Predicting Company’s Financial Failure or Bankruptcy Using Financial Ratios and Deep Learning

A Project Report Submitted by

Prathap Karunakaran (M20MA007)

In partial fulfillment of the requirements for the award of the degree of

M.Tech

Text

Description automatically generated

**Declaration**

I hereby declare that the work presented in this Project Report titled Predicting Company’s Financial Failure or Bankruptcy Using Financial Ratios and Deep Learning Report –M.Tech submitted to the Indian Institute of Technology Jodhpur in partial fulfilment of the requirements for the award of the degree of M.Tech., is a bonafide record of the research work carried out under the supervision of Dr.Manish Agarwal. The contents of this Project Report in full or in parts, have not been submitted to, and will not be submitted by me to, any other Institute or University in India or abroad for the award of any degree or diploma.

Signature

Prathap Karunakaran (M20MA001)

Abstract

Predicting Financial Failure or Bankruptcy is a very important task for financial institutions. Not only for lending but for building profile of an organization, its stability plays an important role. Though, there has been wide variety of research that is going on to predicting bankruptcy, research published by Liang .D et al. in explaining role of Financial Ratio’s and Corporate Governance Indicators in predicting Bankruptcy or financial failure is an important one. The research or implementation we have done is to compare and analyze performance of Machine Learning methods with some of the Deep Learning methods to obtain better result in terms of predicting financial failures. For evaluating the models, we have used train and test loss along with Accuracy, Type 1 and Type 2 Errors. In addition, LSTM with DenseNet obtained good result comparing to previous studies with the same datasets.

Contents:

# Introduction

# Background

## Data Profiling

## Implementation

### Artificial Neural Network

### Long Short Term Memory

### DenseNet

### Recurrent Neural Network

2.3. Validation

# Experiments and Results

4 Reference

1. Introduction:

The Prediction of Bankruptcy is a very important task for financial institutions. Their main aim is to predict the likelihood that a firm may go bankrupt or not. Financial institutions are in need of effective prediction models in order to make appropriate lending decisions, stock price prediction and more. Financial ratios can be considered as crucial determinant of reliability for the bankruptcy prediction models in addition to Corporate Governance Indicators. Financial Ratio’s can be classified into seven categories: solvency, profitability, cash flow ratios, capital structure ratios, turnover ratios, growth, and others. Generally, CGIs can be classified into five categories: broad structure. All of them plays a role in solving this classification problem. Further we look into various development in the fields along with some of the implementations.

1. Background

Financial failure prediction is a very essential and important task that occupies the efforts of many researchers, since an inaccurate decision about the companies’ financial status could cause costly financial losses. The prediction of companies’ financial status might be done using statistical techniques like Linear Discriminant Analysis (LDA), Multi-Discriminant Analysis (MDA) and Logistic Regression (LR or Logit); or by ML algorithms [6]. Altman [11] used MDA to predict companies’ financial status using their financial statements. Later, Ohlson [12] adopted Logit to predict companies’ financial failure. Brozyna et al. [5] used Linear Discriminant Analysis and Logistic Regression to predict the financial status of Polish and Slovak companies. Jones and Hensher [10] proposed a mixed Logit model, and compared it with a typical Logit model in predicting companies financial distress, proving that the mixed Logit model yields better results than the standard one. More recently, several researchers have compared the statistical techniques with ML techniques on forecasting companies’ financial failure or Bankruptcy. For instance, Pompe and Feelders [9] compared the performance of Linear Discriminant Analysis with classification trees and neural networks in this problem, and proved that neural networks outperform the rest of methods. Min and Lee [8] compared SVM, MDA, Logit and three-layer fully connected back-propagation neural networks regarding bankruptcy prediction, with SVM obtaining the best results. However, in recent studies, ML algorithms showed better performance than the statistical models concerning bankruptcy prediction. For this reason, many researchers have considered it as a classification problem, and have applied standard ML classification or regression methods for prediction [2], [3].

