Engineering Applications of Artificial Intelligence 65 (2017) 465–470

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A deep learning approach for credit scoring using credit default swaps

Cuicui Luo *, Desheng Wu, Dexiang Wu

Stockholm Business School, Stockholm University, Stockholm, Sweden

A R T I C L E I N F O

Keywords:
Deep learning
CDS
Credit scoring
Machine learning

A B S T R A C T

After 2007–2008 crisis, it is clear that corporate credit scoring is becoming a key role in credit risk management. In this paper, we investigate the performances of credit scoring models applied to CDS data sets. The classification performance of deep learning algorithm such as deep belief networks with Restricted Boltzmann Machines are evaluated and compared with some popular credit scoring models such as logistic regression, multi-layer perceptron and support vector machine. The performance is assessed using the classification accuracy and the area under the receiver operating characteristic curve. It is found that DBN yields the best performance.

1. Introduction

During US subprime mortgage crisis and the European sovereign debt crisis, established financial institutions in the USA and Europe suffered catastrophic losses. The crisis has raised concerns regarding the use of credit default swaps (CDS). Accordingly, credit risk management is becoming an increasingly important factor and attracted significant attention from researchers and market participants. In order to effectively manage the credit risk exposures and optimize profits, it has become one major focus for financial institutions to develop an accurate credit scoring model. A range of different statistical and machine learning techniques have been developed to build credit rating models.

After commercial scorecard was introduced, many statistical methods have been used for credit risk assessment. Though with wide application, these models have difficulty in modeling complex financial systems due to the use of fixed functions and statistical assumptions. Related studies have shown that machine learning techniques are superior to that of statistical techniques in dealing with credit scoring problems (Saberi et al., 2013). Logistic regression is one of the most frequently used. Statistical model used in credit scoring. Meanwhile, some shallow architectures such as support vector machines (SVMs) and multi-layer perceptron (MLPs) with a single hidden layer, have been widely applied to credit scoring.

Shallow architectures have been shown effective in solving many simple or well-constrained problems. However, these methods mainly focus on the outputs of classifiers at the abstract level, while neglecting the rich information hidden in the confidence degree (Hinton and Salakhutdinov, 2006). Their limited modeling and representational power can cause difficulties when dealing with more complicated real-world applications. To tackle these drawbacks, Hinton and Salakhutdinov (2006) first successfully introduced training algorithms for deep architectures. The deep belief networks (DBN) with sufficient hidden layers are developed as a powerful ensemble technique to capture the rich information in the confidence degree. Since then, deep networks have been applied with success in classification task, e.g., computer vision (Krizhevsky et al., 2012), health state classification (Tamilselvan and Wang, 2013; Abdel-Zaher and Eldeib, 2016), speech and language processing (Mohamed et al., 2012; Ling et al., 2013) and emotion recognition (Le and Provost, 2013). Recently, the deep belief networks (DBN) has alo been applied in financial predcition (Ribeiro and Lopes, 2011) Whether such deep architectures have theoretical advantages compared to shallow architectures in credit risk assessment remains an open question. To the best of our knowledge, there were few studies on credit risk assessment by using the DBN.

Many empirical studies have investigated the performance of these credit scoring models in credit risk assessment. Bellotti and Crook (2009) test support vector machines against several other well-known algorithms on a large credit card database and find that SVMs are successful in comparison to established approaches to classifying credit card customers who default. Li et al. (2006) find SVMs outperform MLP in credit scoring by applying consumer credit data. It is also found that SVMs perform slightly better than LR by applying SVM to a database of applicants for building and loan credit (Schebesch and Stecking, 2005). According to Bellotti et al. (2011), support vector machines can produce notably better predictions of international bank ratings than the standard method. Han et al. (2013) find orthogonal support vector machine achieves better performance in achieves better performance. However,

* Corresponding author.

