JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 1209-1230 (2023)
DOI: 10.6688/JISE.202309_39(5).0012

Self-Attention Recurrent Conditional Generative Adversarial Networks for Corporate Credit Rating Prediction*

SHU-YING LIN[1,+] AND AN-CHI WANG[2]
[1]*Department of Finance*
Minghsin University of Science and Technology
Xinfeng, 304 Taiwan
E-mail: sylin@must.edu.tw[+]
[2]*MediaTek Incorporated*
Hsinchu Science Park, Hsinchu, 300 Taiwan

Financial risk management has always been a critical issue; banks, debt issuers, and government officials all need credit ratings in order to make intelligent financial decisions. Most of the existing studies on corporate credit rating prediction utilize financial statement features as their input data. Credit rating is closely related to credit risk. However, very few studies consider credit risk elements, such as credit systematic risk / beta and Credit Default Swap (CDS) spread data in credit rating prediction. Furthermore, the application of generative adversarial learning for corporate credit rating prediction was rarely investigated. In this work, a novel generative adversarial network (GAN), Self-Attention Recurrent Conditional GAN (SAR-CGAN) for corporate credit rating prediction is proposed. The proposed model takes advantage of Conditional GAN and Recurrent GAN to improve prediction performance. The financial statement features and corporates' CDS spread-related features: credit systematic risk / beta and quarter mean of CDS spread are used as input features. The proposed model adopts long short-term memory networks (LSTM) based on self-attention to process historical data and generate corporate credit rating. We improve the recurrent-based GAN model by modifying the network structure, in which the self-multi-head attention layer is added to capture the weighted importance of the time series data. Moreover, a data sampling strategy is designed to alleviate the overfitting issue and enhance the effectiveness of the proposed GAN model. The experimental results indicate that the proposed model performs better than other state-of-art models on the applied datasets.

Keywords: corporate credit rating, self-attention mechanism, LSTM, credit systematic risk, generative adversarial network

1. INTRODUCTION

Credit risk assessment [1-4] is one of the most critical issues in financial risk management. Financial Institutions conduct credit ratings on companies to reduce default risks and systemic risks. Credit ratings are not only used for financial instruments of banks, but also as risk management tools. Bond investors, debt issuers and regulators all use credit ratings. Debt issuers use credit ratings to measure corporate risk, which represents the debtor's credit risk level and predicts its ability to meet its payable obligations, so that the company issuing bonds can estimate the return that investors may need. In addition, banks and regulators often rely on credit ratings to make sound financial decisions. In general,

Received October 13, 2022; revised November 3, 2022; accepted November 7, 2022.
Communicated by Churn-Jung Liau.
[+] Corresponding author.
[*] This research was supported in part by the National Science and Technology Council of Taiwan under Grant No. MOST 111-2637-H-159-001.

credit ratings are published by credit rating agencies, for instance, Moody's, Fitch Ratings, and Standard & Poor's (S&P). However, these agencies charge quite high fees for their services since human resources are required in the process of evaluating credit ratings. In addition, the process usually takes time and may not immediately reflect the real condition of the company.

In the past few years, several machine learning methods [5] have been implemented into corporate credit rating research, for instance, support vector machines (SVM) [6, 7], decision trees (DT) [4, 8], and so on. By applying these machine learning methods, the shortcomings of traditional credit rating scoring that relies on manual decision-making can be eliminated. More notably, machine learning methods help to achieve high accuracy and efficiency in binary classification tasks. Although these methods have the above-mentioned advantages, their performance is still less than expected when dealing with time series problems or multi-category classification. Deep learning methods based on neural networks [9-11], such as back-propagation neural network (BPNN) [12], can effectively deal with high-dimensional, nonlinear data. However, the ability of multi-layer perceptron and back-propagation neural networks tackling with multi-class classification problems is less outstanding than that of binary classification.

B. Chen, *et al*. [13] proposed a GRU-based architecture, which is a Self-Multi-head Attention-based Gated Recurrent Unit model (SMAGRU) that captures the weights assigned to the market benchmark. Time series characteristics are used to evaluate the credit rating of enterprises. Experimental comparisons with statistical and machine learning benchmark models on distinct data sets indicate that the SMAGRU model has advantages in predicting credit rating. The relevant research showed that the temporal financial statement features are critical, and also that the multi-head self-attention mechanism greatly enhances the temporal feature characteristic in addition to improving the rating performance.

In the above-mentioned existing methods, the financial statement features are normally used as input features; however, few of them consider credit risk elements, such as Credit Default Swap (CDS) spread data [14] and credit systematic risk β [15] as input features. Furthermore, we found that the application of generative adversarial learning for corporate credit rating prediction was rarely discussed. As a result, we further added the CDS spread data and credit systematic risk β as input features and applied generative adversarial learning for corporate credit rating prediction.

In our research, we utilized companies' quarterly financial statements, quarter mean of their CDS spread, and the credit systematic risk β as input features. Credit systematic risk β calculated with historical CDS spread data is used to measure the size of the systemic risk of a single asset relative to the risk of the market. Adding this indicator improves the performance of credit rating prediction.

The core concept of Generative Adversarial Networks (GAN) [16] is learning through the confrontation between the generator and the discriminator; the generator produces samples close to the real data to confuse the discriminator. The goal of the discriminator is to discriminate between the outputs of the generator and the real samples. The two networks compete with each other and constantly adjust their parameters to render the discriminative network incapable of discriminating whether or not the output of the generator is true. One of the variants of GAN is Conditional Generative Adversarial Network (CGAN) [17], which was proposed to generate pictures under certain conditions. Moreover, Recurrent

Generative Adversarial Networks (RecGAN) [18] utilize GAN and Recurrent Neural Networks (RNNs): Gated Recurrent Units (GRUs) to effectively model the latent and temporal features.

In this work, we propose a novel Self-multi-head Attention-based Recurrent Conditional GAN (SAR-CGAN) classification model to solve the corporate credit rating problem. The financial statement features and corporates' CDS spread-related features: credit systematic risk β and quarter mean of CDS spread were used as input features. We leverage the advantage of CGAN and allow the conditioned inputs of the discriminator to increase their learning effectiveness. We also utilize the temporal feature capturing capability of RecGAN. Besides the financial statement features, we additionally use the corporates' CDS Spread data to calculate the corporates' credit systematic risk β, and take it as one of the features. The proposed model adopts long short-term memory networks (LSTM) based on self-attention to process historical financial statement data and generate corporate credit rating. As for the discriminator, it is also comprised of long short-term memory network, and its input corporate rating is conditioned with the corporate's financial statement features. We improved the Recurrent-based CGAN classification model by modifying the network structure, in which the self-multi-head attention layer was added in the generator to improve the performance. Moreover, we also designed a data sampling strategy to alleviate the overfitting problem and enhance the effectiveness of the proposed GAN model. The experimental results demonstrate that the proposed model performs better than other state-of-art models.

The rest of this paper is arranged as follows. In Section 2, we introduce related work on credit default swaps, text mining, deep learning and GAN architectures. In Section 3, the details of the proposed model are presented. The experiment and evaluation are given in Section 4. The conclusions and future work suggestions are presented in Section 5.