Some researchers combined several ML algorithms so as to enhance the efficiency of the companies’ financial failure prediction. Fedorova et al. [1] applied several combinations of RBF (Radial Basis Function) network and MLP in order to predict Russian companies’ bankruptcy, applied to a balanced dataset from all the available data. Iturriaga et al.[5] combined MLP and SOM (Self-Organized Maps) in order to predict US banks’ financial failure up to three years before it occurs. Another balanced dataset was employed by Lanbouri et al., who proposed a hybrid model (DBN and SVM) to predict French companies’ financial distress. However, the authors of these works have used small dataset to evaluate the performance of the proposed combinations of algorithms. Datasets considered for bankruptcy prediction datasets are not usually well balanced, as only few bankrupted companies are in the sample, it is necessary to rely on data balancing techniques like SMOTE.

we have used the dataset described in Zieba and Co. Data which has data of Polish companies that went bankrupt between 2007 and 2013, and from companies that continued to operate between 2000 and 2012, were used to create an extremely imbalanced dataset. In the same work, the Polish companies’ dataset was used to compare several classifiers’ performance with approaches that applies Extreme Gradient Boosting (EXGB) for learning an ensemble of decision trees, obtaining significant results with respect to the other methods they studied such as J48, RF, SVM and AdaBoost.

2.1 Data Profiling

Company bankruptcy was defined based on the business regulations of the Taiwan Stock Exchange. The Taiwan Stock Exchange is a financial institution located in Taipei, Taiwan. It has over 900 listed companies. It was established in 1961 and began operating as a stock exchange on 9 February 1962. The data were collected from the Taiwan Economic Journal for the years 1999 to 2009. The Taiwanese Bankruptcy Prediction data was obtained from UCI Machine Learning Repository.

The Taiwanese Bankruptcy Dataset contains:

* 95 features (X1-X95, business regulations of Taiwan Stock Exchange)
* 1 Vector of labels

Our aim is to use these features to understand their impact on the selected models and how they can help us recognizing the companies that are close to bankrupcty. All the variables that are part of this dataset are numeric and complete in nature. The data includes a majority of numerical attributes that help understand the possibility of bankruptcy. We will use various predictive models to see how accurately we predict which companies will face bankruptcy in the future. This is the same data set that was used by Liang D and Co[2] for research and donated to UCI community.

|  |  |
| --- | --- |
| X1 Cost of Interest-bearing Debt | X49 Fixed Assets Per Employee |
| X2 Cash Reinvestment Ratio | X50 total assets to GNP price |
| X3 Current Ratio | X51 Return On Total Assets(C) |
| X4 Acid Test | X52 Return On Total Assets(A) |
| X5 Interest Expenses/Total Revenue | X53 Return On Total Assets(B) |
| X6 Total Liability/Equity Ratio | X54 Gross Profit /Net Sales |
| X7 Liability/Total Assets | X55 Realized Gross Profit/Net Sales |
| X8 Interest-bearing Debt/Equity | X56 Operating Income /Net Sales |
| X9 Contingent Liability/Equity | X57 Pre-Tax Income/Net Sales |
| X10 Operating Income/Capital | X58 Net Income/Net Sales |
| X11 Pretax Income/Capital | X59 Net Non-operating Income Ratio |
| X12 Working Capital to Total Assets | X60 Net Income-Exclude Disposal Gain or Loss/Net Sales |
| X13 Quick Assets/Total assets | X61 EPS-Net Income |
| X14 Current Assets/Total Assets | X62 Pretax Income Per Share |
| X15 Cash/Total Assets | X63 Retained Earnings to Total Assets |
| X16 Quick Assets/Current Liability | X64 Total Income to Total Expenses |
| X17 Cash/Current Liability | X65 Total Expenses to Assets |
| X18 Current Liability to Assets | X66 Net Income to Total Assets |
| X19 Operating Funds to Liability | X67 Gross Profit to Sales |
| X20 Inventory/Working Capital | X68 Net Income to Stockholder's Equity |
| X21 Inventory/Current Liability | X69 One if Net Income is Negative for the Last Two Years; Zero Otherwise |
| X22 Current Liabilities/Liability | X70 (Inventory +Accounts Receivables) /Equity |
| X23 Working Capital/Equity | X71 Total Asset Turnover |
| X24 Current Liabilities/Equity | X72 Accounts Receivable Turnover |
| X25 Long-term Liability to Current Assets | X73 Days Receivable Outstanding |
| X26 Current Liability to Current Assets | X74 Inventory Turnover |
| X27 One if Total Liability exceeds Total Assets; | X75 Fixed Asset Turnover |
| X28 Equity to Liability | X76 Equity Turnover |
| X29 Equity/Total Assets | X77 Current Assets to Sales |
| X30 (Long-term Liability+Equity)/Fixed Assets | X78 Quick Assets to Sales |
| X31 Fixed Assets to Assets | X79 Working Capital to Sales |
| X32 Current Liability to Liability | X80 Cash to Sales |
| X33 Current Liability to Equity | X81 Cash Flow to Sales |
| X34 Equity to Long-term Liability | X82 No-credit Interval |
| X35 Liability to Equity | X83 Cash Flow from Operating/Current Liabilities |
| X36 Degree of Financial Leverage | X84 Cash Flow to Total Assets |
| X37 Interest Coverage Ratio | X85 Cash Flow to Liability |
| X38 Operating Expenses/Net Sales | X86 CFO to Assets |
| X39 (Research and Development Expenses)/Net Sales | X87 Cash Flow to Equity |
| X40 Effective Tax Rate | X88 Realized Gross Profit Growth Rate |
| X41 Book Value Per Share(B) | X89 Operating Income Growth |
| X42 Book Value Per Share(A) | X90 Net Income Growth |
| X43 Book Value Per Share(C) | X91 Continuing Operating Income after Tax Growth |
| X44 Cash Flow Per Share | X92 Net Income-Excluding Disposal Gain or Loss Growth |
| X45 Sales Per Share | X93 Total Asset Growth |
| X46 Operating Income Per Share | X94 Total Equity Growth |
| X47 Sales Per Employee | X95 Return on Total Asset Growth |
| X48 Operation Income Per Employee |  |