https://doi.org/10.1016/j.engappai.2016.12.002
Received 19 September 2016; Received in revised form 20 October 2016; Accepted 1 December 2016
Available online 13 December 2016
0952-1976/

another study conducted by Van Gestel et al. (2006) finds no significant difference between SVM, LR in terms of proportion of test cases correctly classified. The performance of four learning algorithms for corporate credit ratings are compared over a data set consisting of real financial data (Zhong et al., 2014). Lessmann et al. (2015) compare 41 classifiers in terms of six performance measures across eight real-world credit scoring data sets. They suggest that several classifiers predict credit risk significantly more accurately than the industry standard LR across eight real-word credit scoring data sets. The credit scoring accuracy of five neural network models for both the German and Australian credit data sets is investigated by West (2000). He finds that LG is the most accurate of the traditional methods and also suggests that neural network credit scoring models can achieve fractional improvements in credit scoring accuracy. Bhattacharyya et al. (2011) evaluate the performance of two advanced data mining techniques, random forests, support vector machines and logistic regression, for credit card fraud detection. Despite a bewildering array of models, relatively little research compares the performance of these models with DBN in terms of their classification accuracy in credit scoring. Meanwhile, the above studies only consider two-class classification problems.

To our knowledge, this is the first comprehensive study of DBN model in corporate credit rating based on CDS data. Therefore, this paper fills in such a literature gap by introducing DBN as the algorithm for credit rating to generate fast and accurate individual classification results. The goal of the paper is to provide a set of descriptive results and tests that lay a foundation for future theoretical and empirical work on DBN in credit scoring in CDS markets. In this paper, we investigate the performances of different credit scoring models by conducting experiments on a collection of CDS data. The data set contains 661 companies with eleven input attributes and three classification categories. The 6-month, 1-year, 2-year, 3-year, 4-year, 5-year, 7-year and 10-year spreads, recovery rate, sector and region will be used as input variables for the learning algorithms. The output variable contains three rating categories: A, B and C. In our experiments, we compared the results of MLR, MLP, and SVM with the Deep Belief Networks (DBN) with the Restricted Boltzmann Machine by applying 10-fold cross-validation. Our findings demonstrate that the deep learning algorithm significantly outperforms the baselines. Our paper contributes to this literature by investigating the performance of DBN in corporate credit scoring.

The remainder of the paper is organized as follows. Section 2 describes the credit scoring models examined in the paper. Section 3 describes CDS data set. Section 4 presents the empirical results from comparing the models. Section 5 summarizes the paper and makes concluding remarks.

2. Models

In this section, we present four popular machine learning algorithm used for credit scoring.

2.1. Multinomial logistic regression

Logistic regression is one of the most frequently used statistical model in credit scoring. The basic setup of Multinomial Logistic Regression (MLR) is the same as in logistic regression and the only difference is that the dependent variables are categorical rather than binary.

Suppose we have a set of training data with n observations $\{x_i, y_i\}$, where $x_i \in \mathbf{R}^m$ and $y_i \in \{1, \dots, K\}$ for $i = 1, \dots, n$. The probability that the i-th observation belongs to the j-th category with the exception of the last class is

$$\pi_{ij} = P(y_i = j) = \frac{e^{\beta_j \cdot x_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k \cdot x_i}} \tag{1}$$

Because the sum of all the probabilities equals one, the last class has probability

$$\pi_{iK} = 1 - \sum_{j=1}^{K-1} \pi_{ij} = \frac{1}{1 + \sum_{k=1}^{K-1} e^{\beta_k \cdot x_i}} \tag{2}$$

where β_j is the set of regression coefficients associated with outcome j and β is an m*(K-1) matrix.

The negative multinomial log-likelihood is:

$$L = -\sum_{i=1}^{n} \left(\sum_{j=1}^{K-1} Y_{ij} \cdot ln\pi_{ij} + (1 - \sum_{j=1}^{K-1} Y_{ij}) \cdot ln \left(1 - \sum_{j=1}^{K-1} \pi_{ij} \right) \right) \tag{3}$$

where Y_{ij} denotes the output variable that the i-th observation belongs to the j-th category.

The above log-likelihood function is solved by a Quasi-Newton maximization Method which is described in details by Weka (Hall et al., 2009). A ridge estimator is used in order to prevent over-fitting by penalizing large coefficients (Le Cessie and Van Houwelingen, 1992).

