Deep Neural Networks with L1 and L2 Regularization for High Dimensional Corporate Credit Risk Prediction

Mei Yang[a] meiyang@cqu.edu.cn, **Ming K. Lim**[d] Ming.lim@glasgow.ac.uk, **Yingchi Qu**[a] quyingchi0411@163.com, **Xingzhi Li**[c] lixingzhi1991@hotmail.com, **Du Ni**[b,*] mt15nn@mail.wbs.ac.uk

[a] School of Economics and Business Administration, Chongqing University, Chongqing, 400030, PR China

[b] School of Management, Nanjing University of Posts and Telecommunications, Jiangsu, 210003, PR China

[c] School of economics and management, Chongqing Jiaotong University, Chongqing, 400074, PR China

[d] Adam Smith Business School, University of Glasgow, Glasgow G14 8QQ, the United Kingdom

Corresponding author

E-mail addresses: mt15nn@mail.wbs.ac.uk (Du Ni)

Abstract

Accurate credit risk prediction can help companies avoid bankruptcies and make adjustments ahead of time. There is a tendency in corporate credit risk prediction that more and more features are considered in the prediction system. However, this often brings redundant and irrelevant information which greatly impairs the performance of prediction algorithms. Therefore, this study proposes an HDNN algorithm that is an improved deep neural network (DNN) algorithm and can be used for high dimensional prediction of corporate credit risk. We firstly theoretically proved that there was no regularization effect when L1 regularization was added to the batch normalization layer of the DNN, which was a hidden rule in the industrial implementation but never been proved. In addition, we proved that adding L2 constraints on a single L1 regularization can solve the issue. Finally, this study analyzed a case study of credit data with supply chain and network data to show the superiority of the HDNN algorithm in the scenario of a high dimensional dataset.

Keywords

High dimensional data, Credit risk, Deep neural network, Prediction, L1 regularization

1. Introduction

Under the influence of Corona Virus Disease 2019, companies in the supply chain have been facing fierce pressure such as increased costs, shortage of inventory, and logistics disruption (Chowdhury et al., 2021; Moosavi et al., 2022; Singh et al., 2021). When a company is suffering from supply chain challenges, other related companies may also take a hit because of the bullwhip effect (Agca et al., 2021; Roukny et al., 2018). Besides, negative news and comments also deteriorate the credit of a company constantly (Bonsall IV et al., 2018; Kiesel, 2021). Therefore, researchers are trying to collect all the possible factors that affect the corporate credit risk (CCR) to avoid prediction inaccuracy. As a result, the dataset dimension will become unacceptably high for the ordinary CCR prediction models, which is prone to causing dimension disaster (Arias-Castro et al., 2018; Fernández-Martínez et al., 2020).

Generally, classical machine learning algorithms such as Support Vector Machine (SVM), Neural Network (NN), and Logistic regression (LR) have shown a good predictive ability for common credit risk datasets (Barboza et al., 2017; Chen et al., 2020b; Han et al., 2020). But for high dimensional datasets, researchers hardly resort to a proper solution in that although adding more related information about the same object can improve the prediction accuracy (Wu et al., 2022; Zhang et al., 2022), this usually introduces a large amount of redundant and irrelevant information, causing poor performance of machine learning algorithms (Ayesha et al., 2020; Danenas et al., 2015; Tan et al., 2014). Therefore, it is necessary to select truly relevant features through feature selection to improve the model accuracy. The most common approach is human selection operated by experienced supply chain experts, but it will lose objectivity. So, some researchers use regularization to diminish redundant parameters during training, thus promoting network sparsity (Emmert-Streib et al., 2019; Ghaddar et al., 2018; Salehi et al., 2019). In CCR prediction, deep neural network (DNN) algorithms have attracted much attention because they can mine the latent features of the data as much as possible (Bouwmans et al., 2019; Kim et al., 2017; Liu et al., 2017b). Besides, the DNN algorithm can automatically extract features to pursue objectivity that may suffer from manually selected features (Chen et al., 2020a; Suryanarayana et al., 2018). However, DNN algorithms rarely use regularization largely because L1 regularization cannot produce sparsity in the DNN algorithm (Van Laarhoven, 2017). However, there is a lack of theoretical proof of the phenomenon and a solution to the issue. Although Liu et al. (2017a) affirmed that the DNN algorithm has advantages in processing high dimensional data, they only combined it with the greedy algorithm to obtain the optimal solution. In other words, the advantages of L1 regularization for high dimensional data cannot be fully exploited in DNN algorithms.

To address the above problems, this study considers the influence of network information and supply chain information on CCR and proposes an HDNN algorithm that is an improved DNN algorithm for high dimensional CCR datasets. We firstly theoretically prove that there is no regularization effect when L1 regularization is combined with normalization, which means that L1 regularization will fail in the DNN algorithm. Besides, we propose to add L2 constraints on a single L1 regularization for high dimensional feature selection. This not only allows the DNN algorithm to perform feature selection through L1 regularization but also adds L2 norm to prevent overfitting.

The rest of this study is structured as follows. Section 2 reviews the related literature. Section 3 presents the data and methods of this study. Section 4 introduces the HDNN algorithm of this study and presents the algorithm results. Section 5 discusses the effectiveness of the HDNN algorithm compared to other algorithms. Finally, the conclusion and possible future research directions are presented in Section 6.

2. Literature Review

This section will review past CCR studies as well as high dimensional feature selection studies to demonstrate the research trends. The details are as follows.

2.1 Corporate credit risk prediction

CCR prediction shows the advantages of responding to the corporate credit crisis in advance, which has attracted much attention from researchers and practitioners (Basturk et al., 2021; Chen et al., 2021; Mansi et al., 2011). Earlier researchers mainly used financial data directly related to companies to predict credit risk (Chang et al., 2018; Trustorff et al., 2011). For example, Chang et al. (2018) used 16 indicators such as asset-liability ratio, net profit ratio, and solvency to help companies assess credit risk and improve loan business efficiency. However, these public financial data are often released on a quarterly or annual basis, and the timely credibility has been questioned (Cisi et al., 2020; Lev, 2018). To address this issue, researchers turn to online news and comments to capture crisis information quickly (Bonsall IV et al., 2018; Kiesel, 2021; Wei et al., 2019). Furthermore, many studies have shown that Corona Virus Disease 2019 caused great damage to the global supply chain, and companies in the supply chain are often mutually influenced (Chowdhury et al., 2021; Moosavi et al., 2022; Singh et al., 2021). Once a company has a credit crisis, other companies in the supply chain will deeply suffer too (Agca et al., 2021; Roukny et al., 2018). Therefore, adding more information about supply chain and news is important to the existing CCR prediction system.

Linear discriminant analysis and linear relationship analysis based on statistical methods are considered to be the most classical methods of traditional CCR prediction (Mylonakis et al., 2010; Psillaki et al., 2010; Ryu et al., 2005). However, these models are suitable for scenarios with few features and potentially linear relationships. If the features are increased, the poor performance of the prediction will occur (Albu et al., 2019; Hassani et al., 2020). Therefore, machine learning algorithms that are good at processing complex data structures, are used for CCR prediction more frequently (Bhatore et al., 2020; Lappas et al., 2021; Ma et al., 2019). In addition, compared to statistical methods based on assumptions about the data distribution, machine learning algorithms allow machines to automatically learn useful knowledge from massive amounts of data (Borlea et al., 2021; Chiang et al., 2014; Ni et al., 2020).