2. RELATED WORK

We first introduce the related work on corporate credit rating prediction. Next, we discuss how credit default swap is used in corporate risk management assessment and why we use systemic risk β as our research material. Secondly, the self-attention mechanism is explained. Thirdly, long short-term memory neural networks are presented. Finally, fundamental concepts of generative adversarial network and conditional generative network models are discussed.

2.1 Corporate Credit Rating Prediction

There are a variety of classification methods applied to corporate credit rating prediction problems. SVM is one of the methods widely utilized for corporate credit rating classification. In [6], a ordinal multi-class support vector machine (OMSVM) is proposed as an extension of binary SVMs, using ordinal pairwise partitioning, for tackling multiple-ordinal-classes classification problems. The proposed methods based on SVM in [6, 7, 19] show the greater efficiency compared to manual decision-making. Moreover, various deep learning methods are proposed to deal with high-dimensional and non-linear data problems. In [12], the comparison of the performance of various deep learning methods such as convolutional neural networks (CNN), CNN2d, LSTM, GRU shows that LSTM performs best

when dealing with temporal data. Chen *et al.* [13] proposed a GRU-based architecture, SMAGRU, which is a self-multi-head attention-based gated recurrent unit model. Attention-weighted time series characteristics are used to evaluate the credit rating of an enterprise. From the research, it was found that the temporal financial statement features are critical and also that a multi-head self-attention mechanism greatly enhances the temporal feature characteristics and improves the performance.

2.2 Systematic Risk β

Financial risk management is important in making intelligent investment decisions [20]. We can analyze and spread the unsystematic part of total risk based on Markowitz portfolio theory [21]. Hence, systematic risk is the most critical part of the investor's risk measured by beta. Systematic risk β in finance [15] is a measure of how an individual asset moves on average when the overall market increases or decreases. The systematic risk β of company c is calculated as follows:

$$\beta_c = \frac{Covr(r_c, r_m)}{Var(r_m)}. \tag{1}$$

r_c refers to the rate of return of company c, and r_m refers to the rate of return of the whole market. If $\beta_c > 1$, then the fluctuation of r_c is higher than the market; in other words, the rate of return of company c is riskier than that of the market, and vice versa.

In [22], the relationships among credit rating, beta, and systematic risk are discussed, indicating that a higher rating should specify a lower credit risk and therefore lower beta. The relationship between credit rating changes and beta has been examined in studies; they found that systemic risks increase after downgrading.

2.3 Attention Mechanism

Attention mechanism [23] is proposed to reduce the computational cost by focusing on important parts of the observed aspects [24], rather than the whole picture. Attention mechanisms can assign weights to every input information, and then extract key information to enhance the performance. The attention mechanisms are generally used in machine translation, speech recognition, abstract generation, and image annotation. Hu *et al.* [25] proposed a hierarchical attention network (HAN) to assemble a word and sentence attention-based document representation.

Self-attention mechanism [23] is one of the attention mechanisms proposed for *Transformer* architecture, and appeared quite effective in many seq2seq models such as [26, 27]. Self-attention empowers the model to attend to certain pieces of useful information of the sequence and therefore improves sequence learning.

2.4 Long Short-Term Memory Neural Network

The long-short term memory neural network (LSTM) is an application of recurrent neural networks [28]. LSTM is a superior variant of recurrent neural networks since it is capable of learning long-term sequence data and is less affected by vanishing gradient [29]. LSTM consists of four gates and a unique cell stage: input gate, output gate, forget gate, and memory cell, employed to forget information and add new information to update

the next state.

Corporate credit scores do not fluctuate much from quarter to quarter; therefore, relying on LSTM's advantages to capture the changes in credit rating helps to predict the credit rating more accurately than treating each quarter as an independent observation. In this study, we implement LSTM into our generator to capture the historical changes of the corporate.

2.5 Classification based on Generative Adversarial Networks

The core concept of generative adversarial networks (GAN) [16] is learning through the confrontation between the generator *G* and discriminator *D*. Generative model *G* aims to generate data which approximate the true data, and the discriminator *D* continuously learns to enhance the ability to differentiate the real data from the generated data. The value function of two models: $V(G, D)$, is shown in Eq. (2). Through the confrontation, the generator will eventually become capable of producing samples that are close to real data.

$$\min_G \max_D V(D, g) = E_{x \in p_{data}(x)}[\log D(x)] + E_{z \in p_z(z)}[\log(1 - D(G(z)))] \tag{2}$$

where P_{data} is the true distribution of data *x*, $P_z(z)$ denotes prior noise distribution, and $D(x)$ indicates the probability that *x* is from the true data.

GAN has gained success in computer vision, for instance, image generation [30, 31] and style transfer [32]. Researchers have since seen that there is still room for research on GAN, and have developed innovated architectures based on GAN. One of the variants of GAN is Conditional generative adversarial network (CGAN) [17] which was proposed to generate pictures under certain conditions. Both the generative and discriminative models are conditioned on extra information *y*; *y* is fed into both the discriminative model and generative model as additional input. In the generative model, the prior input noise distribution $P_z(z)$ and *y* are combined into joint hidden representation. The input of the discriminative model are the generated pictures conditioned with *y*, $G(z|y)$, and the real samples, *x* with *y*. The objective function is shown as Eq. (3),

$$\min_G \max_D V(D, G) = E_{x \in p_{data}(x)}[\log D(x \mid y)] + E_{z \in p_z(z)}[\log(1 - D(G(z \mid y)))]. \tag{3}$$

To take advantage of the Recurrent Neural Network (RNN) and capture the dynamic behaviors of users and items, a recurrent generative adversarial network (RecGAN) [18] is proposed to solve food and movie recommendation problems. In RecGAN, both the generator and discriminator are comprised of the RNN-based model: Gated Recurrent Unit (GRU). The generator is responsible for generating the ratings, while the discriminative model is assigned to discriminate the generated ratings from the true distribution. The minmax game for RecGAN is formulated as Eq. (4):

$$Q^* = \min_G \max_D \sum_{t=1} \sum_{i=1} \sum_{j=1} (E_{r \in D(r|i,j)_{real|t}}[\log D(r \mid i, j, t)] + E_{r \in D(r|i,j)gen|t}[\log(1 - D(r \mid i, j, t))]). \tag{4}$$

$D(r|i, j, t)$ denotes the discriminator's probability prediction of $(r|i, j, t)$ being sampled from the real distribution $D(r|i, j)_{real|t}$. There are *N* users, *M* items, and *T* timestamps.

3. PROPOSED APPROACH

In this section, we introduce the proposed model, SAR-CGAN, a Self-multi-head Attention-based Recurrent Conditional GAN model for credit rating predictions. Firstly, the main concept of SAR-CGAN is explained in Section 3.1. Secondly, the methods of feature extraction are illustrated in Section 3.2. In Section 3.3, the procedure of adversarial training is discussed. Next, Section 3.4 describes the proposed data sampling strategy to amplify the learning effectiveness of the proposed model.

