levels of layers, obtained by constructing simple yet non-linear components that one by one transform the representation at one level, starting with the input data, toward a representation at a higher abstract level [LeCun et al. 2015]. The main idea of DL is to extract feature layer by layer and combine low-level features to form high-level features, which can find underlying expressions of data and classify the data into different categories.

3.2.1. Deep Neural Networks (DNN)

As shown by Figure 1, a DNN consists of interconnected layers of neurons. Basically, these layers are input, hidden, and output. Neurons that connect to the input data compose the input layer, which identifies the most fundamental element of the data, and passes it to the hidden layers. The hidden layers can further analyse, draw data representations, and send their results to the next layer, and so on until the output layer, which classifies the data into established classes. For some authors DNN means that the hidden layers of the neural network are from two [Zhang 2014], and for other authors must contain a relatively large number of hidden layers (more than five) [Liao 2017].

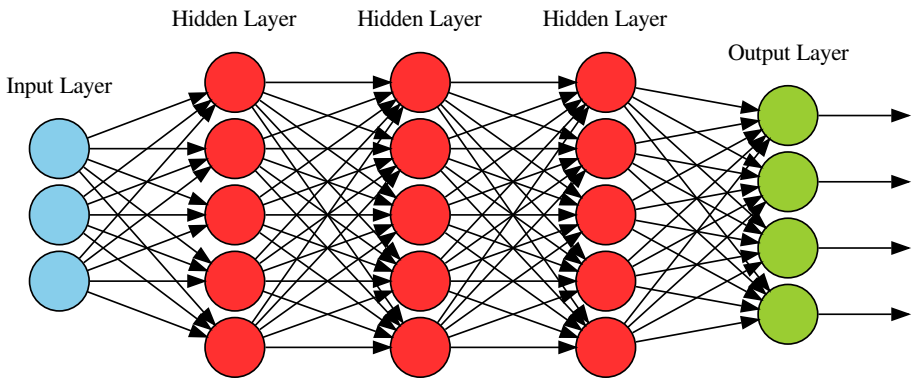

Figure 1. Example of an DNN with three hidden layers

Despite the architecture, there are also other parameters which play a key role for the DNN performance. Following we briefly discuss some of them:

- The **Batch size** determines how many training examples are presented in the model before the weight and bias are updated. The larger the batch size, the more system memory space is required [Canziani et al. 2016]. Otherwise, it also must be large enough to provide a satisfactory representation of the training set in each iteration.
- The **Number of Epochs** is the number of times that the entire dataset is passed through the neural network during training. Updating weights and other parameters with a small number of epochs would lead to underfitting. As the number of

epochs increases, the weight and other parameters are updated in the neural network, so training accuracy and validation accuracy will increase. However, when the number of epochs reaches a certain point, the accuracy of the validation begins to decrease while the accuracy of the training is still increasing, what means the model is overfitting. Thus, the ideal number of epochs is the point at which the validation accuracy reaches its highest value [Sinha et al. 2010].

- The **Learning Rate** defines how quickly a network updates its parameters [Zhang 2016]. Instead of using a constant learning rate to update the parameters (e.g., networks weights) for each training season, an adaptive learning rate called Adadelta Optimizer was used, which allows to specify different learning rates. Instead of accumulating the sum of square gradients over time, Adadelta restricts itself to a window of previous gradients that accumulate to some fixed size. This ensures that learning continues to progress even after many update iterations have been made.
- The **Activation Function** is used to introduce nonlinearity into DNN. It is the nonlinear transformation performed on the input data and the transformed output which will be send to the next layer as input data [Lin et al. 2013]..
- The **Network Initialization Mode** determines how to set initial random weights for DNN layers [Lin et al. 2013].