Commonly used machine learning algorithms in CCR prediction include neural NN, SVM, LR, and ensemble algorithms (Bhatore et al., 2020; Lappas & Yannacopoulos, 2021; Ma & Lv, 2019; Yang et al., 2022). Take NN as an example. The earliest literature using NN to predict corporate credit risk can trace back to 1988 when Dutta et al. (1988) used NN to predict CCR. The results showed that the prediction accuracy of NN was 18.6% higher than that of traditional linear methods. With the increase of features that affect CCR, SVM tends to attract researchers' attention due to its outstanding advantages in high dimensional prediction (Erfani et al., 2016; Upadhyay et al., 2020). Because SVM can map the dataset features to a high dimensional space and directly classify the training samples in the high dimensional space, the curse of dimensionality can be avoided cleverly (Erfani et al., 2016). Zhang et al. (2015b) used the SVM algorithm to calculate a corporate credit risk dataset with 31 indicators, proving that the model can accurately classify the credit status of SMEs. In addition to traditional machine learning algorithms such as NN and SVM, deep learning has also attracted much attention because it can mine the potential features of data as much as possible. It has been widely used in speech recognition, natural language processing, and image recognition (Bouwmans et al., 2019; Kim et al., 2017; Liu et al., 2017b).

In the field of credit risk prediction, researchers mainly use variants of deep learning such as Deep Belief Network (DBN), Long Short-Term Memory Network (LSTM), and DNN. For example, Luo et al. (2017) applied DBN to corporate credit scoring and found that the model's classification performance outperformed traditional algorithms such as LR, NN, and SVM. Shen et al. (2021) proposed an LSTM classification model for credit risk assessment which is also more competitive than other traditional algorithms. In addition to the advantages of prediction, deep learning has also shown significant advantages in mining the feature. Liu et al. (2022) based on the DNN algorithm to transform original features with a nonlinear relationship into more separable features. (Guo et al., 2022) used deep learning methods to analyze local government debt risk by mining hidden government sentiment in different texts. In general, when predicting CCR, DNN algorithms tend to achieve better prediction performance than traditional single-classifier methods and are good at mining more information.

2.2 High dimensional feature selection

Researchers often store large amounts of information for a more comprehensive analysis record during big data analysis (Mansi et al., 2011; Salkuti, 2020). While this massive amount of information can provide some benefits for optimal decision-making, it also complicates the dataset. Generally, as the feature dimension increases, a large amount of redundant and irrelevant information is usually introduced, which causes the poor performance of the machine learning algorithm (Ayesha et al., 2020; Danenas & Garsva, 2015; Tan et al., 2014). To date, there are two main methods for high dimensional data processing: feature extraction and feature selection. The former mainly combines different attributes to obtain new attributes, thus changing the original feature space (Guyon et al., 2008; Kuncheva et al., 2013), such as principal component analysis (Bro et al., 2014) and fuzzy sets (Hedrea et al., 2021). In contrast to the feature extraction, feature selection selects subsets from the original feature dataset without changing the original feature space (Chandrashekar et al., 2014; Li et al., 2017), and the commonly used feature selection methods are the screening method, encapsulation method, and embedding method (Li et al., 2017). The screening method is independent of the classifier used by the subsequent algorithm and is prone to deviation from the subsequent learning algorithm (Wang et al., 2019). The encapsulation method can obtain a higher classification accuracy when filtering parameters, but the selected features rely too much on the algorithm classifier, which easily leads to overfitting (Perez-Riverol et al., 2017). Compared with the former two methods the most commonly used feature selection method actually is the regularization-based Embedding method which incorporates feature selection into the algorithm optimization process to learn the most important properties in a given situation (Emmert-Streib & Dehmer, 2019; Ghaddar & Naoum-Sawaya, 2018; Salehi et al., 2019). For example, Tan et al. (2010) proposed the L1 sparse SVM algorithm for ultra-high-dimensional datasets, and proved that the algorithm were better than other competing algorithms; Pappu et al. (2015) added L1 regularization to the algorithm and removed more than 98% of the features of high dimensional datasets without affecting the performance of the algorithm. In addition, deep learning algorithms have also been applied to high dimensional feature mining. Erfani et al. (2016) used the DBN algorithm to convert high dimensional features into low-dimensional feature sets. Liu et al. (2017a) also pointed out that DNN algorithms have outstanding advantages when dealing with high dimensional data. However, L1 regularization is rarely used in DNN algorithms. The fundamental reason is that regularization cannot obtain regularization effect in DNN algorithm. For example, although Liu et al. (2017a) use the DNN algorithm when dealing with high dimensional data, it is only combined with the greedy algorithm to obtain the optimal solution. In other words, the advantages of L1 regularization for high dimensional data cannot be exploited in DNN algorithms.

According to the literature review, it can be seen that previous work has considered financial information closely related to companies to predict CCR, but they ignored the fact that non-financial information may also have an impact on CCR prediction results. Or worse, although the DNN algorithm shows excellent ability in CCR, it does not take advantage of L1 regularization in high dimensional feature selection. Therefore, this study introduces network information and supply chain information into CCR prediction, and proposes an improved DNN algorithm for high dimensional CCR dataset prediction.

3. Materials and methods

In this section, we introduce the data sources of this study and the prediction method for the high dimensional CCR dataset. The details are as follows.

3.1 Data sources

The data used in this study are from multiple sources such as Compustat and Bloomberg to form a high dimensional CCR prediction dataset. The data span from January 1, 2009 to December 31, 2019. In addition to corporate credit rating data, we also employ corporate financial data and non-financial data such as supply chain data and network data. Specific variables are shown as follows.

(1) Credit rating data. The credit risk data are mainly from the Compustat global database. The database is a comprehensive financial database with more than 5,000 accounting-adjusted items covering more than 50,000 listed companies worldwide. In total, this study collected 1,440 rating campaigns from 441 companies.

(2) Corporate financial data. Financial data for target companies are also collected from the Compustat global database. The data collected in this study involves the financial data published by these companies, including 19 data indicators such as working capital ratio, debt-equity ratio, retained earnings ratio, and weekly average daily return of bonds. These data can reflect changes in the company's policies and strategies for the entire market, as well as show the impact of financial performance on the company's operations.

(3) Network data. Online activity data such as search trends and website visits provide the latest information and are viewed as a complement to slower financial reporting (Fondeur et al., 2013; Phillips et al., 2018; Sousa-Pinto et al., 2020). The news data collected for this study comes from 10 indicators from the Wikipedia database, Google Trends database, and Facebook homepage text. These three databases have high visibility around the world and are often the first choice for people to search (Moat et al., 2016; Weng et al., 2017).

(4) Supply Chain Data. Supply chain data were primarily from the Bloomberg Supply Chain Database which contains more than 20,000 pieces of quantified supply chain data. Based on the data, this study can trace the upstream of the supply chain to locate suppliers throughout the supply chain, and also can trace the downstream to locate all customers of the main company. Data categories use each company's data for a total of 29 indicators.

To sum up, this study introduces network information and supply chain information on the basis of company financial information. Out of a total of 88 indicators, 87 indicators are used for prediction. Compared with the traditional CCR prediction dataset with at most 20 indicators (Wang et al., 2011; Zhang et al., 2021), this dataset has a higher dimension, which may lead to the curse of dimensionality.