3.1 Overview

The structure of our proposed model (SAR-CGAN) is shown in Fig. 1. In the input layer, we collect historical corporate financial statement data, credit default swap spread data, and corporate credit rating data. The collected data are then preprocessed to generate time series data. The numerical time series data include financial statement features and CDS-based credit systematic risk β. The credit rating data are transformed to one-hot-encoded form. In the adversarial training phase, financial statement features, CDS spread, and CDS-based credit systematic risk β time series data are sent into the generator, and the generator outputs the corporate credit rating prediction. The credit rating prediction conditioned with its generator input time series data is then defined as Fake data, while the actual credit rating with its time series data is defined as Real data. Next, Real and Fake data are then concatenated and set as the input of the discriminator. The generator and discriminator compete and optimize each other through iterative training in order to make the generator equipped with the ability to produce high quality prediction results.

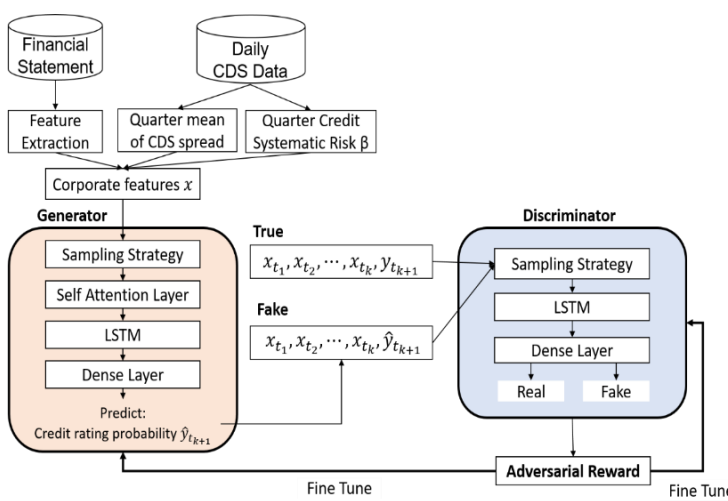


Fig. 1. Overview of the proposed model.

The proposed model leverages the advantage of CGAN and RecGAN to improve the prediction results. The proposed model adopts self-attention LSTM to process time series data. Moreover, we also designed a data sampling strategy for the generator and the discriminator to alleviate the overfitting issue and enhance the effectiveness of GAN model.

3.2 Feature Extraction

3.2.1 Financial statement data

Several existing methods for predicting corporate credit rating [4, 7, 11, 12, 33], utilize financial ratios which are calculated by the elements in financial statement, such as profitability, management efficiency, liquidity, *etc.* According to the research [34], using financial ratios may result in feature cross effects. Therefore, inspired by [13], we determined to directly utilize the elements of financial statements, as listed in Table 1. The financial statement features are comprised of a corporate's financial records and financial structure that are known to affect corporate credit ratings. We obtained the quarterly corporate financial statement data from the website, Wharton Research Data Service (WRDS). After we selected the financial statement features, we first fill in the missing value and conduct min-max normalization.

Table 1. Financial statement features.

1	Total Revenue	15	Total Current Liabilities
2	Sales/Turnover (Net)	16	Total Liabilities
3	Net Income	17	Cash
4	Total Inventories	18	Debt in Current Liabilities
5	Total Operating Expense	19	Total Long-Term Debt
6	Total Income Taxes	20	Financing Activities(Net Cash Flow)
7	Selling, General and Administrative Expenses	21	Retained Earnings
8	Sale of PP&E and Investments	22	Dividends per Share
9	Total Receivables	23	Total Stockholders Equity
10	Investing Activities (Net Cash Flow)	24	Total Interest and Related Expense
11	Cash and Cash Equivalents	25	Total Market Value
12	Inventory	26	Comprehensive Income
13	Total Current Assets	27	Operating Activities(Net Cash Flow)
14	Total Assets		

3.2.2 CDS spread data and credit systematic risk β

In this study, credit systematic risk β is leveraged as the index of corporate credit risk. Considering that CDS spread is highly related to credit risk assessment, we decided to use the credit default swap spread to calculate credit systematic risk β, as shown in Eq. (5).

$$\beta_{i,t} = \frac{Cov(CDS_{c,t}, CDS_{m,t})}{Var(CDS_{m,t})}; CDS_{m,t} = \frac{\sum_{c=1}^{C} CDS_{c,t}}{C} \tag{5}$$

$CDS_{c,t}$ refers to the daily CDS spreads of company c within time interval t; $CDS_{m,t}$ denotes the daily CDS spreads of the whole market within time interval t; there are total C companies. The time interval is the time span from the last financial statement released date to the current financial statement released date (a quarter).

The companies' Credit Default Swap spread data were crawled from Datagrapple.com. The data are updated daily, excluding non-trading days. We chose all the companies

in the North America list on the website, totaling 113 companies' daily CDS spread data crawled.

3.2.3 S&P's corporate credit rating data

S&P's corporate credit rating data were collected from the Compustat database of Wharton Research Data Services, WRDS. There are 22 classes (from AAA to SD) in Standard & Poor's (S&P) corporate credit rating. To implement credit ratings to the multiclass classification, we first transferred the credit ratings into numbers 0-21, as shown in Table 2. To be clear, the big or small numbers mean nothing but represent the class number. Since the ratings under CCC+ are sparse, we put the ratings under CCC+ all into class 0. Finally, there were 17 classes in total. We then utilized one-hot encoding to transform the class numbers into vectors for implementation purpose.

Table 2. S&P credit rating.

Class No.	STANDARD & POOR's	Class No.	STANDARD & POOR's
16	AAA	8	BBB
15	AA+	7	BBB–
14	AA	6	BB+
13	AA–	5	BB
12	A+	4	BB–
11	A	3	B+
10	A–	2	B
9	BBB+	1	B–
		0	CCC+, CCC, CCC–, CC, D, SD

3.3 Generative Adversarial Training

3.3.1 Problem setting

The main purpose of our research is to predict corporate credit rating by leveraging previous k quarters' financial statement features, mean of CDS spread, and credit systematic risk β data. Let t_{k+1} denote the predicted quarter; the previous k quarters are denoted as $t_1, t_2, \ldots, t_k$. The time series data on the extracted features are demonstrated as follows:

- Financial Statement features: $[f_{t_1}, f_{t_2}, \ldots, f_{t_k}]$, f_{t_l} is a set of 27 financial statement features at time t_l.
- Mean of CDS spread: $[cds_{t_1}, cds_{t_2}, \ldots, cds_{t_k}]$, cds_{t_l} is a set of mean of CDS spread data each quarter.
- Credit Systematic Risk β: $[\beta_{t_1}, \beta_{t_2}, \ldots, \beta_{t_k}]$, β_{t_l} is credit systematic risk β on financial statement released date t_l.

The features above were then merged and utilized as the inputs of generator and conditions of discriminator:

$$x = x_{t_1}, x_{t_2}, \ldots, x_{t_k} = [f_{t_1}, cds_{t_1}, \beta_{t_1}], [f_{t_2}, cds_{t_2}, \beta_{t_2}], \ldots, [f_{t_k}, cds_{t_k}, \beta_{t_k}].$$

Finally, we let $\hat{y}_{t_{k+1}}$ denote the predicted credit value at quarter t_{k+1}.