2.2. Support vector machine

Support Vector Machine (SVM) was first introduced in 1992 by (Boser et al., 1992). It is a classification and regression tool that applies machine learning technique to maximize predictive accuracy while automatically avoiding over-fit to the data. It can learn both simple and highly complex classification models and employs sophisticated mathematical principles to avoid over-fitting. We describe SVMs for two-class classification here and it can be extended to K class classification easily by constructing K two-class classifiers (Boser et al., 1992).

Given a set of training data $\{v_i, y_i\}$, where $v_i \in \mathbf{R}^m$, and $y_i \in \{-1, 1\}$ for $i = 1, \dots, n$. The Convex quadratic programming problem (QP) for SVM classification is

$$Maximize \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(v_i, v_j) \tag{4}$$

with constraints

$$0 \leq \alpha_i \leq C \tag{5}$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \tag{6}$$

where C is an upper bound and α_i is a Lagrange multiplier for each sample i. $K(v_i, v_j)$ denotes the value of the SVM kernel function for i-th and j-th inputs.

In this paper, we consider Gaussian radial basis function (RKF) with parameter σ, which is defined by

$$K(\mathbf{v}_i, \mathbf{v}_j) = e^{\frac{-\|\mathbf{v}_i - \mathbf{v}_j\|^2}{2\sigma^2}} \tag{7}$$

For any given input $\mathbf{v}$, the output prediction of a non-linear SVM is explicitly:

$$u = sign \left(\sum_{i=1}^{n} \alpha_i y_i K(v_i, \mathbf{v}) + b \right) \tag{8}$$

where bias b and vector of α_i are the variables determined by the above QP optimization problem.

The Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient conditions for an optimal solution of a positive definite QP problem. The KKT conditions for the QP problem (1) are

$$\alpha_i = 0 \Longrightarrow y_i u_i \geq 1 \tag{9}$$

$$\alpha_i = C \Longrightarrow y_i u_i \leq 1 \tag{10}$$

$$0 < \alpha_i < C \Longrightarrow y_i u_i = 1 \tag{11}$$

The Sequential Minimal Optimization (SMO) (Platt et al., 1998) gives an efficient way of solving the above support vector machine optimization problem. It decomposes the QP problem into QP subproblems and solves the smallest possible optimization problem by using two Lagrange multipliers at each step. The SMO algorithm iterates until the entire training set obeys the KKT conditions.

In this paper, we consider a three-class classification problem and the problem is solved by using pairwise classification by one-versus-one reduction. The details of this approach is described in Hastie and Tibshirani (1998).

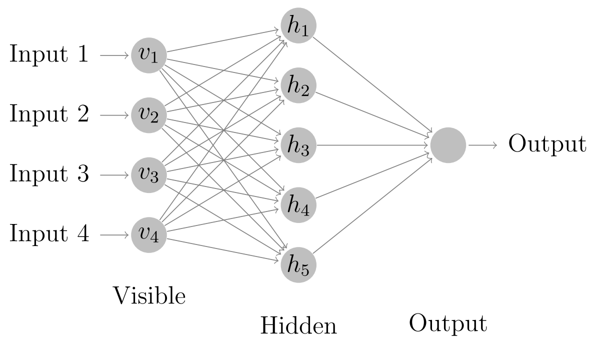

Fig. 1. Multi-layer feed-forward perceptron with bias nodes.

2.3. Multilayer perceptron

The Multi-layer Perceptron (MLP) is a simple feed-forward neural network with an input layer, several hidden layers and one output layer. MLP network is widely used for pattern classification, recognition and prediction. The basic structure of a MLP with one hidden layer and one output is illustrated in Fig. 1.

For the purpose of credit scoring, it is sufficient to consider a MLP with one hidden layer and one or two outputs. Let's consider a MLP consisting of two layers and assume that there are n neurons **v** in visible layer and m neurons **h** in hidden layer. In each layer, an activation function f(x) is used to determine the neurons output value. These neurons in hidden layer receive signals from the neurons in the visible layer.

In this paper, a traditional sigmoid function is used in the hidden layer:

$$f(x) = \frac{1}{1+e^{-x}} \tag{12}$$

For example, neuron h_j receives a signal from v_i with a weight factor w^1_{ij}. for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

The net input into neuron h_j is given by

$$n_j = \sum_{i=1}^{n} v_i w_{ij} + b_j, \quad j = 1, 2, \ldots, m. \tag{13}$$

where b_j is a bias value.