3.2 DNN algorithm

Inspired by artificial NN (Liu et al., 2017b; Reagen et al., 2016), the DNN algorithm aims to enable computer programs to think like humans (Ni et al., 2021; Zheng et al., 2017). Although artificial neural networks have been widely used in various prediction tasks (Abiodun et al., 2019; Albu et al., 2019; Wang, 2003). the prediction performance of the artificial algorithm largely depends on the quality of the input features. Oppositely, the DNN algorithm can handle more complex situations by increasing the depth of the network, so it is widely used in image recognition, machine translation and other fields (Bouwmans et al., 2019; Shewalkar, 2019; Zhang et al., 2015a). Besides, the DNN algorithm can automatically extract features to pursue the objectivity of prediction results that may suffer from manually selected features (Chen et al., 2020a; Suryanarayana et al., 2018). In addition, (Liu et al., 2017a) also confirmed the advantages of DNN algorithms in dealing with high dimensional data. Therefore, this study uses DNN as the base algorithm to construct a prediction model. The basic framework is shown in Figure 1.

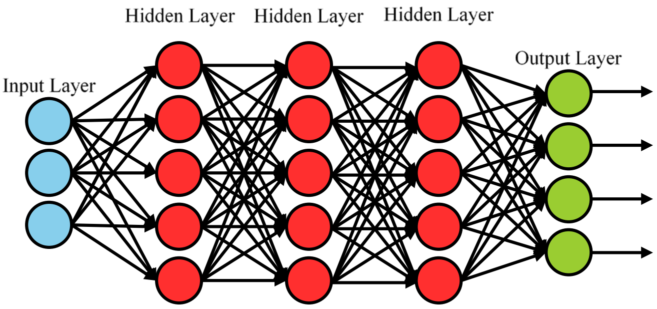


Figure 1 Basic structure of the DNN algorithm

As shown in Figure 1, the neural network layers inside DNN are divided into the input layer, hidden layer, and output layer. The layers are fully connected, that is, any neuron in the *i* layer must be connected to any neuron in the *i+1* layer. It is supposed the DNN has L layers where $\{x_1, x_2, \ldots\ldots, x_n\}$ is the input layer data, $\{y_1, y_2, \ldots\ldots, y_m\}$ is the output layer data, $\{h_1^{(l)}, h_2^{(l)}, h_3^{(l)}, \ldots\ldots, h_{n_l}^{(l)}\}$ is the output data of the *l* layer. $w_{ij}^{(l)}$ is the weight from the *j* neuron in the *l-1* layer to the *i* neuron in the *l* layer. $b_i^{(l)}$ is the bias of the *i* neuron in the I layer. $f(\cdot)$ is the activation function. This study uses the sigmoid and swish functions to enhance the learning ability of the neural network. Therefore, the *i* neuron y_i in the I layer is shown below.

$$y_i = f\left(\sum_{j=1}^{s_{L-1}} w_{ij}^{(l)} h_j^{(l-1)} + b_i^{(l)}\right) \tag{1}$$

At the same time, we also set up a dropout layer to prevent the model from over-reliance on local features. Batch Normalization is used to normalize the input to avoid deepening the number of the NN layers and making the model difficult to train. Subsequently, the Stochastic gradient descent algorithm is used to update the model parameters. Finally, the algorithm results are as close to the real credit degradation probability as possible. The specific values of DNN parameters are shown in Table 2.

Table 2 The DNN algorithm parameter settings

Number	**Parameters**	**Value**
1	Full connection layer	3
2	Learning rate	0.001
3	Dropout	0.5
4	Epoch	10000
5	Batch size	1024
6	Adam	Stochastic Gradient Descent

4. Results

For high dimensional datasets, this study process by improving the DNN algorithm. We propose to incorporate L2 constraints into a single L1 regularization for high dimensional feature selection. This not only allows the DNN algorithm to perform feature selection via L1 regularization, but also increases the L2 norm to prevent overfitting. The specific content and calculation results are as follows.

4.1 DNN for high dimensional datasets

High-dimensional datasets contain much decision-making information, but also contain many irrelevant or redundant features for a target task (Ayesha et al., 2020; Danenas & Garsva, 2015; Tan et al., 2014). Therefore, it is necessary to select truly relevant features through feature selection to improve the prediction accuracy, and L1 regularization is widely used to produce sparse models. However, L1 regularization is rarely used in DNN algorithms. The fundamental reason is that the combination of regularization and normalization will not produce regularization after adding L1 regularization to the DNN algorithm (Van Laarhoven, 2017), and L1 regularization will fail in the DNN. This means that the advantages of L1 regularization in dealing with high dimensional data cannot be exploited in DNN algorithms. The specific proof is as follows.

(1) L1 regularization failure proof

Assuming that the first fully connected layer of the model FC_1 is denoted as $\mathcal{F},$ the batch normalization layer $BN_1$is denoted as $\mathcal{G}$. The composite function of the subsequent layers is denoted as $\mathcal{H}$, then the model $F = \mathcal{H} \circ \mathcal{G} \circ \mathcal{F}$. Let the weight matrix of FC_1 is: $W = (\omega_1, \omega_2, \ldots, \omega_m)'$. Among them, *m* is the number of neurons in the fully connected layer, the bias vector is b, and the input matrix is X, then the output matrix of the fully connected layer $\mathcal{F}(x)$ is Wx.

Let the input mean and variance of each batch of input be $\bar{x}$ and D, then the mean and the variance of $\mathcal{F}(x)$ are $W\bar{x}$ and $W'DW$, respectively. Let $y = BN_1(\mathcal{F}(x))$, then the i component of the *y* layer is $y_i = \left((x-\bar{x})'W'(W'DW)^{-1}W(x-\bar{x})\right)_i^{1/2}$. Let FC_1 layer has an L1 regularization penalty, and the penalty coefficient is λ. Then the optimization objective of the model is:

$$z = \min_{W,b,\omega} [L(F) + \lambda \|W\|_1] \qquad （2）$$

Where, $F = \mathcal{H} \circ \mathcal{G} \circ \mathcal{F}$, $L(F)$ is the loss function of the algorithm.

$$L(F) = -\sum_{i=1}^{m} \left[r_i \log\left(F(x_i)\right) + (1 - r_i) \log\left(1 - F(x_i)\right)\right] \qquad （3）$$

m are the number of training samples. When the first i a training sample degradation r_i take 1, otherwise r_i take 0. x_i is the independent variable vector of the *i* training sample, and $F(x_i)$ is the degradation probability of the i training sample. $\lambda\|W\|_1$ is the given regularization penalty, and W and b are the weight matrix and bias coefficient vector of the full connection layer $\mathcal{F}$ respectively, and ω are other coefficients of the algorithm. Given the weight matrix $W_1 = W^*$, for any real number $\alpha > 0$, suppose another weight matrix, where: $W_2 = \alpha W^*$, therefore,

$$\begin{aligned} y_i^2 &= \left((x-\bar{x})'(\alpha W)'(\alpha W' D \alpha W)^{-1} \alpha W (x-\bar{x})\right)_i^{1/2} \\ &= \left((x-\bar{x})'(W)'(W'DW)^{-1}W(x-\bar{x})\right)_i^{1/2} = y_i^1 \end{aligned} \qquad （4）$$

It can be seen that for different weight matrices W_1、W_2, if $W_2 = \alpha W_1$, then $\mathcal{G} \circ \mathcal{F}_1 = \mathcal{G} \circ \mathcal{F}_2$, therefore, when $0 < \alpha < 1$

$$L(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F}\_1\,) + \lambda \|W\|\_1 \geq L(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F}\_2\,) + \lambda \|\alpha W\|\_1 \tag{5}$$

