Bankruptcy Prediction: Artificial Neural Networks, Borderline SMOTE and Principal component analysis

Abstract

This study delves into the challenging task of corporate bankruptcy prediction, where traditional statistical methods have fallen short. To address this issue, recent advancements in machine learning, specifically Artificial Neural Networks (ANNs), are examined for their potential to provide a more accurate approach. The research uses a dataset consisting of 95 financial ratios, and employs extensive parameter tuning and cross-validation techniques to optimize the model's performance. Logistic regression is used as a baseline model and compared to the performance of neural network models in predicting bankruptcy. To balance the dataset prior to performing machine learning models, multiple oversampling strategies such as SMOTE and BSM have been employed. To further enhance the performance of ANNs, the study addresses the issue of overfitting by using principal component analysis (PCA) as a preprocessing step in the third model, 5-layer-ANN-PCA-BSM, and uses dropout and batch normalization in the second model, 6-layer-ANN-SMOTE. Both models are different modifications of the first model, the 5-layer-ANN-SMOTE model. The research findings reveal that ANNs outperform logistic regression in predicting bankruptcy. Additionally, the study demonstrates that PCA and Borderline SMOTE are effective in improving the performance of ANNs, as evidenced by the 5-layer-ANN-PCA-BSM model's accuracy, recall, and F1-score equal to 0.96, 0.98, and 0.98, respectively. Overall, this study provides novel insights into the use of ANNs for corporate bankruptcy prediction, and the effect of incorporating oversampling strategies and preprocessing techniques such as PCA, dropout, and batch normalization, which can significantly improve model performance. These findings have important implications for financial institutions and investors, who can leverage these techniques to better predict and mitigate bankruptcy risk.

1 Introduction

Predicting corporate bankruptcy has been a challenging task for financial institutions and investors, as it can have significant consequences on various stakeholders. For investors and financial analysts, predicting the likelihood of a company's bankruptcy allows them to make informed investment decisions and mitigate risk. For the management of a company, bankruptcy analysis using financial ratios can serve as an early warning about the company's prospects, allowing them to take safeguarding actions in order to avoid bankruptcy and improve performance [1]. These early warning signs can also benefit creditors in deciding the limit of their exposure to the company by reducing their credit limit or demanding collateral.

Numerous academic scholars have attempted to create bankruptcy prediction models utilising various modelling methodologies due to its aforementioned significance [2]. Traditional methods of predicting bankruptcy, such as ratio analysis and regression analysis, have limitations, and their accuracy is questionable. However, recent advances in machine learning, specifically in neural network models, have shown promising results in predicting bankruptcy [3].

Our research aims to break new ground by going beyond the limitations of existing literature, which mostly focuses on using only 5-15 financial ratios in bankruptcy prediction research. This limitation has potentially hindered the accuracy of previous studies, as a more comprehensive assessment of a company's financial health may require consideration of a broader range of financial ratios. By analyzing a dataset containing a wider range of 95 financial ratios, our research seeks to provide a more nuanced and accurate approach to bankruptcy prediction.

This study focuses on the application of various Artificial Neural Networks (ANNs) to predict company bankruptcy, using parameter tuning and cross-validation techniques to optimize the model's performance. Over the past few years, ANNs have become a popular algorithm for bankruptcy prediction, with recent studies showing superior performance compared to traditional statistical methods. This can be attributed to their ability to accurately identify patterns and relationships between financial ratios and the likelihood of bankruptcy. Moreover, in high-dimensional datasets, such as ours, which contains 95 ratios, traditional machine learning models such as Support Vector Machines (SVMs) and K-Nearest Neighbors (KNNs) may produce unreliable accuracy scores and require feature selection methods for optimization. However, ANNs can efficiently handle vast amounts of financial data without feature selection, thus providing better predictions of bankruptcy. Additionally, ANNs are effective in dealing with missing and biased data, non-linear systems, and complex correlations, making them a leading choice in bankruptcy prediction research [3].