2.2 Implementation

2.2.1 Artificial Neural Network

ANN is a non-linear model similar to the biological neural network. An Artificial Neural Network is a computational network that resembles the original neurons of a human brain, hence ANN processing parts are called Artificial Neurons. The capabilities to learn, generalize the training data and derive results from complicated data is done by more interconnected neurons in ANN. For pattern, classification, trend identification, prediction, optimization problems, Artificial neural network is used. Without any programming, ANN will learn from the training data which has input and target output known. The capability to analyze information and answer the questions of a specific field is called an expert system that the learned neural network does. ANN processing elements are in the form of algorithms or hardware devices which is modeled after the human brain cerebral cortex neuronal structure. These networks are also called Neural Networks which is formed of many layers. The multiple layers that are interconnected called Multilayer Perceptron. Nodes refers to the neurons which are present in one layer. These nodes have an Activation function. The ANN has 3 main layers which are input layer, Hidden layer and Output layer.



Fig 1: Artificial Neural Network Architecture.

In the Input Layer, The input patterns are fed to the input layers. One or more hidden layers can be present in the ANN architecture. The processing that takes place in the inner layers is called hidden layers which calculates the output based on the weights which is the sum of weighted synapse connections. By removing the redundant information, the input is redefined by the hidden layers and send the information to the next hidden layer for further processing. This hidden layer connects to the output layer where the output is shown.

2.2.2 Long Short Term Memory

LSTM solve the Vanishing Exploding gradients problem encountered during the operation of a basic Recurrent Neural Network. The Layers that get a small gradient update stops learning in RNN which are the earlier layers. RNN can forget the longer sequences, therefore having a short-term memory. LSTM’s were created as the result to short-term memory. LSTM is a type of Recurrent Neural Network developed for the handling sequential prediction problems like Product Recommendation, Weather Forecasting, Text Translation, Stock Market prediction, etc. LSTM have gates that can regulate the flow of information that is which data in a sequence is important to keep to make predictions or throw away the non relevant data. Passing the relevant information to the long chain of sequences to make predictions. LSTM network comprises of different memory blocks called cells. There are two states which is the cell state and the hidden state.