That, $\alpha \to 0$, then

$$L(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F}_1) + \lambda \|W\|_1 \geq L(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F}_2) \tag{6}$$

Because

$$L(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F}_2) + \lambda \|\alpha W\|_1 \geq \min_{W,b,\omega} [L(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F}) + \lambda \|W\|_1] \tag{7}$$

Therefore,

$$z = \min_{W,b,\omega} [L(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F}) + \lambda \|W\|_1] = \min_{W,b,\omega} [L(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F})] \tag{8}$$

That is, L1 regular penalty in the full connection layer $\mathcal{F}$ fails in the optimal solution. For a neuron in the first fully connected layer, its input matrix is X, its weight vector is w, and its activation function is g, then the output of the neuron after batch standardization is

$$y_{\mathrm{BN}}(X; w, \gamma, \beta) = g\left(\frac{Xw - \mu(Xw)}{\sigma(Xw)}\gamma + \beta\right) \tag{9}$$

Set, $w' = \alpha w$

$$y_{\mathrm{BN}}(X; w', \gamma, \beta) = y_{\mathrm{BN}}(X; \alpha w, \gamma, \beta) = g\left(\frac{X\alpha w - \mu(X\alpha w)}{\sigma(X\alpha w)}\gamma + \beta\right) = g\left(\frac{Xw - \mu(Xw)}{\sigma(Xw)}\gamma + \beta\right) = y_{\mathrm{BN}}(X; w, \gamma, \beta) \tag{10}$$

Suppose the loss function of the algorithm is

$$L_\lambda(w, \theta) = L(w, \theta) + \lambda \|w\|_1 \tag{11}$$

Among them, the $L(w, \theta)$ losses for the algorithm prediction results. $\lambda \|w\|_1$ is w L1 norm regularization losses, θ said other parameters of the algorithm. According to the structure of the NN, there is a function L_1 of y_{BN}, $L(w, \theta) = L_1(y_{\mathrm{BN}}(w), \theta)$. So, $L(\alpha w, \theta) = L_1(y_{\mathrm{BN}}(\alpha w), \theta) = L_1(y_{\mathrm{BN}}(w), \theta) = L(w, \theta)$, then

$$L_\lambda(\alpha w, \theta) = L(w, \theta) + \lambda \alpha \|w\|_1 = L_{\alpha\lambda}(w, \theta) \tag{12}$$

Where is any positive real number α. Suppose w, θ the optimal estimation is w^*, θ^*, then,

$$(w^*, \theta^*) = \arg\min L_\lambda(w, \theta) \tag{13}$$

Set w_1^*, θ_1^* be a set of parameter estimates that minimize $L(w, \theta)$, then

$$L(w_1^*, \theta_1^*) \leq L_\lambda(w^*, \theta^*) \leq \lim_{\alpha \to 0} L_\lambda(\alpha w_1^*, \theta_1^*) = \lim_{\alpha \to 0} L_{\alpha\lambda}(w_1^*, \theta_1^*) = L(w_1^*, \theta_1^*) \tag{14}$$

Then,

$$L(w_1^*, \theta_1^*) = L_\lambda(w^*, \theta^*), \qquad \lambda \|w^*\|_1 = 0 \tag{15}$$

Therefore, L1 regularization will fail in the DNN algorithm.

（2）Proposed Feature Selection

Although that the L1 norm regularization alone will fail in the DNN algorithm has been proved in the above section. It was found that adding certain constraints to the weights would make the L1 regularization still effective after further analyzing the model dynamics. Specifically, when adding an L2 norm constraint to the weight vector w of the neuron and set $\|w\|_2 = \alpha$, we obtain the optimal estimation of the model parameters w, θ is

$$(w^*, \theta^*) = \arg\min_{w,\theta} L_\lambda(w, \theta) \qquad s.t. \|w\|_2 = \alpha \tag{16}$$

By the Cauchy inequality

$$1 \leq \|w\|_1 \leq \sqrt{k} \tag{17}$$

Where k is the dimension of w, so L1 norm regularization is still valid. Analyzing the dynamics of the model, we have the following findings.

Proposition 1: For batch regularization layer $y_{\mathrm{BN}}(X; w, \gamma, \beta)$, its intra-layer weight has scale invariance, namely $y_{\mathrm{BN}}(X; \alpha w, \gamma, \beta) = y_{\mathrm{BN}}(X; \alpha w, \gamma, \beta)$, but the gradient of its weight is inversely proportional to its scale, namely $\nabla y_{\mathrm{BN}}(X; w, \gamma, \beta) = \frac{1}{\alpha} \nabla y_{\mathrm{BN}}(X; \alpha w, \gamma, \beta)$

Proof: Suppose, $y_i = X_i \boldsymbol{w} = \sum_{j=1}^{p} x_{ij} \omega_j$,$\hat{y}_i = \frac{y_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$,$z_i = \hat{y}_i \gamma + \beta$, then

$$\frac{\partial l}{\partial \hat{y}_i} = \frac{\partial l}{\partial z_i}\gamma,\ \frac{\partial l}{\partial \hat{\boldsymbol{y}}} = \frac{\partial l}{\partial \boldsymbol{z}}\gamma,\ \frac{\partial l}{\partial \sigma_{\mathcal{B}}^2} == -\frac{1}{2}\gamma(\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2} \frac{\partial l}{\partial \boldsymbol{z}} \cdot (\boldsymbol{y} - \mu_{\mathcal{B}}),\ \frac{\partial l}{\partial \mu_{\mathcal{B}}} = -\frac{\gamma}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \frac{\partial l}{\partial \boldsymbol{z}} \cdot \boldsymbol{1}$$

$$\frac{\partial l}{\partial y_i} = \gamma \frac{\partial l}{\partial z_i} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} - \frac{1}{2}\gamma(\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2} \frac{\partial l}{\partial \boldsymbol{z}} \cdot (\boldsymbol{y} - \mu_{\mathcal{B}}) \cdot \frac{2(y_i - \mu_{\mathcal{B}})}{m} - \frac{\gamma}{m\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \frac{\partial l}{\partial \boldsymbol{z}} \cdot \boldsymbol{1}$$

$$\frac{\partial l}{\partial \boldsymbol{y}} = \frac{\gamma}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \frac{\partial l}{\partial \boldsymbol{z}} - \frac{\gamma(\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2}}{m} \frac{\partial l}{\partial \boldsymbol{z}} \cdot (\boldsymbol{y} - \mu_{\mathcal{B}})(\boldsymbol{y} - \mu_{\mathcal{B}})^T - \frac{\gamma}{m\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \frac{\partial l}{\partial \boldsymbol{z}} \boldsymbol{1} \cdot \boldsymbol{1}^T$$

$$\frac{\partial l}{\partial \boldsymbol{w}} = \frac{\partial l}{\partial \boldsymbol{y}} X = \frac{\gamma}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \frac{\partial l}{\partial \boldsymbol{z}} \left(I - \frac{1}{m(\sigma_{\mathcal{B}}^2 + \epsilon)} (\boldsymbol{y} - \mu_{\mathcal{B}}) \cdot (\boldsymbol{y} - \mu_{\mathcal{B}})^T - \frac{1}{m} \boldsymbol{1} \cdot \boldsymbol{1}^T \right) X \tag{18}$$

Then,

$$\frac{\partial l}{\partial \alpha \boldsymbol{w}} = \frac{1}{\alpha} \cdot \frac{\partial l}{\partial \boldsymbol{w}} \tag{19}$$

$$\nabla y_{\mathrm{BN}}(X; w, \gamma, \beta) = \frac{1}{\alpha} \nabla y_{\mathrm{BN}}(X; \alpha w, \gamma, \beta) \tag{20}$$

In fact, for any weight with scale invariance, there are $f(\boldsymbol{w}) = f(\alpha \boldsymbol{w})$, so

$$\nabla f(\boldsymbol{w}) = \frac{1}{\alpha} \nabla f(\alpha \boldsymbol{w}) \tag{21}$$

Proposition 1 shows that for batch regularization layers, although the scale change of the weight vector will not change the layer output, it will change the gradient of the layer and affect the model training.