Thus the activation of neuron n_j is

$$a_j = f(n_j) = f\left(\sum_{i=1}^{n} v_i w_{ij} + b_j\right) \tag{14}$$

The output value, Y, for output neuron, k, can be expressed as a function of the input values and network weights, w, as follows:

$$Y_k = \sum_{j=1}^{m} f\left(\sum_{i=1}^{n} v_i w_{ij} + b_j\right) w_{jk} + b_k \tag{15}$$

w_{ij} and w_{jk} are the weights of hidden and output layer respectively.

The parameters for the network are determined iteratively, commonly via the back propagation learning algorithm.

2.4. Deep belief networks

Deep Belief Network (DBN) by Hinton et al. (2006) is a multi-layer generative graphical model, which is obtained by training and tacking several layers of Restricted Boltzmann Machines (RBM) in a greedy manner (Salakhutdinov et al., 2007). We will introduce RBM and then the structure of DBN in the following two subsections.

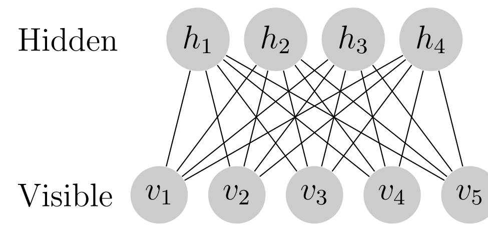

Fig. 2. Restricted Boltzmann Machine with 5 visible units and 4 hidden units.

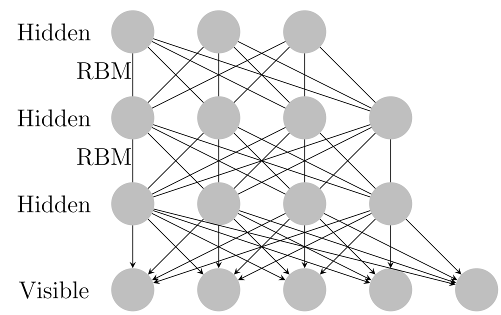

Fig. 3. DBN.

2.4.1. Restricted Boltzmann machines

An RBM is a bipartite undirected graph that contains two variably-sized layers. The key assumption that RBMs make is that the hidden units are conditionally independent given the visible units, and vice versa. The inputs in visible layer are connected to the neutrons in hidden Layer and there is no visible to visible or hidden to hidden neutron connection. Fig. 2 depicts the connective structure of a RBM.

Consider a RBM with n visible unites **v** and m hidden units **h**, the energy function is defined as:

$$E(v,h) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} v_i h_j \tag{16}$$

where a_i,b_j are biases for binary visible unit v_i and hidden unit h_j, respectively. w_{ij} is the weight between visible unit i and hidden unit j. a_i and b_j are the binary states of the visible and hidden units.

The probability that the network has visible vector v and hidden vector h for an RBM is defined by the Boltzmann distribution

$$P(v,h) = \frac{1}{Z} e^{-E(v,h)} \tag{17}$$

where Z is the partition function that makes sure the distribution sums to 1. From the definition of P(v,h), we can see that the probability of a configuration increases as its energy decreases. Training an RBM in an unsupervised manner involves manipulating the energy function so that it would assign low energy (and therefore high probability) to values of **v** that are similar to the training data, and high energy to values that are far from the training data. Learning in a RBM can be achieved very effectively by the Contrastive Divergence algorithm (Hinton et al., 2006).

2.4.2. DBN

A DBN is multi-layer belief networks which is created as a stack of RBMs. Each layer is Restricted Boltzmann Machine and they are stacked each other to construct DBN. As shown in Fig. 3, the hidden layer of the first RBM is treated as a visible layer for the second RBM. This second RBM will learn the feature distribution of the hidden layer of the first RBM. The input layer of the first RBM is the input layer for the whole network. As layers are stacked the network learns increasing complex combinations of features from the original data.

Table 1
Statistics of numerical attributes.