3.3.2 The generative model

The generator in the proposed model was designed as the classifier of the corporate credit rating. Its outputs are the predicted probabilities of the one-hot-encoded 17-class-vector. The generator's inputs are the corporate features x, which are the combinations of financial statement element and CDS spread, and corporate credit systematic risk β. The overview of the generator is shown in Fig. 2.

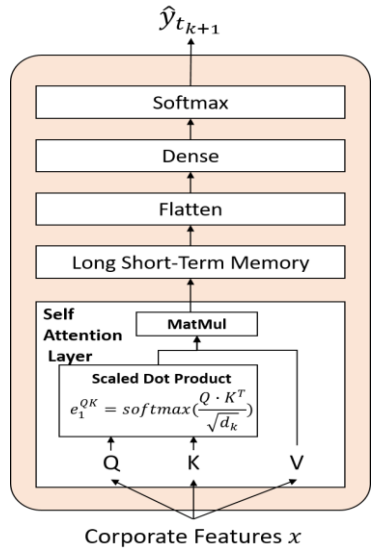


Fig. 2. Overview of the generator.

Fig. 2 illustrates the details of the generator architecture; the corporate feature inputs were first sent into the self-attention layer to capture the self-attention weighted vector of the input features. The attention weighted features were then processed by LSTM in order to deal with the temporal characteristics. Next, the processed data were flattened and assigned to the dense layer to generate the credit rating prediction. In the following section, we demonstrate the components of our proposed generator.

3.3.2.1 Self-attention layer

Regarding the purpose of capturing the relationship between the features in the temporal data and further focus on certain features, we first allocate the input data to the Self-Attention layer. Subsequently, the attentional temporal data are fed into the LSTM to further capture the temporal information. In this study, we leverage an attention mechanism proposed by [23], which is called "Self-Attention." The overview of the Self-Attention mechanism is shown in Fig. 3.

The input of the generator, x is comprised of temporal data of financial statement features f, quarter mean of CDS spread cds, and credit systematic risk β. In the beginning of Self-Attention, the input x is duplicated to multiply with the weight matrices W^Q, W^K, W^V and the representations of the three linear outputs are query Q, key K, and value V, re-

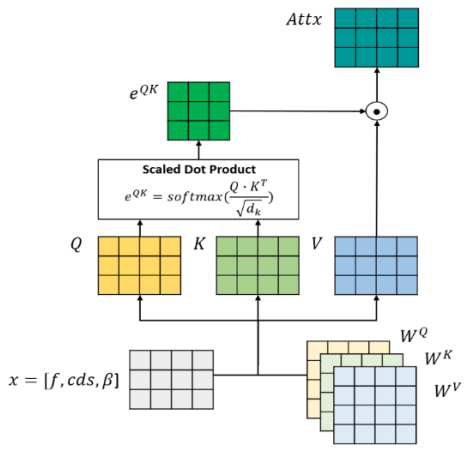


Fig. 3. Self-attention mechanism.

spectively. The query Q represents the encoded input vector, key K represents the vector that will be multiplied with Q to get attention matrix, and value V represent the encoded input vector that the attention information will be multiplied with. In order to capture the similarity scores of the features, Q is dot product by K^T. Furthermore, to scale down the similarity scores, $Q \cdot K^T$ is divided by $\sqrt{d_k}$; d_k denotes the dimension of query Q, key K, and value V. Finally, we use *Softmax* to transform the scores into probabilities. The Self-Attention mechanism is computed in Eqs. (6)-(8):

$$e^{QK} = \frac{Q \cdot K^T}{\sqrt{d_k}}, \tag{6}$$

$$e^{QK} = Soft\max(\frac{Q \cdot K^T}{\sqrt{d_k}}), \tag{7}$$

$$Att = e^{QK} \cdot V. \tag{8}$$

As Eq. (8) shows, the Self-Attention layer dot product V with the probability e^{QK} outputs attentional data *Attx*.

The attentional corporate data *Attx* were computed by the Self-Attention layer, as shown in Eq. (9):

$$Attx = [Attx_{t_1}, Attx_{t_2}, \ldots, Attx_{t_k}] = SelfAttention(x_{t_1}, x_{t_2}, \ldots, x_{t_k}). \tag{9}$$

3.3.2.2 LSTM layer and fully connected layers for prediction

In order to obtain the fluctuation of historical corporate credit rating, we then apply LSTM to extract $X^L_{t_{l+1}}$, the temporal LSTM latent vector of the attentional corporate data, as shown in Eq. (10):

$$X^L_{t_{l+1}} = LSTM(Attx), \text{ where } l \in [1, k] \tag{10}$$

where $Attx = [Attx_{t_1}, Attx_{t_2}, \ldots, Attx_{t_k}]$ denotes the attentional corporate data, including financial statement features, mean of CDS spread, and credit systematic risk β data, from t_1 to t_k. l denotes each time period from 1 to k. The attentional corporate data $Attx$ is fed into the LSTM layer to derive $X^L_{t_{l+1}}$ at each time period from the LSTM cell.

The output of LSTM, $X^L_{t_{l+1}}$, is flattened and fed into dense layers to generate a credit rating prediction, as shown in Eqs. (11)-(12). The flattened layer is used to convert the 3-dimensional data into the input vector of linear layers. Let $X^F_{Flatten}$ denote the vectors after flattening $X^L_{t_{l+1}}$:

$$X^F_{Flatten} = Flatten(X^L_{t_{l+1}}), \text{ where } l \in [1, k], \tag{11}$$

$$y'_{t_{k+1}} = W_d \cdot X^F_{Flatten} + b_d, \tag{12}$$

where W_d and b_d are the weight and bias of the dense layer, respectively. Next, we apply a *Softmax* layer at the end of generator to adapt the output vector y' to the probability of the credit rating prediction, $\hat{y}$, as shown in Eq. (13):

$$\hat{y}_{t_{k+1}} = Softmax(y'_{t_{k+1}}). \tag{13}$$

3.3.3 The discriminative model

The discriminative model in this study is inspired by the recurrent generative adversarial network model [18]. A discriminator is normally utilized to evaluate whether the prediction of the generator is similar enough to the original real value. The evaluation is then leveraged as a reward for the generator. The real data $y_{t_{k+1}}$ and fake data $\hat{y}_{t_{k+1}}$, as shown in Eq. (14), are the real credit rating and the prediction result of generator G, respectively:

$$\hat{y}_{t_{k+1}} = G([x_{t_1}, x_{t_2}, \ldots, x_{t_k}]), \tag{14}$$

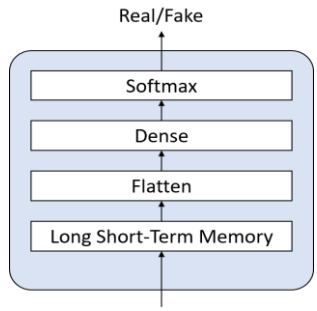
where G denotes the generative model and k of t_k is set to be 6, that is, we feed the previous six quarters of corporate features into generator G to predict the credit rating of the next quarter t_{k+1}. X_{t_k} refers to the combination of financial statement features f_{t_k}, quarter mean of CDS spread data cds_{t_k}, and credit systematic risk β_{t_k} at time t_k.