Proposition 2: For the batch regularization layer, let its weights be constrained by the 2-norm, that is, $\|w\|_2 = \alpha$. Meanwhile, we add a 1-norm penalty to the loss function, which can make the weight coefficients sparse under certain conditions. At this time, the degree of sparsity is related to the 2-norm constraint value α and the 1-norm penalty coefficient.

Proof: Let the objective function of model optimization be $L_\lambda(\boldsymbol{w}, \theta, X, y) = L(\boldsymbol{w}, \theta, X, y) + \lambda \|\boldsymbol{w}\|_1$, we get

$$\nabla L_\lambda(\boldsymbol{w}, \theta, X, y) = \nabla L(\boldsymbol{w}, \theta, X, y) + \lambda\, sign(\boldsymbol{w}) \tag{22}$$

Let $L(\boldsymbol{w}, \theta, X, y)$ obtain the minimum value at $\boldsymbol{w}^*$, at this time $\nabla L(\boldsymbol{w}^*, \theta, X, y) = 0$. Perform second-order Taylor expansion on $L_\lambda(\boldsymbol{w}, \theta, X, y)$ at $\boldsymbol{w}^*$, then

$$L_\lambda(\boldsymbol{w},\theta,X,y) = L(\boldsymbol{w}^*,\theta,X,y) + \lambda\|\boldsymbol{w}\|_1 + \frac{1}{2}(\boldsymbol{w}-\boldsymbol{w}^*)'H(\boldsymbol{w}-\boldsymbol{w}^*) \qquad (23)$$

Where H is the Hessian matrix of $L_\lambda(\boldsymbol{w},\theta,X,y)$. Orthogonal decomposition of H, let $H = Q'\Lambda Q$, where Q is an orthogonal matrix. Λ is a diagonal matrix, set as $\mathrm{diag}(h_{11}, h_{22}, \ldots, h_{mm})$. Since H is positive and semi–definite, $h_{ii} \geq 0$. Therefore,

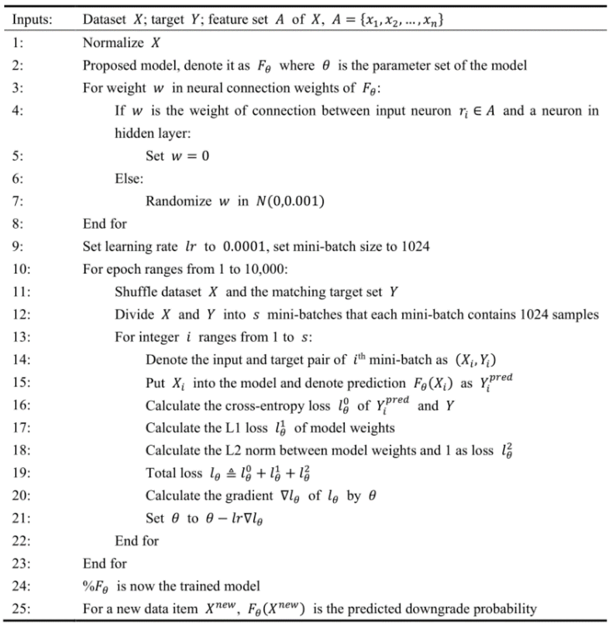
$$L_\lambda(\boldsymbol{w},\theta,X,y) = L(\boldsymbol{w}^*,\theta,X,y) + \sum_{i=1}^{m}\left(\lambda|w_i| + \frac{1}{2}h_{ii}(w_i - w_i^*)^2\right) \qquad (24)$$

Obviously, when we get to the minimum $L_\lambda(\boldsymbol{w},\theta,X,y)$, there are $sign(w_i) = 0$ or $sign(w_i) = sign(w_i^*)$. Let $\nabla L_\lambda(\boldsymbol{w},\theta,X,y) = \lambda\, sign(\boldsymbol{w}) + H(\boldsymbol{w}-\boldsymbol{w}^*) = 0$, substitute $sign(w_i) = sign(w_i^*)$ to get

$$w_i = sign(w_i^*)\left(|w_i^*| - \frac{h_{ii}}{\lambda}\right) \qquad (25)$$

Therefore, $w_i = sign(w_i^*)\max\left\{|w_i^*| - \frac{h_{ii}}{\lambda}, 0\right\}$. This means that when $|w_i^*| < \frac{h_{ii}}{\lambda}$, $w_i = 0$; when $|w_i^*| \geq \frac{h_{ii}}{\lambda}$, $w_i = sign(w_i^*)\left(|w_i^*| - \frac{h_{ii}}{\lambda}\right)$, so $\boldsymbol{w}$ sparse. And because the scale-invariant $L(\boldsymbol{w},\theta,X,y)$, $h_{ii}(\boldsymbol{w}) = \frac{1}{\alpha^2}h_{ii}(\alpha\boldsymbol{w})$, so the degree of sparsity is related to the 2-norm constraint values α and 1-norm Penalty coefficients are related. Theorem 2 shows that under the 2-norm constraint, the penalty coefficient λ and 2-norm constraint value α have the same effect on the sparsity of weight $\boldsymbol{w}$. Therefore, during hyperparameter debugging of the algorithm, one value can be fixed, and the other value can be debugged.

In summary, this study proposes the HDNN algorithm that is an improved DNN algorithm for high dimensional CCR datasets. We add L2 constraints on a single L1 regularization to prevent L1 regularization from failing in the DNN algorithm. Specifically, the L1 norm with a penalty coefficient of 0.01 is added to the first fully connected layer of the algorithm, and the L2 norm of its weight is constrained to 1. In this way, the sparse solution of the algorithm is obtained through L1 regularization, and the L2 norm is also added to better cope with the overfitting. The pseudo-code of the HDNN algorithm is shown in Figure 2.



```
Inputs: Dataset $X$; target $Y$; feature set $A$ of $X$, $A = \{x_1, x_2, \dots, x_n\}$
Normalize $X$
Proposed model, denote it as $F_\theta$ where $\theta$ is the parameter set of the model
For weight $w$ in neural connection weights of $F_\theta$:
    If $w$ is the weight of connection between input neuron $r_i \in A$ and a neuron in hidden layer:
        Set $w = 0$
    Else:
        Randomize $w$ in $N(0,0.001)$
End for
Set learning rate $lr$ to 0.0001, set mini-batch size to 1024
For epoch ranges from 1 to 10,000:
    Shuffle dataset $X$ and the matching target set $Y$
    Divide $X$ and $Y$ into $s$ mini-batches that each mini-batch contains 1024 samples
    For integer $i$ ranges from 1 to $s$:
        Denote the input and target pair of $i^{\text{th}}$ mini-batch as $(X_i, Y_i)$
        Put $X_i$ into the model and denote prediction $F_\theta(X_i)$ as $Y_i^{pred}$
        Calculate the cross-entropy loss $l_\theta^0$ of $Y_i^{pred}$ and $Y$
        Calculate the L1 loss $l_\theta^1$ of model weights
        Calculate the L2 norm between model weights and 1 as loss $l_\theta^2$
        Total loss $l_\theta \triangleq l_\theta^0 + l_\theta^1 + l_\theta^2$
        Calculate the gradient $\nabla l_\theta$ of $l_\theta$ by $\theta$
        Set $\theta$ to $\theta - lr\nabla l_\theta$
    End for
End for
%$F_\theta$ is now the trained model
For a new data item $X^{new}$, $F_\theta(X^{new})$ is the predicted downgrade probability
```