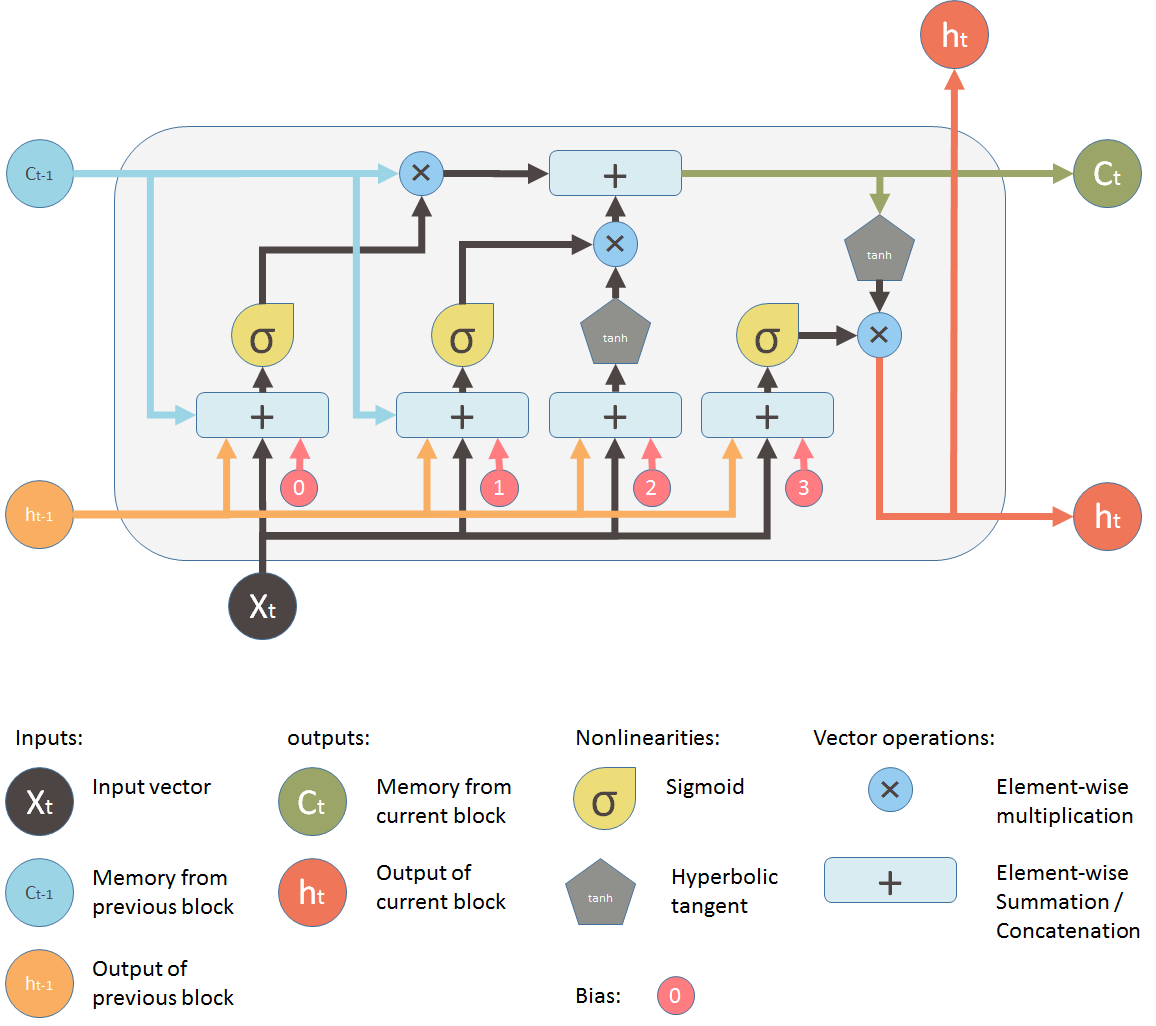


Fig 2: LSTM ARCHITECTURE.

Forget gate decides what information should be relevant or irrelevant. Through the sigmoid function, the previous hidden state information and the current input information is passed. Values lies between 0 which means forget and 1 means to keep. Input Gate performs the operations to update the cell status. The previous hidden state and current input is passed to a sigmoid function which decides what values will be updated by transforming the values to be between 0 and 1. The hidden state and current input is passed to the tanh function to squish values between -1 and 1 for the regulation of the network. Then tanh output is multiplied with the sigmoid output. Which information is important to keep from the tanh output is decided by the sigmoid output. The cell state gets pointwise multiplied by the forget vector which has a possibility of dropping values in the cell state if it gets multiplied by values near 0. The output from the input gate is taken and pointwise addition is done which updates the cell state to new values that the neural network finds relevant to get the new cell state. What the next hidden state should be is decided by the Output gate. The previous inputs information are hold by the hidden state which is used for predictions. The output is the hidden state. To the next time step, the new cell state and the new hidden state is then carried over.