Furthermore, inspired by the existing conditional GAN model [17], the discriminator learns better under certain conditions. Therefore, in our proposed method, both the real data and fake data are conditioned with the corporate features x_{t_k}. The proposed architecture of the discriminator is shown in Fig. 4. The Real input data X_{real} and Fake input data X_{fake} of the discriminator are formed as Eqs. (15) and (16), respectively:

$$X_{real}: [x_{t_1}, x_{t_2}, \ldots, x_{t_k}, y_{t_{k+1}}], \tag{15}$$

$$X_{fake}: [x_{t_1}, x_{t_2}, \ldots, x_{t_k}, \hat{y}_{t_{k+1}}]. \tag{16}$$

The discriminator, which is comprised of the LSTM model and Dense layer, takes samples from the Real input data and Fake input data; its aim is to learn to distinguish them.



X_{real}: $[x_{t_1}, x_{t_2}, \ldots, x_{t_k}, y_{t_{k+1}}]$

X_{fake}: $[x_{t_1}, x_{t_2}, \ldots, x_{t_k}, \hat{y}_{t_{k+1}}]$

Fig. 4. Overview of the proposed discriminator.

3.3.4 Generative adversarial training

In the generative adversarial network, generator G is dedicated to generating the predicted value close enough to the real value and discriminator D is committed to differentiating the generated value from the real value. In the model, generative adversarial learning is used to let G and D compete with each other. The value function $V(D, G)$ of the minimax game is formulated as Eq. (17). $D(X_{real})$ and $D(X_{fake})$ indicate that the discriminator's prediction of the probability of X_{real} and X_{fake} are sampled from the true data, respectively. Therefore, we maximize D by $\log D(X_{real})$ and minimizing G by $\log(1 - D(X_{fake}))$:

$$\min_G \max_D V(D,G) = E[\log D(X_{real})] + E[\log(1 - D(X_{real}))], \tag{17}$$

where $X_{fake} = [x_{t_1 \ldots t_k}, G([x_{t_1 \ldots t_k}])]$, $X_{real} = [x_{t_1 \ldots t_k}, y_{t_{k+1}}]$.

After defining the value function of the whole model, we further go into the loss functions of generator G and discriminator D. For the generator, we include not only the classification loss, but also the reward from discriminator as the total loss of the generator. We utilize categorical-cross-entropy as the classification loss function of generator, G_{cls_loss}, as computed in Eq. (18):

$$G_{cls\_loss} = -\sum_{s \in S_G} \sum_{j=1}^{Q} y_{sj} \log(\hat{y}_{sj}), \tag{18}$$

where S_G refers to the sample set of the generator and $\hat{y}_{sj}$ indicates the probability of the s^{th} sample belonging to the j^{th} category; y_{sj} refers to the one-hot encoded ground truth vector indicating whether class label j is the correct classification; Q is the number of classes.

The adversarial loss function of generator is shown in Eq. (19). $D(X^s_{fake})$ represents the discriminator's prediction of the probability of X^s_{fake} being sampled from the true data. The adversarial loss function of generator $G_{adversarial\_loss}$ will become lowest when the prediction results of generator can fool discriminator; in other words, the prediction results are similar enough to the real labels so that the discriminator cannot tell whether they are fake or real. The total loss of generator G_{loss} is shown in Eq. (20). We add the adversarial loss by classification loss since this method achieves better performance than multiplication:

$$G_{adv\_loss} = \frac{1}{|S_G|}\sum_{s\in S_G}\log(1-D(X^s_{fake})), \tag{19}$$

$$G_{loss} = G_{cls\_loss} + G_{adv\_loss}. \tag{20}$$

On the other hand, we apply binary-cross-entropy as loss function for the discriminator since in our proposed model, the discriminator is utilized as a classifier of the real and fake data. The loss function of discriminator is shown in Eq. (21):

$$D_{loss} = -\frac{1}{|S_{real}|}\sum_{s\in S_{real}}\log D(X^s_{real}) - \frac{1}{|S_G|}\sum_{s\in S_G}\log(1-D(X^s_{real})), \tag{21}$$

where S_{real} refers to the real sample set and S_G refers to the sample set of the generator.

3.4 Sampling Strategy

In order to prevent overfitting, we proposed a novel data sampling strategy. In this strategy, we assign a weight to each sample and select samples for each training epoch based on the weight. In the model training process, the weight of all training data is first initialized with equal probability and is updated at the beginning of each generator training epoch and discriminator training epoch. In regard to the generator, we consider both classification loss and adversarial loss; therefore, both of the losses are used to update the weight. In our study, we utilize reward probability $Prob^s_{Reward}$, which is calculated by reward weight $Weight^s_{Reward}$, as the distribution of selecting samples for the generator. Furthermore, considering that the generator should strengthen training data with lower loss, we should assign higher probabilities to those observations. In Eq. (22), $Loss^s_{Reward}$ denotes the reward loss of sth sample which is derived from Eq. (20). The $Weight^s_{Reward}$ is defined as the multiplicative inverse of $Loss^s_{Reward}$; therefore, the data with lower loss will weigh more and have higher sampling probability. The $Weight^s_{Reward}$ of all the N training samples are then normalized to obtain the reward probability $Prob^s_{Reward}$:

$$Prob^s_{Reward} = \frac{Weight^s_{Reward}}{\sum_{i=1}^{N} Weight^i_{Reward}}; Weight^s_{Reward} = \frac{1}{Loss^s_{Reward}}. \tag{22}$$

On the other hand, $Prob^s_{cls}$ is used as the distribution of selecting samples for the discriminator. We consider that the discriminator should focus on distinguishing between the well-classified generated samples and the real samples. The sampling probabilities for the discriminator are defined in Eq. (23):

$$Prob^s_{cls} = \frac{Weight^s_{cls}}{\sum_{i=1}^{N} Weight^i_{cls}}; Weight^s_{cls} = \frac{1}{Loss^s_{cls}} \tag{23}$$

where $Loss^s_{cls}$ is the categorical cross entropy of sth sample which is derived from Eq. (18). $Weight^s_{cls}$ of all N training samples are then normalized to obtain classification probability $Prob^s_{cls}$.

After the sampling probability update is completed, we randomly select *n* samples from the training data as the *Global Dataset*, which is shared by the generator and discriminator at the beginning of each training epoch.

We apply hierarchical sampling strategy to the generative adversarial learning. The process of the sampling strategy for the generator is as follows. *Global Dataset* shared by both generator and discriminator is selected from the training data at the beginning of each epoch. Furthermore, the sampling probability $Prob^{s}_{Reward}$ will be updated at the beginning of each generator training epoch, *G_epoch*. There are two kinds of input samples in each *G_epoch*. First, *m* samples are selected randomly from the global dataset as the general samples, in terms of maintaining the randomness of the samples. On the other hand, *m* samples are selected from the training data as the strategic samples according to the sampling probability, $Prob^{s}_{Reward}$, so that the generator can continue to strengthen learning and generate the samples with lower loss. The "*General Samples*" and "*Strategic Samples*" will be updated in each training *G_epoch*. The sampling strategy is applied to prevent overfitting and optimize the performance of the generator.