Attributes	Description	Min	Max	Missing
v_1	recovery rate	0.0274	0.4900	0
v_2	6-month spread	0.0002	6.9648	16
v_3	1-year spread	0.0003	4.9937	35
v_4	2-year spread	0.0004	3.6558	0
v_5	3-year spread	0.0005	3.1216	2
v_6	4-year spread	0.0007	2.8275	0
v_7	5-year spread	0.0011	2.6605	6
v_8	7-year spread	0.0017	2.4277	6
v_9	10-year spread	0.0020	2.1939	0

Table 2
Number of companies in each credit category.

Rating	Number
A	275
B	374
C	12
Total	661

3. Data

In this section, we review briefly the basic features of a typical CDS contract. We then discuss the CDS data used in our experiment.

3.1. CDS

A credit default swaps (CDS) contract is a derivative that protects the buyer against credit events by a particular company or sovereign entity, such as bankruptcy, failure to pay and restructuring. CDS is the most popular credit derivative. The CDS were created in the 1994 and have drastically increased since the early 2000s. By the end of 2007, the outstanding CDS amount was $62.2 trillion. Although it decreased after the financial crisis of 2007–2008, it remained considerable at about $12 trillion at end-December 2015.

The CDS contracts are traded in the over-the-counter market between large financial institutions. The buyer of the contract makes periodic payments to the seller and the seller of protection pays compensation to buyer if a credit event occurs and the contract is terminated. The CDS spread is the premium paid per year by protection buyer to seller.

3.2. Data

The XR 14 (no restructuring) CDS contracts on 2016-03-16 were obtained from Markit. The data set consists of 661 publicly-traded firms from eight regions including North American, Aisa, Europe etc. in ten different sectors (Industrials, Consumer Services, Technology, Utilities, Telecommunications Services, Healthcare, Financials, Energy, Basic Materials, Consumer Goods). We divide these 661 firms into three aggregated rating categories: A, B, C. $A = \{A, AA, AAA\}$ and $B = \{B, BB, BBB\}$. C only contains all firms rated CCC in the composite curve rating provided by Markit. Ratings B and C are referred to as "below investment grade".

Each firm in the dataset contains 11 explanatory attributes. Among them, there are 9 numerical attributes including recovery rate, spreads(6-month, 1-year, 2-year, ⋯, 5-year, 7-year, 10-year) and 2 categorical attributes including sector and region mentioned above. Tables 1, 2 show the basic statistics of these numerical attributes.

3.2.1. Data pre-processing

First, all categorical attributes are converted into binary numeric attributes. By using the one-attribute-*per*-value approach, an attribute with k values is transformed into k binary attributes if the class is nominal (see (Breiman et al., 1984)). Then all the numerical attributes are normalized.

Table 3
Classification accuracy comparisons.

	AUC	False Negative	False positive	Accuracy
MLR	0.849	12.56%	10.14%	77.31%
SVM	0.863	8.02%	4.54%	87.4%
MLP	0.922	6.96%	5.30%	87.75%
DBN	1	0	0	100%

4. Result

In this paper, the four algorithms compared are Multinomial Logistic Regression (MLR), the Multilayer Perceptron (MLP), Support Vector Machine (SVM) and Deep Belief Network (DBN) with Restricted Boltzmann Machines. We have used weka (Waikato environment for knowledge analysis) tool to calculate efficiency based on accuracy regarding correct instances generated with confusion matrix for SVM, MLP and MLR and statistical software R for DBN.

To minimize the impact of data dependency and improve the reliability of the estimates, 10-fold cross validation is used to create random partitions of the data sets. In 10-fold cross-validation, the training set is divided into 10 subsets of equal size. Then each of these 10 subsets is tested using the classifier trained on the remaining 9 subsets for the credit scoring models. An advantage of cross validation is that the credit scoring model is developed with a large proportion of the available data and that all of the data is used to train the credit scoring models can also be used for supervised classification.

4.1. Empirical result

Table 3 summarizes the results of our experiments. For each model, the accuracy, AUC, FN and FP are displayed. These results are averages of errors determined for each of the 10 independent data set partitions used in the cross-validation methodology.