2.2.3 DenseNet

DENSENET refers to Densely Connected Convolution Networks. DenseNet is composed of **Dense blocks** in which the densely connected layers are present. In Densenet, there will be connections from all subsequent layers in a feed-forward fashion. The feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. DenseNets require fewer parameters than traditional CNN, as there is no need to learn redundant feature maps. Very narrow densenet layers are presentand they just add a small set of new feature-maps. DenseNets are splitted into **DenseBlocks and transition layers. The feature maps dimensions remains same within a block in denseblock, but the number of filters changes between them**. These layers are called Transition Layers.

#### A group of layers connected to all their previous layers is Dense block. A single layer looks like Batch Normalization, ReLU activation, 3x3 Convolution. In Transition layer, DenseNet concatenates all the feature maps. Concatenating the feature maps of different sizes would be impracticable. Thus, the feature maps of each layer has the same size in each dense block. However down-sampling is essential to CNN. A transition layer in dense network is made of Batch Normalization, 1x1 Convolution, Average pooling.

2.2.4 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is an ANN which uses sequential data or time series data. The training data is utilized to learn in RNN. Their memory is distinguished as they take prior inputs information to influence the current input and output. Depending on the prior elements within the sequence, the output of RNN is decided. RNNs run into two problems which are exploding gradients and vanishing gradients. These issues are defined by the size of the gradient. When there is a too small gradient, it continues to become smaller, until they become insignificant the weight parameters are updated. So that the algorithm will no longer learn. When there is too large gradient, Exploding gradients will occur which creates an unstable model. In this, The model weights will grow too large. The solution to these issues are the number of hidden layers within the neural network is reduced, eliminating some of the complexity in the RNN model.

2.3 Validation:

Several metrics are often used to measure the performance of any classifier, computed by combining the results obtained within the confusion matrix. Four categories are composing this matrix:  
1) True Positives (TP): number of samples correctly classified as bankrupt.  
2) False Positives (FP): number of samples incorrectly classified as bankrupt.  
3) True Negatives (TN): number of samples correctly classified as solvent.  
4) False Negatives (FN): number of samples incorrectly classified as solvent.

Thus, since the binary classification accuracy results aren't reliable while the info considered is extremely imbalanced (the classifiers always tend to predict the majority class and ignore the minority class), several metrics are computed to form a far better judgment about each classifier’s performance and reliability. These metrics are:  
Accuracy: Performance of the classifier in terms of assigning the right class to every instance.

Accuracy =(TP + TN)/(TP + TN + FP + FN)  
Type I error : Also called as False Positive Rate (FPR). It represents the failure of the classifier to assign bankrupt companies to the ‘bankrupt’ class (wrong prediction), while its actual class is ‘bankrupt’ (real status).   
Type 1 error = FP/(TN+FP)   
Type II error : Also called as False Negative Rate (FNR), represents the failure of the classifier in assigning solvent companies to ‘solvent’ class (wrong prediction), while its actual class is ‘solvent’ (real status).   
Type II Error = FN/(TP+FN) = 1 – Recall

3 Experiments and Results:

We are comparing the result generated by Liang D & Co [2] with our results as the underlying data set remains the same. FC is combination of Financial Ratio’s and Corporate Governance Indicators and he compared performance of the model with FC and Financial Ratio’s. For this experiment we are using LSTM + Densenet, GRU and ANN for evaluation of result.