The sampling strategy process of the discriminator is as follows. The sampling probability $Prob^{s}_{cls}$ will be updated at the beginning of each discriminator training epoch, *D_epoch*. To maintain the randomness of the selected samples, *m* samples are selected randomly as the real samples from the global dataset, which is shared with the generator. On the other hand, *m* samples are selected from the training data according to the sampling probability, $Prob^{s}_{cls}$, so that the discriminator can strengthen learning to distinguish the samples generated by the generator with lower loss. The "*Real Samples*" and "*Generated Samples*" will be updated in each training *D_epoch*. By applying this method, the generated data will more likely be comprised of the low-classification-loss samples, which are the samples that are similar enough to the real data. Therefore, with the well-trained *Generated Samples*, it is more difficult for the discriminator to distinguish; thereby, it is able to optimize the performance of the discriminator.

4. EXPERIMENT AND EVALUATION

In order to evaluate the performance of our proposed method, SAR-CGAN, we executed experiments and assessed the effectiveness the components and the selected features of our model. Furthermore, we compared the proposed model with other baseline models and discuss the results.

4.1 Dataset and Experiment Setup

We conducted experiments on 3 real world datasets comprised of North American companies; their credit rating labels are S&P's credit ratings, albeit there are a few differences between the 3 datasets. D1 dataset includes 113 corporates whose CDS spread data are accessible; thus, their input features are 29 features in total (27 financial features combined with the credit systematic risk β and its quarterly mean CDS spread). On the other hand, in order to compare our model with existing methods, we extracted 20 financial statement features (the ones utilized in [13]) from D1 as the D2 dataset. D1 and D2 datasets include data on 113 companies in 17 ratings from March 2006 to December 2016.

Furthermore, we constructed the D3 dataset to evaluate the performance of our pro-

posed method on the general market of North America. The D3 dataset has 27 financial statement features, the same as those of D1 dataset. D3 dataset includes data on 959 companies in 17 ratings from March 2006 to December 2016.

To evaluate the temporal characteristic and the effect of the attention mechanism, CDS spread, and credit systematic risk β, we conducted ablation experiments within the proposed model (SAR-CGAN). We split the datasets randomly into two parts: 85% as training dataset and 15% as testing dataset. The training dataset is used for learning in the model, while the testing set is used to measure the performance of the proposed model. For each experiment, we ran 5 times each and recorded the average of the results.

4.2 Parameter Setting and Evaluation Metric

We implemented our method using Python, Tensorflow 2.0 library with NVIDIA GTX 1080 Ti. To train the models, we tried various parameter settings; the following settings achieved the best performance: learning rate for generator: 0.005, learning rate for discriminator: 0.00001, epoch: 50, batch size: 100. The settings for the elements of the generator: Self-Attention layer: 64, LSTM layer: 128; The settings for discriminator: LSTM layer: 128.

We leveraged accuracy, precision, recall, and F1 score to evaluate the prediction results. For precision, it can be interpreted as how precise the model can predict over those predicted positively. Recall, on the other hand, calculates how many of the true positives the model can capture under the total actual positives. Lastly, F1 score can be used to observe the balance between Precision and Recall. Comparison with Baseline Models

In order to evaluate the performance of our proposed model, SAR-CGAN, we compared our proposed approach with the following models on both datasets. The models are described as follows:

- Support Vector Machine (SVM): We used the one-against-rest SVM method to deal with our multiclass classification problem.
- XGBoost: XGBoost [35] is a scalable and portable version of Gradient Boosting Decision Tree.
- Multi-layer Perceptron (MLP): MLP is a simple neural network.
- Self-Multi-head Attention Gated Recurrent Unit (SMAGRU): is comprised of a Self-Multi-head Attention mechanism and GRU, as proposed in [13].
- Self-Attention Recurrent Generative Adversarial Network (SAR-GAN): We implemented the model based on our proposed model except that the input of discriminator is not conditioned.
- SAR-CGAN: The proposed model is based on Financial Statement, CDS spread, Credit Systematic Risk β feature extraction and Self-Attention Recurrent Condition-GAN with sampling strategy.

4.3 Evaluation of Selected Features

In our proposed method, we selected quarterly financial statement data (F), CDS spread and credit systematic risk β as the input data. To evaluate if the credit systematic risk β contributes to the effectiveness of the proposed model, we conducted experiment on the D1 dataset; the results are shown in Table 3.

Table 3. Evaluation of the selected features.

Features	**D1** (27 financial statement features + CDS spread feature + Credit Systematic Risk β)			
	Accuracy	Precision	Recall	F1-Score
F	0.933	0.937	0.934	0.934
F+β	0.934	0.937	0.935	0.935
F+CDS	0.939	0.943	0.940	0.939
***F+β+CDS* (SAR-CGAN)**	**0.947**	**0.949**	**0.948**	**0.948**

Table 3 shows that the combination of quarterly financial statement data (*F*) and credit systematic risk β as input data results in a better performance than the components themselves. It implies that credit systematic risk β definitely contributes to the prediction of credit rating. Moreover, the proposed model considering quarterly financial statement data (*F*), *CDS* spread and credit systematic risk β, outperforms others, including *F+β* and *F+CDS* models. It implies that quarterly financial statement data, *CDS* spread and credit systematic risk β all contribute to improve the prediction performance.

4.4 Evaluation of Time Series Characteristic

In SAR-CGAN, we utilize time series data (***F+β+CDS***) of previous *k* quarters to predict the credit rating of the next quarter. To examine the effect of time step, we conducted experiments on the D1 and D2 datasets with all of the parameters unchanged except for time step *k*.

Table 4. Evaluation of time series characteristic.

Time step	**D1** (27 financial statement features + CDS spread feature + Credit Systematic Risk β)			
	Accuracy	Precision	Recall	F1-Score
$k = 1$	0.863	0.867	0.865	0.865
$k = 2$	0.915	0.919	0.916	0.916
$k = 3$	0.924	0.927	0.924	0.924
$k = 4$	0.929	0.932	0.930	0.930
$k = 5$	0.943	0.946	0.944	0.944
$k = 6$ (SAR-CGAN)	**0.947**	**0.949**	**0.948**	**0.948**
$k = 7$	0.930	0.933	0.931	0.930
$k = 8$	0.927	0.931	0.928	0.928

Table 4 reveals that the proposed method with time step $k = 1$ has the worst performance of all; this indicates that using time series data definitely contributes to better performance. Among the time steps shown in Table, $k = 6$ reaches the best performance; therefore, our proposed method used corporate data of the previous 6 quarters to predict corporate credit ratings of the next quarter.

4.5 Evaluation of Model Components

In this section, we verify the effectiveness of each component used in our model. The innovation of our proposed method includes the Self-Attention Mechanism and the sampling strategy. In this section, we compare the proposed model, SAR-CGAN with its

variants without the 2 components. Table 5 shows that our proposed model, SAR-CGAN, maintains the best performance in accuracy, precision, recall, and F1-score compared to the proposed model without Self-Attention layer and the proposed model without sampling strategy. That is, including both the Self-Attention mechanism and sampling strategy definitely boost the performance of the proposed model.