Figure 2 The pseudo-code of the HDNN algorithm

4.2 Methods evaluation

The predictive effect of the HDNN algorithm was assessed firstly by using the area under the curve (AUC) of the receiver operating characteristic (ROC) (Fan et al., 2006; Pepe, 2000). ROC and AUC are metrics to measure the effectiveness of the learner. The closer the ROC curve approaches the upper left corner, the better the accuracy of the model is. In other words, a classifier with a larger AUC value has higher accuracy. Secondly, when testing the generalization performance of the model, we cannot simply use k-fold cross-validation for the chronological order of the dataset in this study, otherwise, it will lead to predicting past phenomena with the future (Bergmeir et al., 2012; Bergmeir et al., 2018). Therefore, this study performs subsequent segmentation of time series whereby sliding windows. Specifically, on the basis of the common 70% training sample division in time-series datasets (Siami-Namini et al., 2018; Xue et al., 2011), we fluctuate 10% of the training samples up and down and calculate 10-time average in each dataset (Meng et al., 2018). In addition, we also compared the HDNN algorithm with traditional machine learning algorithms such as SVM, NN, LR and used non-parametric tests to evaluate the significance of the HDNN algorithm.

4.3 Methods results

This study uses the high dimensional CCR dataset as input to test the HDNN algorithm's predictive ability. According to the HDNN algorithm, we set parameters and input commands on the python platform. Furthermore, we allocated training samples and test samples by a ratio of 70% to 30% in line with a common way of sample division for time series (Siami-Namini & Namin, 2018; Xue et al., 2011). By applying the data to the algorithm, we get the ROC of the HDNN algorithm. As shown in Figure 3, the prediction accuracy AUC is 80.12%, which suggests great performance.

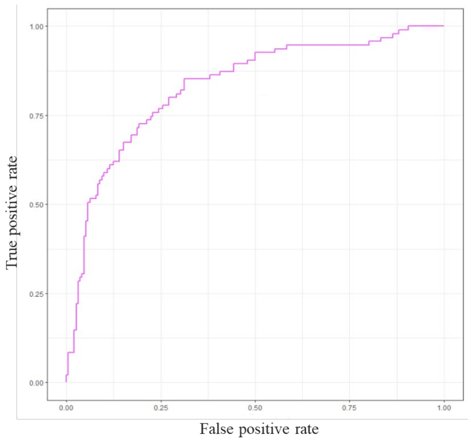


Figure 3 The ROC curve of the HDNN algorithm

5. Discussion

This section discusses the predictive effectiveness of the HDNN algorithm. As can be seen from Section 2, Section 3 and Section 4, this study considers the impact of network information and supply chain information on CCR, and proposes an HDNN algorithm for predicting high dimensional CCR data. To address the high dimensional feature selection problem, we theoretically theoretically prove that there is no regularization effect when L1 regularization is added to the batch normalization layer of the DNN, which is a hidden rule in the industrial implementation but never been proved. Also, we proved that adding L2 constraints on a single L1 regularization can solve this issue. Therefore, to verify the prediction ability of the HDNN algorithm, we conduct a comparison between the HDNN algorithm's performance for CCR prediction and other competing algorithms, such as NN, SVM, LR. Notably, the remaining SVM, NN, LR, and other classifiers are set according to the default software package to avoid the deviation of the results caused by the artificially specified parameters. In addition, we fluctuate 10% of the training samples up and down on 70% of the training samples and perform 10 computations on each dataset separately, because time series data does not allow random sampling cross-validation (Meng et al., 2018; Siami-Namini & Namin, 2018; Xue et al., 2011). The specific values are shown in Table 3.

Table 3 The AUC of different proportional training sample

Methods	60% training set	70% training set	80% training set
NN	0.668	0.692	0.674
SVM	0.713	0.738	0.702
LR	0.696	0.717	0.726
HDNN	0.787	0.801	0.794

Table 3 shows the prediction results of HDNN and popular machine learning algorithms. The AUC values that measure the prediction accuracy reveal that, despite the different training samples, the HDNN algorithm still has the highest prediction accuracy. The second ranking is the SVM algorithm. The final decision function of the SVM algorithm is determined by the number of support vectors, thus leading to unique advantages when dealing with high dimensional data (Ni et al., 2018). The performance of NN and LR algorithms is average, which may be due to the fact that a large number of indicators have been added to the dataset and the dimension is too high. On the whole, compared with the current popular competing algorithms, the HDNN algorithm works best.

Moreover, this study also uses nonparametric tests to evaluate differences among these algorithms. Specifically, the Friedman test is adopted to examine the differences in algorithm performance on different datasets. Since the statistical result is $\chi^2_{(30)}$ = 8.2, p = 0.04, which suggests there is a significant performance difference between the algorithms. It means that the HDNN algorithm can be applied to high dimensional CCR prediction in a targeted manner. This is the main contribution of our method to the current high dimensional CCR prediction.

6. Conclusions

This study clearly reveals the effectiveness of the HDNN algorithm in predicting high dimensional datasets, and proves the superiority of the proposed algorithm by comparing with existing algorithms. Specifically, we consider the impact of external information, including supply chain information and network information, on CCR. Although more information has been shown to be helpful in improving the CCR prediction accuracy (Wu et al., 2022), this often leads to a dramatic increase in data dimensionality, which can greatly reduce the performance of prediction algorithms along with redundant and irrelevant information. Therefore, we propose the HDNN algorithm to predict high dimensional CCR. For high dimensional datasets, we theoretically proved that there was no regularization effect when L1 regularization was added to the batch normalization layer of the DNN, which was a hidden rule in the industrial implementation but never been proved. This study solved this issue by adding L2 constraints on a single L1 regularization, which not only performed feature selection through L1 regularization but also added L2 norm to better cope with the overfitting problem. Finally, this study analyzed a real case including supply chain and network information, and obtains the prediction accuracy of the HDNN algorithm is 80.12%, showing the superiority of the HDNN algorithm in the scenario of a high dimensional dataset. Such enhanced high dimensional CCR prediction is very important in practice. For example, adding supply chain information helps protect a company's industrial chain from supply chain risk spillovers. Timely network information can help companies to identify the early signs that could damage their financial stability. In addition, for the company owner, early detection and prediction of the company's credit situation, even a slight increase in prediction accuracy, can reduce future risks and translate into company benefits. Overall, the HDNN algorithm offers a possible solution for great accuracy prediction of high dimensional CCR.