3.2.2. Convolutional Neural Networks (CNN)

CNN is one of the most used deep learning models. Inspired by the visual perception natural mechanism of living beings, CNN are usually designed to process data with grid topology, for example, a color image composed of three two-dimensional vectors that contain pixel intensities in the three color channels. However, it has also been succesfully applied to many other domain besides computer vision tasks. Figure 2 shows a classical example of CNN, in which the neurons are arranged in three dimensions: width, height and depth. For example, an image that has dimensions $32\times32\times3$ (the 3 refers to RGB values) neurons in a layer will connect to a custom region of the previous layer. In addition, the final output layer has dimensions $1\times1\times d$, where d is the number of classes. The dimensions are reduced from 3072 to d, with the output being a single vector of class distributions [Zhang 2016].

A CNN usually employs four main types of layers: convolutional, nonlinear, pooling and fully connected layers. Each layer transforms the input into the output through a different function. In a simple CNN, the input is passed through a number of convolutional layers, nonlinear layers, pooling layers, and fully connected layers. The convolution layer extracts specific patterns from the data by creating a feature map. Having multiple convolution layers allows the network to learn high-level filters with relatively fewer parameters [Doshi 2018]. All units in a resource map share the same filter bank. Different feature maps in a convolutional layer use different filter banks [Cao 2018]. The feature map produced by the convolution layer is passed through a nonlinear activation function. Pooling layers are intended to reduce the spatial dimension of the representation and control overfitting by reducing the number of parameters. Fully connected layers are commonly used as lower layers of the network to better summarize the information transmitted by the previous level layers in the final mapping of the classes [Cao 2018].

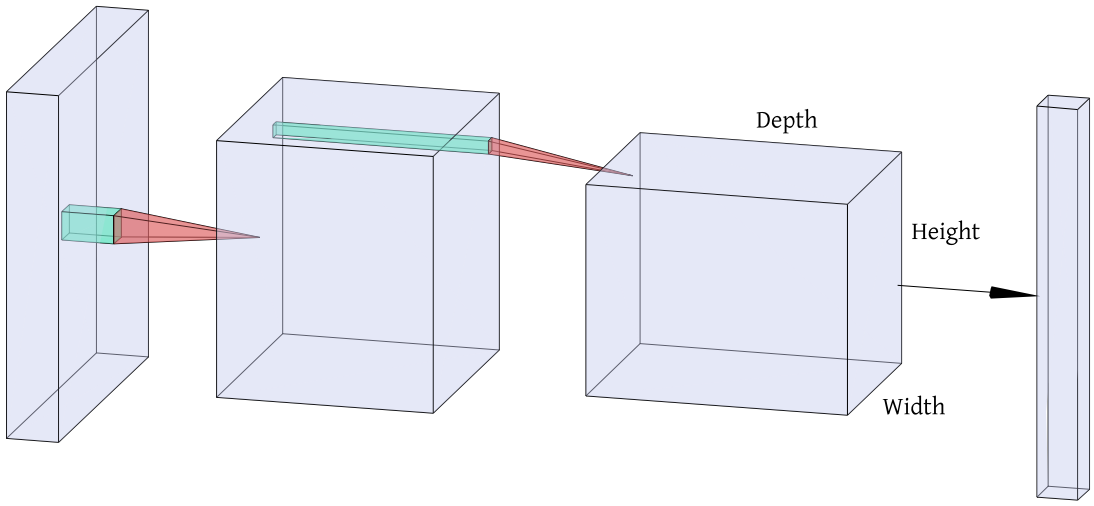

Figure 2. A CNN in three dimensions.

There are four key ideas behind CNN that leverage the properties of natural signals: local connections, shared weights, pooling, and the use of multi-layers [Cao 2018].

4. Experiments

In the following, we compare the deep learning models presented before against five very well-known classification algorithms on the three data sets studied in this paper. For sake of clarity, this section is divided in two parts: experimental setup and results.

4.1. Experimental Setup

In the following, we describe the experimental setup in which our experiments were conducted, as well as the parameter selection step for the classifiers. Table 4 summarize the three credit risk analysis data sets under study which were discussed previously in the paper.

Table 4. Brief description of the credit risk analysis data sets in terms of the number of data items (*#Inst.*), number of attributes (*#Attr.*) and number of classes (*#Classes*).