Accuracy is the percentage of correctly classified instances and provides a measure for the ability to make accurate predictions on previously unseen cases. As shown in Table 3, the accuracy rate of DBN with Restricted Boltzmann Machines is highest. Following DBN is MLP with an accuracy rate of 87.75%, and SVC with 87.4%. The false positive rate is defined as the proportion of the firms that model rating is higher than actual rating. For example, rating B classified as A and rating C classified as B or A. The false negative rate (FN) is the proportion of the firms that model rating is lower than actual rating, i.e., rating B classified as C and rating A classified as B or C. MLR model has the largest false positive rate of 87.4% model and has the lowest accuracy, too. The area under the Receiver Operating Characteristic curve (AUC) also illustrates the performance of a classification model. The larger of the area, the better. MLP algorithm performs slightly better than SVM in terms of AUC, accuracy and false negative rate. Based on all the accuracy measures in Table 3, we conclude that DBN yields significantly better results than other algorithms while MLR has the worst performance.

Comparing in more detail SVM model and MLP model in Table 3, the false positive rate of MLP is larger than SVM. That means, the percentage of modeling rating higher than actual rating is larger by using MLP model. If investors invest in firms with model rating A (which is rating B actually), it may cause potential loss when there is a credit event. Meanwhile, the false negative rate of SVM is larger than that of MLP, which can make investors lose some good investment opportunities based on SVM rating model.

Besides good accuracy, it is important to know how the models perform in each of the aggregated rating categories. The performances of each model are reported in Tables 4–7 respectively. Obviously, the classification performance of DBN is the best and MLR has the worst performance in each of the aggregated rating categories. From Tables 4–6, it is observed that the MLR, SVM and MLP approaches don't perform well in the outer aggregated category C. SVM fails to classify category C and has an error rate of 100%. Except DBN, SVM performs best when

Table 4
The confusion matrix of MLR.

Actual	Classified			
	A	B	C	
A	213	58	4	
B	58	295	21	
C	1	8	3	
Actual	Classfied			Total
	A	B	C	
A	0.775	0.211	0.015	1
B	0.155	0.789	0.056	1
C	0.083	0.667	0.250	1

Table 5
The confusion matrix of MLP.

Actual	Classified			
	A	B	C	
A	238	37	0	
B	25	340	9	
C	0	10	2	
Actual	Classfied			Total
	A	B	C	
A	0.865	0.135	0	1
B	0.067	0.909	0.024	1
C	0	0.833	0.167	1

Table 6
The confusion matrix of SVM.

Actual	Classified			
	A	B	C	
A	222	53	0	
B	18	356	0	
C	0	12	0	
Actual	Classfied			Total
	A	B	C	
A	0.807	0.193	0	1
B	0.048	0.952	0	1
C	0	1	0	1

Table 7
The confusion matrix of DBN.

Actual	Classified			
	A	B	C	
A	275	0	0	
B	0	374	0	
C	0	0	12	
Actual	Classfied			Total
	A	B	C	1
G1				
A	1	0	0	1
B	0	1	0	1
C	0	0	1	1

classifying category B and MLP performs better than MLR and SVM in rating A classification. Table 5 shows that 25 of 661 samples are wrongly classified as rating A by MLP and 18 samples are incorrectly classified as rating A by SVM. Because rating A is referred to as "investment grade", investors who invest in these firms that are incorrectly classified may suffer more credit loss. We can see that SVM does better than MLP in this respect.

5. Conclusion

In the aftermath of 2007–2008 crisis, one of the worst financial crises in modern history, it is clear that corporate credit scoring is becoming a key role in credit risk management. This paper provides a new perspectives on the credit scoring problem using deep learning algorithms. We applied Deep Belief Network (DBN) with Restricted Boltzmann Machines and compared the classification performance with Multinomial Logistic Regression (MLR), Multilayer Perceptron (MLP) and Support Vector Machine (SVM) to credit scoring problem. A collection of CDS data is used for comparison of the performance in corporate credit rating. The test results clearly indicate DBN with Restricted Boltzmann Machines outperforms other algorithms. A key finding is that the good performance of DBN model is consistent on all considered rating categories.

We believe that the flexibility of deep learning models can provide strong support in credit scoring dealing with rich and complicated information in credit risk assessment. As a direction for further research, we will embed DBN within a dynamic model which allows for prediction at all weekly time horizons. The out-of-sample forecasting performance can be investigated.