Despite the outstanding performance of the HDNN algorithm, some directions deserve further consideration. First, in addition to supply chain and network information, many other factors may also affect credit risk such as executive behavior variables, customer profiles, and announcement text features. Therefore, richer data need to be incorporated into CCR prediction to help gain more management implications. Second, although the new information introduced in this study makes contributions to the CCR prediction to a certain degree, these data are acquired indirectly by calculating from the database. We believed that if the original data can be obtained, the performance of the HDNN algorithm may be further enhanced. Finally, this study mainly focuses on the prediction of high dimensional CCR, and the subsequent research can concentrate on the unbalanced and unstructured characteristics of the dataset in parallel to further enhance the interpretability of the model and obtain relevant management insights.

References

Abiodun, O. I., Jantan, A., et al. (2019). Comprehensive review of artificial neural network applications to pattern recognition. *Ieee Access, 7*, 158820-158846.

Agca, S., Babich, V., et al. (2021). Credit shock propagation along supply chains: Evidence from the CDS market. *Management Science*.

Albu, A., Precup, R.-E., et al. (2019). Results and challenges of artificial neural networks used for decision-making and control in medical applications. *Facta Universitatis, Series: Mechanical Engineering, 17*(3), 285-308.

Arias-Castro, E., Pelletier, B., et al. (2018). Remember the curse of dimensionality: The case of goodness-of-fit testing in arbitrary dimension. *Journal of Nonparametric Statistics, 30*(2), 448-471.

Ayesha, S., Hanif, M. K., et al. (2020). Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion, 59*, 44-58.

Barboza, F., Kimura, H., et al. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications, 83*, 405-417.

Basturk, O., & Cetek, C. (2021). Prediction of aircraft estimated time of arrival using machine learning methods. *Aeronautical Journal, 125*(1289), 1245-1259.

Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences, 191*, 192-213.

Bergmeir, C., Hyndman, R. J., et al. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis, 120*, 70-83.

Bhatore, S., Mohan, L., et al. (2020). Machine learning techniques for credit risk evaluation: a systematic literature review. *Journal of Banking and Financial Technology, 4*(1), 111-138.

Bonsall IV, S. B., Green, J. R., et al. (2018). Are credit ratings more rigorous for widely covered firms? *The Accounting Review, 93*(6), 61-94.

Borlea, I.-D., Precup, R.-E., et al. (2021). A unified form of fuzzy C-means and K-means algorithms and its partitional implementation. *Knowledge-Based Systems, 214*, 106731.

Bouwmans, T., Javed, S., et al. (2019). Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. *Neural Networks, 117*, 8-66.

Bro, R., & Smilde, A. K. (2014). Principal component analysis. *Analytical methods, 6*(9), 2812-2831.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering, 40*(1), 16-28.

Chang, Y.-C., Chang, K.-H., et al. (2018). Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. *Applied Soft Computing,* *73*, 914-920.

Chen, S. X., Wang, X. K., et al. (2021). Customer purchase prediction from the perspective of imbalanced data: A machine learning framework based on factorization machine. *Expert Systems with Applications, 173*.

Chen, Z., Pang, M., et al. (2020a). Feature selection may improve deep neural networks for the bioinformatics problems. *Bioinformatics, 36*(5), 1542-1552.

Chen, Z. S., Li, C. H., et al. (2020b). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics, 365*.

Chiang, H.-S., Shih, D.-H., et al. (2014). An APN model for Arrhythmic beat classification. *Bioinformatics, 30*(12), 1739-1746.

Chowdhury, P., Paul, S. K., et al. (2021). COVID-19 pandemic related supply chain studies: A systematic review. *Transportation Research Part E: Logistics and Transportation Review, 148*, 102271.

Cisi, M., Devicienti, F., et al. (2020). The advantages of formalizing networks: new evidence from Italian SMEs. *Small Business Economics, 54*(4), 1183-1200.

Danenas, P., & Garsva, G. (2015). Selection of Support Vector Machines based classifiers for credit risk domain. *Expert Systems with Applications, 42*(6), 3194-3204.

Dutta, S., & Shekhar, S. (1988). *Bond rating: a non-conservative application of neural networks.* Paper presented at the IEEE Int Conf on Neural Networks.

Emmert-Streib, F., & Dehmer, M. (2019). High-dimensional LASSO-based computational regression models: regularization, shrinkage, and selection. *Machine Learning and Knowledge Extraction, 1*(1), 359-383.

Erfani, S. M., Rajasegarar, S., et al. (2016). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition, 58*, 121-134.

Fan, J., Upadhye, S., et al. (2006). Understanding receiver operating characteristic (ROC) curves. *Canadian Journal of Emergency Medicine, 8*(1), 19-20.

Fernández-Martínez, J. L., & Fernández-Muñiz, Z. (2020). The curse of dimensionality in inverse problems. *Journal of Computational and Applied Mathematics, 369*, 112571.

Fondeur, Y., & Karamé, F. (2013). Can Google data help predict French youth unemployment? *Economic Modelling, 30*, 117-125.

Ghaddar, B., & Naoum-Sawaya, J. (2018). High dimensional data classification and feature selection using support vector machines. *European Journal of Operational Research, 265*(3), 993-1004.

Guo, Y., Li, Y., et al. (2022). Local government debt risk assessment: A deep learning-based perspective. *Information Processing & Management, 59*(3), 102948.

Guyon, I., Gunn, S., et al. (2008). *Feature extraction: foundations and applications* (Vol. 207): Springer

Han, J. C., Zhang, Z., et al. (2020). Prediction of Winter Wheat Yield Based on Multi-Source Data and Machine Learning in China. *Remote Sensing, 12*(2).

Hassani, Z., Alambardar Meybodi, M., et al. (2020). Credit risk assessment using learning algorithms for feature selection. *Fuzzy Information and Engineering, 12*(4), 529-544.

Hedrea, R.-C. R., & PETRIU, E. M. (2021). Evolving Fuzzy Models of Shape Memory Alloy Wire Actuators. *SCIENCE AND TECHNOLOGY, 24*(4), 353-365.

Kiesel, F. (2021). It's the tone, stupid! Soft information in credit rating reports and financial markets. *Journal of Financial Research, 44*(3), 553-585.

Kim, J., Shin, N., et al. (2017). *Method of intrusion detection using deep neural network.* Paper presented at the 2017 IEEE international conference on big data and smart computing (BigComp).

Kuncheva, L. I., & Faithfull, W. J. (2013). PCA feature extraction for change detection in multidimensional unlabeled data. *IEEE transactions on neural networks and learning systems, 25*(1), 69-80.

Lappas, P. Z., & Yannacopoulos, A. N. (2021). A machine learning approach combining expert knowledge with genetic algorithms in feature selection for credit risk assessment. *Applied Soft Computing, 107*, 107391.

Lev, B. (2018). The deteriorating usefulness of financial report information and how to reverse it. *Accounting and Business Research, 48*(5), 465-493.

Li, J., Cheng, K., et al. (2017). Feature selection: A data perspective. *ACM computing surveys (CSUR), 50*(6), 1-45.

Liu, B., Wei, Y., et al. (2017a). *Deep Neural Networks for High Dimension, Low Sample Size Data.* Paper presented at the IJCAI.

Liu, J., Zhang, S., et al. (2022). A two-stage hybrid credit risk prediction model based on XGBoost and graph-based deep neural network. *Expert Systems with Applications, 195*, 116624.

Liu, W., Wang, Z., et al. (2017b). A survey of deep neural network architectures and their applications. *Neurocomputing, 234*, 11-26.