Name	#Inst.	#Attr.	#Classes [Distribution]
Australian credit	690	14	2 [307/383]
German credit	1000	20	2 [700/300]
Taiwanese credit	30000	23	2 [6636/23364]

Each experiment is conducted through of a 10-fold stratified cross-validation process averaged over three runs, taking the folds randomly each time. Regarding the parameter selection, a grid search method is executed over each training partition by doing a 3-fold stratified cross-validation. Such a nested cross-validation procedure ensures an unbiased learning as the test data are outside of the learning process. Therefore, the inner

cross-validation is used to parameter selection, while the outer cross-validation to evaluate the predictive performance.

The parameter selection is a key point in the training process of the models. Grid search is adopted here in order to find the better parameter combination based on a predefined parameter search space to be considered. Based on [Carneiro and Gabriel 2018], such a search space is related to the following parameters for each technique:

- DT has two parameters, the minimum number of samples required to split an internal node $m_{split} \in 2, 5, 10, (1 * n)/100$ and the minimum number of samples required to be at a leaf node $m_{leaf} \in 2, 5, 10, (1 * n)/100$, with n denoting the number of training data items;
- RF has one parameter, the number of trees in the forest $t \in 2^1, 2^2, \ldots, 2^{10}$;
- LR has two parameters, the norm used in the penalization $p \in l1, l2$ and the regularization strength $C \in 2^2, 2^4, \ldots, 2^{14}$;
- MLP has two parameters, the initial learning rate $\alpha \in 0.01, 0.05, 0.1, 0.2, 0.3$ and the number of neurons in the hidden layer $n_h \in 10, 20, 50, 100, 500, 1000$.
- SVM has two parameters, the kernel coefficient $\gamma \in 2^4, 2^3, \ldots, 2^{-10}$ and the penalty parameter $C \in 2^{12}, 2^{11}, \ldots, 2^{-2}$.

About the DNN parameters: it has three hidden layers, totaling five fully connected layers; the hidden layers has ten neurons; the batch size is set to twenty; the optimizer is Adadelta; the initial mode selected is the Gaussian function; the activation function is ReLU (except by the output layer, which is the Sigmoid function); and the number of epochs equals is fixed as 500.

About the CNN parameters: it consists of seven layers, four convolutional and three fully connected; and it uses ReLU as activation layer (except for the last Softmax). It is to mention that Convolutional gradients may easily explode or vanish after few training iterations. To keep them stable and training smooth, we added a pooling layer after each two convolutional layers. Notice that convolutional layers have their size in funcion of the number of attributes T in the dataset. The first two convolutional layers have size T, the next two have size T/2, and the fully connected layers have 300 neurons. Other parameters are defined as follows: learning rate $\alpha = 0.0005$; epochs equal to 15; optimizer as Adam; and batch size equal to 8. [Simonyan and Zisserman 2015] employed CNN with two layers followed by filters 3×3 and claimed that it is better than one with 5×5 or 7×7. Following this recommendation, our convolutional layers have a 3×3 filter. The kernel size is optimized over the set $\{5, 10, 20, 50\}$ (same value for each layer).

4.2. Results

Table 5 presents the averaged accuracy of the seven techniques over the three datasets. As pointed out in the literature, RF achieved consistent performance for the three data sets, returning the best results for two of three data sets. LR, which is another well-recommended algorithm for credit assessment, also presented good performance. By the contrary, DT and SVM were clearly outperformed by RF and LR, with SVM also presenting the highest time complexity among all techniques. MLP demonstrated inconsistent performance as it achieved the best result for the German data set, but the worse results for Australian and Taiwanese data sets.

Table 5. Classifiers results in terms of averaged accuracy (AA) and standard deviation (SD) for the three datasets under study.