To evaluating the effectiveness of the loss function utilized in the generator of SAR-CGAN, we compared the sum of classification loss and adversarial loss with the other two kinds of compositions: product of the two losses and classification itself.

Table 5. Evaluation of Attention mechanism and sampling strategy.

Methods	**D1** (27 financial statement features + CDS spread feature + Credit Systematic Risk β)			
	Accuracy	Precision	Recall	F1-Score
Proposed model without Self Attention mechanism	0.915	0.920	0.916	0.916
Proposed model without Sampling Strategy	0.925	0.928	0.926	0.926
Proposed model (SAR-CGAN)	**0.947**	**0.949**	**0.948**	**0.948**

Table 6. Evaluation of generator's loss function.

Loss Function	**D1** (27 financial statement features + CDS spread feature + Credit Systematic Risk β)			
	Accuracy	Precision	Recall	F1-Score
Classification Loss (C) (SAR)	0.933	0.937	0.935	0.935
C * A	0.938	0.941	0.939	0.939
C + A (SAR-CGAN)	**0.947**	**0.949**	**0.948**	**0.948**

Table 6 reveals the experimental results of the three loss functions utilized for the generator. It is found that defining the loss of the generator by considering classification loss and adversarial loss indicates better performance than only considering the classification loss; in other words, the proposed generative adversarial model is effective. Moreover, adding the classification loss and adversarial loss is superior to multiplying the classification loss by adversarial loss.

4.6 Evaluation of Compared Models

Finally, we compared the performance of the proposed SAR-CGAN model with other baselines: Support Vector Machine (SVM), XGBoost, Multi-layer Perceptron (MLP), Self-Multi-head Attention Gated Recurrent Unit model (SMAGRU). In addition, we compared a variant of our proposed model, the Self-Attention Recurrent Generative Adversarial Network model (SAR-GAN), where the input to the discriminator is not conditioned. We utilized 3 different datasets to evaluate the models.

The D1 dataset is comprised of financial statement features, CDS spread data, and Credit systematic risk β data of 113 North America companies. The comparison of the

performance between SAR-CGAN and the existing methods on D1 dataset is shown in Table 7. The proposed model, SAR-CGAN, achieves the best performance among all of the models. SAR-GAN's second-best performance indicates that adding the concept of condition to our method can improve the performance of our proposed model. Among the models, XGBoost had the third-best performance and SVM the worst.

The existing methods generally utilized financial statement features to predict corporate credit rating; however, in our proposed method, we further included corporate's credit rating spread data and credit systematic risk β in input features. Furthermore, since adversarial learning was rarely discussed in corporate credit rating prediction, we proposed a generative adversarial learning method that leveraged the advantage of Recurrent GAN and Conditional GAN to tackle corporate credit rating prediction. A novel sampling strategy is proposed herein to eliminate the overfitting problem. The results of the D1 dataset in Table 7 show that our proposed method, SAR-CGAN, outperforms all the other baselines, including SVM, XGBoost, MLP, and SMAGRU.

Table 7. Comparison of the performance of state-of-art models on D1 dataset.

Models	**D1** (27 financial statement features + CDS spread feature + Credit Systematic Risk β)			
	Accuracy	Precision	Recall	F1-Score
SVM	0.547	0.560	0.548	0.544
XGBoost	0.899	0.898	0.898	0.898
MLP	0.776	0.788	0.776	0.776
SMAGRU	0.894	0.884	0.882	0.880
SAR-GAN	0.934	0.936	0.935	0.934
SAR-CGAN	**0.947**	**0.949**	**0.948**	**0.948**

Table 8. Comparison of the performance of state-of-art models on D2 dataset.

Models	**D1** (113 companies with 27 financial statement features + CDS spread feature + Credit Systematic Risk β)				**D2** (113 companies with 20 financial statement features)			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
SVM	0.547	0.560	0.548	0.544	0.425	0.460	0.428	0.416
XGBoost	0.899	0.898	0.898	0.898	0.848	0.850	0.848	0.848
MLP	0.776	0.788	0.776	0.776	0.611	0.632	0.612	0.600
SMAGRU	0.894	0.884	0.882	0.880	0.672	0.566	0.552	0.542
SAR-GAN	0.934	0.936	0.935	0.934	0.881	0.885	0.883	0.883
SAR-CGAN	**0.947**	**0.949**	**0.948**	**0.948**	**0.895**	**0.899**	**0.896**	**0.896**

In order to evaluate the effectiveness of the features in our proposed method, we utilized the D2 dataset which includes the same companies as the D1 dataset, but only includes the financial statement features in it. The financial statement features in D2 dataset are the ones proposed in [13] as the inputs of SMAGRU. In Table 8, all of the models have higher performance on the D1 dataset comprising financial statement data, corporate's CDS spread data, and credit systematic risk β data. Specifically, SMAGRU also has higher performance in the D1 dataset compared to the D2 dataset. Hence, our proposed input features contribute to the improved prediction performance.

Moreover, to evaluate whether our proposed model performs well in relation to the general market data, we utilized the D3 dataset, which includes data on 959 North America companies to evaluate our proposed model and existing methods. The D3 dataset is comprised of 959 companies and includes 27 financial statement features, since the CDS spread data of all companies are not accessible. The results shown in Table 9 indicate that our proposed method, SAR-CGAN, outperforms the other models in general market-scale prediction.

Table 9. Comparison of the performance of state-of-art models on D3 dataset.

Models	**D3** (959 companies with 27 financial statement features)			
	Accuracy	Precision	Recall	F1-Score
SVM	0.324	0.360	0.326	0.290
XGBoost	0.861	0.862	0.862	0.862
MLP	0.525	0.542	0.524	0.518
SMAGRU	0.813	0.816	0.812	0.812
SAR-GAN	0.886	0.878	0.871	0.874
SAR-CGAN	**0.894**	**0.892**	**0.875**	**0.881**

5. CONCLUSIONS AND FUTURE WORK

In this study, we proposed a novel corporate credit rating prediction model, SAR-CGAN, based on a generative adversarial network. The proposed method considers not only financial statement features, but also CDS-related factors, including quarter mean of CDS spread and credit systematic risk β. The experiment conducted demonstrates that adding credit systematic risk β and CDS spread data definitely enhances the performance of our method. We transformed the data into time series data before sending it to the model. It is observed that using time series data helps the generator extract the latent representation of time series data, thereby improving the predictive capability of our model.

Moreover, the self-attention layer was added in our model to increase learning efficiency. In addition, we designed a sampling strategy to select data samples for adversarial training to alleviate overfitting and improve the performance of corporate credit rating prediction. The effectiveness of the model components: Self-Attention layer and proposed sampling strategy, was examined and the experimental results show that both components are effective in strengthening our proposed model.