Luo, C., Wu, D., et al. (2017). A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence, 65*, 465-470.

Ma, X., & Lv, S. (2019). Financial credit risk prediction in internet finance driven by machine learning. *Neural Computing and Applications, 31*(12), 8359-8367.

Mansi, S. A., Maxwell, W. F., et al. (2011). Analyst forecast characteristics and the cost of debt. *Review of Accounting Studies, 16*(1), 116-142.

Meng, F., Yue, T., et al. (2018). *SFSC: Segment Feature Sampling Classifier for Time Series Classification.* Paper presented at the International Conference of Pioneering Computer Scientists, Engineers and Educators.

Moat, H. S., Olivola, C. Y., et al. (2016). Searching choices: Quantifying decision-making processes using search engine data. *Topics in cognitive science, 8*(3), 685-696.

Moosavi, J., Fathollahi-Fard, A. M., et al. (2022). Supply chain disruption during the COVID-19 pandemic: Recognizing potential disruption management strategies. *International Journal of Disaster Risk Reduction*, 102983.

Mylonakis, J., & Diacogiannis, G. (2010). Evaluating the likelihood of using linear discriminant analysis as a commercial bank card owners credit scoring model. *International business research, 3*(2), 9.

Ni, D., Xiao, Z., et al. (2020). A systematic review of the research trends of machine learning in supply chain management. *International Journal of Machine Learning and Cybernetics, 11*(7), 1463-1482.

Ni, D., Xiao, Z., et al. (2021). Machine learning in recycling business: an investigation of its practicality, benefits and future trends. *Soft Computing, 25*(12), 7907-7927.

Ni, D., Xiao, Z., et al. (2018). Multiple human-behaviour indicators for predicting lung cancer mortality with support vector machine. *Scientific reports, 8*(1), 1-10.

Pappu, V., Panagopoulos, O. P., et al. (2015). Sparse Proximal Support Vector Machines for feature selection in high dimensional datasets. *Expert Systems with Applications, 42*(23), 9183-9191.

Pepe, M. S. (2000). Receiver operating characteristic methodology. *Journal of the American Statistical Association, 95*(449), 308-311.

Perez-Riverol, Y., Kuhn, M., et al. (2017). Accurate and fast feature selection workflow for high-dimensional omics data. *Plos One, 12*(12), e0189875.

Phillips, C. A., Leahy, A. B., et al. (2018). Relationship between state-level Google online search volume and cancer incidence in the United States: retrospective study. *Journal of medical Internet research, 20*(1), e8870.

Psillaki, M., Tsolas, I. E., et al. (2010). Evaluation of credit risk based on firm performance. *European Journal of Operational Research, 201*(3), 873-881.

Reagen, B., Whatmough, P., et al. (2016). *Minerva: Enabling low-power, highly-accurate deep neural network accelerators.* Paper presented at the 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA).

Roukny, T., Battiston, S., et al. (2018). Interconnectedness as a source of uncertainty in systemic risk. *Journal of Financial Stability, 35*, 93-106.

Ryu, Y. U., & Yue, W. T. (2005). Firm bankruptcy prediction: experimental comparison of isotonic separation and other classification approaches. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 35*(5), 727-737.

Salehi, F., Abbasi, E., et al. (2019). The impact of regularization on high-dimensional logistic regression. *Advances in Neural Information Processing Systems, 32*.

Salkuti, S. R. (2020). A survey of big data and machine learning. *International Journal of Electrical & Computer Engineering (2088-8708), 10*(1).

Shen, F., Zhao, X., et al. (2021). A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. *Applied Soft Computing, 98*, 106852.

Shewalkar, A. (2019). Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. *Journal of Artificial Intelligence and Soft Computing Research, 9*(4), 235--245.

Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: ARIMA vs. LSTM. *arXiv preprint arXiv:1803.06386*.

Singh, S., Kumar, R., et al. (2021). Impact of COVID-19 on logistics systems and disruptions in food supply chain. *International Journal of Production Research, 59*(7), 1993-2008.

Sousa-Pinto, B., Anto, A., et al. (2020). Assessment of the impact of media coverage on COVID-19–related Google trends data: Infodemiology study. *Journal of medical Internet research, 22*(8), e19611.

Suryanarayana, G., Lago, J., et al. (2018). Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods. *Energy, 157*, 141-149.

Tan, G. W.-H., Ooi, K.-B., et al. (2014). Predicting the drivers of behavioral intention to use mobile learning: A hybrid SEM-Neural Networks approach. *Computers in Human Behavior, 36*, 198-213.

Tan, M., Wang, L., et al. (2010). *Learning sparse svm for feature selection on very high dimensional datasets.* Paper presented at the ICML.

Trustorff, J.-H., Konrad, P. M., et al. (2011). Credit risk prediction using support vector machines. *Review of Quantitative Finance and Accounting, 36*(4), 565-581.

Upadhyay, P. K., & Nagpal, C. (2020). Wavelet Based Performance Analysis of SVM and RBF Kernel for Classifying Stress Conditions of Sleep EEG. *SCIENCE AND TECHNOLOGY, 23*(3), 292-310.

Van Laarhoven, T. (2017). L2 regularization versus batch and weight normalization. *arXiv preprint* *arXiv:1706.05350*.

Wang, G., & Ma, J. (2011). Study of corporate credit risk prediction based on integrating boosting and random subspace. *Expert Systems with Applications, 38*(11), 13871-13878.

Wang, M., & Barbu, A. (2019). Are screening methods useful in feature selection? An empirical study. *Plos One, 14*(9), e0220842.

Wang, S.-C. (2003). Artificial neural network. In *Interdisciplinary computing in java programming* (pp. 81-100): Springer

Wei, L., Li, G., et al. (2019). Discovering bank risk factors from financial statements based on a new semi-supervised text mining algorithm. *Accounting & Finance, 59*(3), 1519-1552.

Weng, B., Ahmed, M. A., et al. (2017). Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications, 79*, 153-163.

Wu, J., Zhang, Z., et al. (2022). Credit Rating Prediction Through Supply Chains: A Machine Learning Approach. *Production and Operations Management, 31*(4), 1613-1629.

Xue, X., Zhang, W., et al. (2011). *Correlative multi-label multi-instance image annotation.* Paper presented at the 2011 International Conference on Computer Vision.

Yang, M., Lim, M. K., et al. (2022). Repair missing data to improve corporate credit risk prediction accuracy with multi-layer perceptron. *Soft Computing, 26*(18), 9167-9178.

Zhang, H., Shi, Y., et al. (2021). A firefly algorithm modified support vector machine for the credit risk assessment of supply chain finance. *Research in International Business and Finance, 58*, 101482.

Zhang, J., & Zong, C. (2015a). Deep Neural Networks in Machine Translation: An Overview. *IEEE Intell. Syst., 30*(5), 16-25.

Zhang, L., Hu, H., et al. (2015b). A credit risk assessment model based on SVM for small and medium enterprises in supply chain finance. *Financial Innovation, 1*(1), 14.

Zhang, W., Yan, S., et al. (2022). Credit risk prediction of SMEs in supply chain finance by fusing demographic and behavioral data. *Transportation Research Part E: Logistics and Transportation Review, 158*, 102611.

Zheng, N.-n., Liu, Z.-y., et al. (2017). Hybrid-augmented intelligence: collaboration and cognition. *Frontiers of Information Technology & Electronic Engineering, 18*(2), 153-179.