	Australian credit		German credit		Taiwanese credit	
Algs.	AA	SD+/-	AA	SD+/-	AA	SD+/-
DT	84.48	3.60	72.03	3.92	77.46	0.62
LR	86.47	4.34	76.66	4.23	81.03	0.42
MLP	71.88	7.48	**78.03**	4.63	69.95	1.43
RF	**87.16**	4.52	76.83	3.66	81.88	0.43
SVM	72.03	3.92	76.46	3.51	77.88	0.24
DNN	85.10	2.40	75.90	4.30	78.50	0.60
CNN	85.78	3.50	75.04	3.52	**81.89**	0.54

Regarding the predictive performance of the deep learning models, Table 5 shows that DNN was outperformed by RF and LR, but CNN, although clearly outperformed by both RF and LR in the first two data sets, was able to achieve the best predictive result for the Taiwane data set (tied with RF). We believe this happened because this data set has a larger number of instances (30000) than others (690 Australian and 1000 German) and also because of the representational ability of the convolutional layers, which is considered a powerful feature extraction method. Finally, it is important to say that there are many space for architecture and parameter adjustment in both deep learning models evaluated, which can further increase their predictive performance.

5. Conclusion

This paper conducted an investigation about the usage of deep learning models for credit risk evaluation. To be specific we evaluated deep neural networks and convolutional neural networks against decision tree, multi-layer perceptron, logistic regression, support vector machine and random forest. Basically, all these algorithms were adopted to build predictive models, to classify the application of loan as good or bad using three datasets in credit domain from the UCI Machine Learning repository. In addition, grid-search was employed for rigorously tune the traditional techniques. K-Fold stratified cross-validation and nested-CV techniques were used to ensure an unbiased estimator and a very fair comparative analysis among them. After analyzing the results, we found out that the recommended general algorithm for risk credit classification is the random forest because its higher accuracy and simplicity (in comparison with CNN, for example). Future works will investigate more architectures and parameter adjustment for deep learning techniques.

References

Asuncion, A. and Newman, D. (2007). UCI machine learning repository.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., and Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the operational research society*, 54(6):627–635.

Canziani, A., Paszke, A., and Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *CoRR*, abs/1605.07678.

Cao, Y. (2018). *Deep learning based RGB-D vision tasks*. PhD thesis.

Carneiro, M. G. and Gabriel, A. (2018). What's the next move? learning player strategies in zoom poker games. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE.

Chen, J., Chen, W., Huang, C., Huang, S., and Chen, A. (2016). Financial time-series data analysis using deep convolutional neural networks. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, pages 87–92.

Chen, M.-C. and Huang, S.-H. (2003). Credit scoring and rejected instances reassigning through evolutionary computation techniques. *Expert Systems with Applications*, 24(4):433–441.

Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. (2016). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3):653–664.

Doshi, C. (2018). *A deep learning approach to state estimation from videos*. PhD thesis.

Hand, D. J. and Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 160(3):523–541.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

Lessmann, S., Baesens, B., Seow, H.-V., and Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136.

Li, Y., Lin, X., Wang, X., Shen, F., and Gong, Z. (2017). Credit risk assessment algorithm using deep neural networks with clustering and merging. In *2017 13th International Conference on Computational Intelligence and Security (CIS)*, pages 173–176.

Liao, Z. (2017). *Methods for Understanding and Improving Deep Learning Classification Models*. PhD thesis.

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *CoRR*, abs/1312.4400.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Sinha, S., Singh, T., Singh, V., and Verma, A. (2010). Epoch determination for neural network by self-organized map (som). *Computational Geosciences*, 14(1):199–206.

Sun, T. and Vasarhelyi, M. A. (2018). Predicting credit card delinquencies: An application of deep neural networks. *Intelligent Systems in Accounting, Finance and Management*, 25(4):174–189.

Tan, P.-N. (2018). *Introduction to data mining*. Pearson Education India.

Wei, G., Yingjie, S., and Mu, Y. X. (2015). Commercial bank credit risk evaluation method based on decision tree algorithm. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2015 Seventh International Conference on*, pages 285–288. IEEE.

Yeh, I.-C. and Lien, C.-h. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480.

Zhang, B. (2014). *Deep learning with application to hashing*. PhD thesis.

Zhang, D. and Zhou, L. (2004). Discovering golden nuggets: data mining in financial application. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(4):513–522.

Zhang, J. (2016). *Deep learning for multi-label scene classification*. PhD thesis.