However, Artificial Neural Networks (ANNs) have a few drawbacks. It suffers from a well-known issue, namely overfitting, which arises when the model is excessively complex and closely fits the training data, leading to poor generalization performance on new data. To address this problem, one effective approach is to employ principal component analysis (PCA) in ANN models. PCA is a commonly used method that can reduce the dimensionality of the input data by removing irrelevant features and combining highly correlated ones while preserving the essential information of the data. This technique has been widely applied in ANN models as a preprocessing step to improve their performance and prevent overfitting [4].

In our study, we employ logistic regression as the baseline model to compare its performance with neural network models in bankruptcy prediction and loan defaults. Logistic regression is a widely accepted and established model in this field, owing to its unique features such as simplicity, interpretability, and computational efficiency. Its extensive use in previous studies has established a challenging benchmark for more complex models, such as neural networks, to outperform. Our research seeks to extend the existing literature by assessing the performance of neural network models in comparison to logistic regression and identifying the relative strengths and weaknesses of each method. Through this approach, our study aims to contribute to the ongoing discussion on the effectiveness of different methods in predicting financial distress and inform best practices in the field.

Data integrity and quality are critical factors in developing accurate models. However, recent research has overlooked the issue of imbalanced corporate bankruptcy datasets, as highlighted by Smiti & Soui (2020). Imbalanced data can result in biased models that perform poorly in predicting the minority class. To address this, we employ well-established data balancing methods such as SMOTE (Synthetic Minority Oversampling Technique) and BSM (Borderline Synthetic Minority Oversampling), proposed by Chawla et al. (2002) [5], before applying our neural network models, which we will discuss in later sections.

1.1 Literature concerning bankruptcy prediction

Wilson and Sharda (1994) investigated the effectiveness of neural networks in predicting corporate bankruptcy, against that of classical multivariate discriminant analysis. The results indicated that the neural network model outperformed discriminant analysis irrespective of the composition of the bankruptcy and non-bankruptcy firms in the training and testing sets [3].

More recently, Naidu and Govinda (2018) used ANN and Random Forest to construct bankruptcy prediction models. The results showed that the neural network model with the sigmoidal activation function had an error of 4.4349% after 150 epochs. The classification using random forest had a larger inaccuracy of 5.1954%. Thus, neural networks were found to be more trustable than random forests in bankruptcy prediction [15].

Durica (2019) employed a decision tree methodology to forecast the financial standing of Polish firms. Specifically, the study leveraged a classification and regression tree approach in addition to a chi-square automatic interaction detector, yielding credible outcomes [16].

Adisa et al. (2019) created a hybrid model that predicts firm bankruptcy by combining PCA and ANN. In ten-fold cross-validation, study findings demonstrated that PCA-ANN with 32 main components beat other models in terms of training/testing ratio performance [17].

Horak et al. (2020) compared SVM and ANN-based bankruptcy prediction models. A multi-layer perceptron artificial neural network (MLP) and a radial basis function artificial neural network (RBF) were the two types of ANN that were utilised. The findings indicated that the SVM model had an accuracy of about 76%. whereas the ANN-based model achieved an accuracy of 82.79% [18].

Wang (2017) proposed a bankruptcy prediction model that utilized three distinct techniques, namely support vector machines (SVM), a neural network with dropout, and autoencoder. According to the results, the neural network with additional layers had the highest level of accuracy compared to the other methods. Nonetheless, the study had several limitations, including the fact that the SVM method did not provide direct probability estimates and relied on an expensive five-fold cross-validation method. Additionally, when the dataset was insufficient, particularly if the number of instances was less than the number of features, the SVM technique tended to produce subpar results [19].

J Wirgård (2022) employed a Multilayer Perceptron (MLP) model to predict corporate bankruptcies. The model achieved an average accuracy of 82.8% and a standard deviation of 0.0678% after 120 epochs and 30 trials, outperforming two support vector machine (SVM) models. The number of employees, turnover group, and equity ratio were found to have the greatest impact on the bankruptcy prediction, suggesting that they may be important factors to consider when analyzing the likelihood of a firm going bankrupt [20].