Finally, we compared the capability of our model with other state-of-art models, including: SVM, XGBoost, MLP, and SMAGRU on three different real-world datasets. In addition, a variant of our proposed model, the Self-Attention Recurrent Generative Adversarial Network model (SAR-GAN) was compared. The better performances of SAR-CGAN compared to SAR-GAN illustrate the effectiveness of adding condition to the input of the discriminator. The experimental results of the comparisons indicate that our proposed method, SAR-CGAN, outperformed all the other models on all of the applied datasets.

In the future, we can improve our work on different aspects. For the data, we can collect and analyze data of the target companies comprehensively by adding public opinions or expert analysis of the target companies. Since more data will be added into the model, feature selection techniques need to be adopted to eliminate noise and improve the pre-

diction performance. As for the model, the generator of this work is especially suitable for numerical features. We will need to modify the architecture of the generator if textual features are adapted. Furthermore, the proposed model is a general classification method that can also be applied to other time series classification problems.

REFERENCES

1. M. S. Rizwan, G. Ahmad, and D. Ashraf, "Systemic risk: The impact of COVID-19," *Finance Research Letters*, Vol. 36, 2020, p. 101682.
2. W. A. Liu, H. Fan, and M. Xia, "Multi-grained and multi-layered gradient boosting decision tree for credit scoring," *Applied Intelligence*, Vol. 52, 2022 pp. 5325-5341.
3. Q. Lan and S. Jiang, "A method of credit evaluation modeling based on block-wise missing data," *Applied Intelligence*, Vol. 51, 2021 pp. 6859-6880.
4. P. Golbayani, I. Florescu, and R. Chatterjee, "A comparative study of forecasting corporate credit ratings using neural networks, support vector machines, and decision trees," *The North American Journal of Economics and Finance*, Vol. 54, 2020, p. 101251.
5. H.-C. Wu, Y.-H. Hu, and Y.-H. Huang, "Two-stage credit rating prediction using machine learning techniques," *Kybernetes*, Vol 43, 2014, pp. 1098-1113.
6. K.-J. Kim and H. Ahn, "A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach," *Computers & Operations Research*, Vol. 39, 2012 pp. 1800-1811.
7. Y.-C. Lee, "Application of support vector machines to corporate credit rating prediction," *Expert Systems with Applications*, Vol. 33, 2007 pp. 67-74.
8. P.-F. Pai, Y.-S. Tan, and M.-F. Hsu, "Credit rating analysis by the decision-tree support vector machine with ensemble strategies," *International Journal of Fuzzy Systems*, Vol. 17, 2015 pp. 521-530.
9. L. Yijun, C. Qiuru, L. Ye, Q. Jin, and Y. Feiyue, "Artificial neural networks for corporation credit rating analysis," in *Proceedings of International Conference on Networking and Digital Society*, 2009, pp. 81-84.
10. H.-Y. Lin, Y.-L. Hsueh, and W.-N. Lie, "Convolutional recurrent neural networks for posture analysis in fall detection," *Journal of Information Science and Engineering*, Vol. 34, 2018 pp. 577-591.
11. M. Wang and H. Ku, "Utilizing historical data for corporate credit rating assessment," *Expert Systems with Applications*, Vol. 165, 2021, p. 113925.
12. P. Golbayani, D. Wang, and I. Florescu, "Application of deep neural networks to assess corporate credit rating," *arXiv Preprint*, 2020, arXiv:2003.02334.
13. B. Chen and S. Long, "A novel end-to-end corporate credit rating model based on self-attention mechanism," *IEEE Access*, Vol. 8, 2020, pp. 203876-203889.
14. J. Hull, M. Predescu, and A. White, "The relationship between credit default swap spreads, bond yields, and credit rating announcements," *Journal of Banking & Finance*, Vol. 28, 2004 pp. 2789-2811.
15. C. A. Montgomery and H. Singh, "Diversification strategy and systematic risk," *Strategic Management Journal*, Vol. 5, 1984, pp. 181-191.

16. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv Preprint*, 2014, arXiv:1406.2661.
17. M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv Preprint*, 2014, arXiv:1411.1784.
18. H. Bharadhwaj, H. Park, and B. Y. Lim, "RecGAN: recurrent generative adversarial networks for recommendation systems," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 372-376.
19. S. Maldonado, J. Pérez, and C. Bravo, "Cost-based feature selection for support vector machines: An application in credit scoring," *European Journal of Operational Research*, Vol. 261, 2017 pp. 656-665.
20. C.-H. Chen and C.-Y. Hsieh, "Actionable stock portfolio mining by using genetic algorithms," *Journal of Information Science and Engineering*, Vol. 32, 2016, pp. 1657-1678.
21. H. M. Markowitz, *Portfolio Selection*, Yale University Press, CT, 1968.
22. L. Michalkova and K. Kramarova, "CAPM model, beta and relationship with credit rating," in *Proceedings of International Conference on Applied Economics*, 2017, pp. 645-652.
23. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv Preprint*, 2017, arXiv:1706.03762.
24. H. Zhu and T. Kaneko, "Residual network for deep reinforcement learning with attention mechanism," *Journal of Information Science and Engineering*, Vol. 37, 2021, pp. 517-533.
25. Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, 2018, pp. 261-269.
26. C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, and E. Gonina, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4774-4778.
27. M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal transformers," *arXiv Preprint*, 2018, arXiv:1807.03819.
28. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, Vol. 9, 1997 pp. 1735-1780.
29. A. T. A. Hariyanto, K. Wang, W.-C. Tsai, and C.-T. Chen, "Improving ECG SVEB detection using EMD with resampling based on LSTM," *Journal of Information Science and Engineering*, Vol. 38, 2022, pp. 977-1000.
30. S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proceedings of International Conference on Machine Learning*, 2016, pp. 1060-1069.
31. R.-C. Xie, Z.-T. Chen, and G.-S. J. Hsu, "Successive multitask GAN for age progression and regression," *Journal of Information Science and Engineering*, Vol. 37, 2021, pp. 779-792.

32. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of IEEE International Conference on Computer Vision*, 2017, pp. 2223-2232.
33. B. Feng, W. Xue, B. Xue, and Z. Liu, "Every corporation owns its image: Corporate credit ratings via convolutional neural networks," in *Proceedings of IEEE 6th International Conference on Computer and Communications*, 2020, pp. 1578-1583.
34. Z. Huang, H. Chen, C.-J. Hsu, W.-H. Chen, and S. Wu, "Credit rating analysis with support vector machines and neural networks: a market comparative study," *Decision Support Systems*, Vol. 37, 2004 pp. 543-558.
35. T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794.

Shu-Ying Lin is an Associate Professor of the Department of Finance at the Minghsin University of Science and Technology of Taiwan. She received the Bachelor of Finance degree from the National Taiwan University. She received the MBA and Master of Statistics degrees from the National Chung Hsing University and the University of Minnesota, respectively. She received the Ph.D. degree in Finance from the National Central University of Taiwan. Her research interests include corporate finance, financial innovation, machine learning and financial technology.

An-Chi Wang received the BS degree in Department of Information Management and Finance from the National Yang Ming Chiao Tung University, Taiwan. She received the MS degree in Institute of Information Management from the National Yang Ming Chiao Tung University, Taiwan. She is currently an Engineer at MediaTek Inc. Her research interests include corporate credit rating, recommender systems, and deep learning.