LSTM and DenseNet architecture has been used to predict the Bankruptcy

Table

Description automatically generated

Fig 3: LSTM with DenseNet model

Training and Test Loss for the LSTM and DenseNet model is

A picture containing graphical user interface

Description automatically generated

Fig 4: Training Vs Test Loss for LSTM and Dense Net model

ANN , multilayer perceptron architecture is below

Text

Description automatically generated

Fig 5: ANN model

ANN model, training and test loss graph is below

Chart, line chart

Description automatically generated

Fig 6: Training Vs Test loss for ANN model

RNN – Architecture to solve the problem of Bankruptcy is below

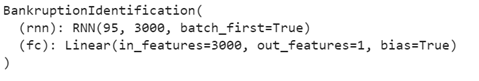


Fig 7: RNN model

Training and Test Loss Graph for GRU is below

Chart, line chart

Description automatically generated

Fig7: Training Vs Test Loss RNN

Chart, line chart

Description automatically generated

Fig 8: ROC curve for LSTM with DenseNet

A picture containing line chart

Description automatically generated

Fig9: Precision Recall Curve for LSTM+DenseNet

Table 2: Results

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Accuracy | Type I Error | Type 2 Error |
| Support Vector Machine -SLR | 81.3 | 16.7 | 20.8 |
| K- Nearest Neighbor -SLR | 77.9 | 20.1 | 24.1 |
| CART – SLR | 78.2 | 22.2 | 21.4 |
| Multi Layer Perceptron – SLR | 80.5 | 18.8 | 20.3 |
| Naïve Bayes – SLR | 77.1 | 12.00 | 33.80 |
| **Artificial Neural Network** | **97.0** | **3.1** | **0** |
| **LSTM+DenseNet** | **99.9** | **0** | **0** |
| **RNN** | **97.0** | **3.1** | **0** |

Comparing various algorithms with or with out sampling techniques LSTM+DenseNet performance is good. All the predictions for LSTM+DenseNet are done with 4 epochs based on training and test loss validation.

4 References:

1. E. Fedorova, E. Gilenko, and S. Dovzhenko, ‘‘Bankruptcy prediction for Russian companies: Application of combined classifiers,’’ Expert Syst. Appl., vol. 40, no. 18, pp. 7285–7293, Dec. 2013.
2. H. Jawazneh, A. Mora, and P. Castillo, ‘‘Predicting the financial status of companies using data balancing and classification methods,’’ in Proc. Int. Work-Conf. Time Ser. (ITISE). Granada, Spain: Godel Impresiones Digitales SL, 2017, pp. 661–673.
3. T. Le, M. Lee, J. Park, and S. Baik, ‘‘Oversampling techniques for bankruptcy prediction: Novel features from a transaction dataset,’’ Symmetry, vol. 10, no. 4, p. 79, Mar. 2018.
4. F. J. L. Iturriaga and I. P. Sanz, ‘‘Bankruptcy visualization and prediction using neural networks: A study of U.S. Commercial banks,’’ Expert Syst. Appl., vol. 42, no. 6, pp. 2857–2869, Apr. 2015.
5. J. Brozyna, G. Mentel, and T. Pisula, ‘‘Statistical methods of the bankruptcy prediction in the logistics sector in Poland and Slovakia,’’ Transformations Bus. Econ., vol. 15, no. 1, pp. 80–96, 2016.
6. S. S. Devi and Y. Radhika, ‘‘A survey on machine learning and statistical techniques in bankruptcy prediction,’’ Int. J. Mach. Learn. Comput., vol. 8, no. 2, pp. 133–139, Apr. 2018.
7. M. Zięba, S. K. Tomczak, and J. M. Tomczak, ‘‘Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction,’’ Expert Syst. Appl., vol. 58, pp. 93–101, Oct. 2016.
8. J. Min and Y. Lee, ‘‘Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters,’’ Expert Syst. Appl., vol. 28, no. 4, pp. 603–614, May 2005.
9. P. P. M. Pompe and A. J. Feelders, ‘‘Using machine learning, neural networks, and statistics to predict corporate bankruptcy,’’ Comput.-Aided Civil Infrastruct. Eng., vol. 12, no. 4, pp. 267–276, Jul. 1997.
10. J. Brozyna, G. Mentel, and T. Pisula, ‘‘Statistical methods of the bankruptcy prediction in the logistics sector in Poland and Slovakia,’’ Transformations Bus. Econ., vol. 15, no. 1, pp. 80–96, 2016.
11. E. I. Altman, ‘‘Financial ratios, discriminant analysis and the prediction of corporate bankruptcy,’’ J. Finance, vol. 23, no. 4, pp. 589–609, Sep. 1968.
12. J. A. Ohlson, ‘‘Financial ratios and the probabilistic prediction of bankruptcy,’’ J. Accounting Res., vol. 18, no. 1, pp. 109–131, Apr. 1980.
13. [Liang, D., Lu, C.-C., Tsai, C.-F., and Shih, G.-A. (2016) Financial Ratios and Corporate Governance Indicators in Bankruptcy Prediction: A Comprehensive Study. European Journal of Operational Research, vol. 252, no. 2, pp. 561-572.](https://isslab.csie.ncu.edu.tw/download/publications/1.pdf)