Acknowledgment

This work is supported by Marianne and Marcus Wallenberg Foundation (Grant Number: MMW 2015.0007).

References

Abdel-Zaher, A.M., Eldeib, A.M., 2016. Breast cancer classification using deep belief networks. Expert Syst. Appl. 46, 139–144.
Bellotti, T., Crook, J., 2009. Support vector machines for credit scoring and discovery of significant features. Expert Syst. Appl. 36 (2), 3302–3308.
Bellotti, T., Matousek, R., Stewart, C., 2011. A note comparing support vector machines and ordered choice models predictions of international banks ratings. Decis. Support Syst. 51 (3), 682–687.
Bhattacharyya, S., Jha, S., Tharakunnel, K., Westland, J.C., 2011. Data mining for credit card fraud: a comparative study. Decis. Support Syst. 50 (3), 602–613.
Boser, B.E., Guyon, I.M., Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory. ACM, pp. 144–152.
Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A., 1984. Classification and Regression Trees. CRC Press.
Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The weka data mining software an update. ACM SIGKDD Explor. Newsl. 11 (1), 10–18.
Han, L., Han, L., Zhao, H., 2013. Orthogonal support vector machine for credit scoring. Eng. Appl. Artif. Intell. 26 (2), 848–862.
Hastie, T., Tibshirani, R., et al., 1998. Classification by pairwise coupling. Ann. Stat. 26 (2), 451–471.
Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18 (7), 1527–1554.
Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105.
Le, D., Provost, E.M., 2013. Emotion recognition from spontaneous speech using hidden markov models with deep belief networks. In: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, pp. 216–221.
Le Cessie, S., Van Houwelingen, J.C., 1992. Ridge estimators in logistic regression. Appl. Stat. 191–201.
Lessmann, S., Baesens, B., Seow, H.-V., Thomas, L.C., 2015. Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. Eur. J. Oper. Res. 247 (1), 124–136.
Li, S.-T., Shiue, W., Huang, M.-H., 2006. The evaluation of consumer loans using support vector machines. Expert Syst. Appl. 30 (4), 772–782.
Ling, Z.-H., Deng, L., Yu, D., 2013. Modeling spectral envelopes using restricted boltzmann machines and deep belief networks for statistical parametric speech synthesis. IEEE Trans. Audio Speech Lang. Process. 21 (10), 2129–2139.
Mohamed, A., Dahl, G.E., Hinton, G., 2012. Acoustic modeling using deep belief networks. IEEE Trans. Audio Speech Lang. Process. 20 (1), 14–22.
Platt, J., et al., 1998. Sequential minimal optimization: A fast algorithm for training support vector machines.
Ribeiro, B., Lopes, N., 2011. Deep belief networks for financial prediction. In: International Conference on Neural Information Processing. Springer, pp. 766–773.
Saberi, M., Mirtalaie, M.S., Hussain, F.K., Azadeh, A., Hussain, O.K., Ashjari, B., 2013. A granular computing-based approach to credit scoring modeling. Neurocomputing 122, 100–115.

Salakhutdinov, R., Mnih, A., Hinton, G., 2007. Restricted boltzmann machines for collaborative filtering. In: Proceedings of the 24th international conference on Machine learning. ACM, pp. 791–798.
Schebesch, K.B., Stecking, R., 2005. Support vector machines for classifying and describing credit applicants: detecting typical and critical regions. J. Oper. Res. Soc. 56 (9), 1082–1088.
Tamilselvan, P., Wang, P., 2013. Failure diagnosis using deep belief learning based health state classification. Reliab. Eng. Syst. Saf. 115, 124–135.
Van Gestel, T., Baesens, B., Suykens, J.A., Van den Poel, D., Baestaens, D.-E., Willekens, M., 2006. Bayesian kernel based classification for financial distress detection. Eur. J. Oper. Res. 172 (3), 979–1003.
West, D., 2000. Neural network credit scoring models. Comput. Oper. Res. 27 (11), 1131–1152.
Zhong, H., Miao, C., Shen, Z., Feng, Y., 2014. Comparing the learning effectiveness of bp, elm, i-elm, and svm for corporate credit ratings. Neurocomputing 128, 285–295.