2 Methodology

2.1 Dataset

The Taiwan bankruptcy dataset, sourced from the Taiwan Economic Journal for the years 1999-2009, comprises 6819 entries, 96 ratios, and 2 classes. Its objective is to predict a company's bankruptcy within a year based on its financial data. The target variable is binary, with 1 indicating a bankrupt company and 0 representing a non-bankrupt company. The dataset is highly imbalanced, with 6599 (96.7%) non-bankrupt and only 220 (3.3%) bankrupt firms, based on the regulations of the Taiwan Stock Exchange. The dataset was obtained from a reliable Kaggle source established in 2019 [7]

2.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique used to extract and represent the underlying patterns in a high-dimensional data set. By reducing the number of features and eliminating noise in the data, PCA can improve the performance of prediction algorithms. PCA achieves this by identifying the principal components (PCs), which are a new set of variables that explain the most variance in the data. The first PC is identified based on the eigenvector of the covariance matrix, with the variable holding the most significant variance represented by the highest eigenvalue. Subsequent PCs are identified by the eigenvectors of the remaining major components. Mathematically, for a dataset of n data points $x_1, x_2, \dots, x_n$ in a high-dimensional space with d dimensions, PCA finds the d' principal components that explain the most variance in the data.

2.3 Artificial Neural Network (ANN)

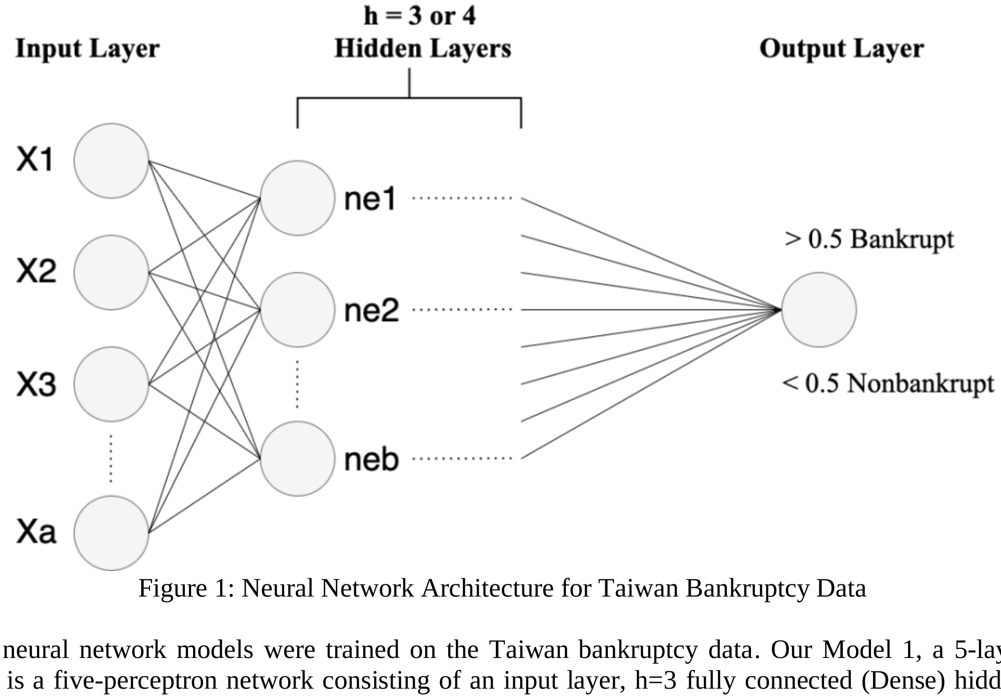

Figure 1: Neural Network Architecture for Taiwan Bankruptcy Data

Multiple neural network models were trained on the Taiwan bankruptcy data. Our Model 1, a 5-layer-ANN-SMOTE, is a five-perceptron network consisting of an input layer, h=3 fully connected (Dense) hidden layers, and the output layer with the sigmoid activation function. The input layer consists of Xa=95 nodes, one for each financial ratio, and the three hidden layers have neb=51 nodes each. The output layer has only one neuron with a response of 0 representing non-bankrupt, and 1 representing bankrupt firms. The model classifies data on a scale of 0 to 1 and for firms with output above 0.5 are categorised as bankrupt firms while below 0.5 are for firms which are recognised as non-bankrupt [11].

The parameter values fitted into each model is through extensive hyperparameter training executed for each model. For model 1, 6 parameters: no. of neurons, activation function, batch size, epochs, learning rate and optimizer value are tuned by simulating temporary neural network models along with employing stratified kfold cross validation technique with k=5 to best optimise model performance. The performance is optimised by maximising validation score based on the metric F1-score. F1-score is chosen, rather than accuracy score, as the main metric as accuracy can be misleading when dealing with imbalanced data [12]. Thus, Model 1 is defined as a 5-layer network with SELU activation function between the layers (excl. output layer), batch size = 851, learning rate = 0.0371, optimizer = 0.7373 and 51 nodes for each hidden layer. The Scaled Exponential Linear Unit (SELU) activation was first introduced by Klambauer et. al (2017) which induces self-normalizing properties. It can be mathematically expressed as:

$$\begin{aligned} f(x_i) &= \lambda x_i\,, \quad if\ x_i > 0 \\ f(x_i) &= \lambda\alpha(e^{x_i} - 1), \quad if\ x_i < 0 \end{aligned} \tag{1}$$

where x_i is the layer input, α and λ (known as scale) are pre-defined constants, α = 1.673 and λ = 1.050. The values of alpha and scale are selected so that the mean and variance of the inputs are kept between two subsequent layers [13].

The loss parameter specifies the objective function that the model will optimize during training. In this case, binary cross-entropy is used as the loss function. This loss function is commonly used in binary classification problems, where the output of the model is a probability distribution over two classes. For this reason, all three models are trained using binary cross-entropy loss function. This function can be expressed as:

$$Binary\ Cross\ Entropy\ Loss = \frac{1}{n}(\Sigma_{i=1}^{n} - y_i \log{(p_i)} - (1 - y_i)\log{(1 - p_i)} \tag{2}$$

where y_i represents our actual bankruptcy variable value, p_i is the probability of class 1 and $1 - p_i$ is the probability of class 0. Also, all our models are built using Tensorflow, thus it automatically adjusts the weights and biases through backpropagation when the output of a previous layer is fed into as an input for the next layer. Adam optimizer function is applied for Model 1 as it works pretty well on relatively large datasets (with thousands of training samples or more) in terms of both training time and validation score.

Model 3, the 5-layer-ANN-PCA-BSM, is similarly a five-layer neural network, however, the input layer now has Xa=41 nodes, rather than 95, due to Principal Component Analysis (PCA). Each of the three hidden layers consists of 53 neurons and the hyperparameter tuning resulted in RELU activation function being selected, along with batch size = 885, epochs=36 and learning rate = 0.01. The Adam optimizer function is also applied for this model. RELU (Rectified Linear Unit) is a piecewise linear function that, if the input is positive, outputs the input directly; else, it outputs zero. This can be expressed as:

$$f(x_i) = \max(0, x_i) \tag{3}$$

where x_i is our layer input. A major modification of Model 1 gives us Model 2, which is our 6-layer-ANN-SMOTE model. This model consists of 6 layers, which are selected by tuning the no. of hidden layers. The input layer of Model 2 remains same i.e. consisting of 95 nodes and the following four hidden layers have 26 nodes each. An additional hyperparameter called normalization is employed so the model can add a Batch normalization layer whenever normalization is greater than 0.5. Batch normalization (also known as batch norm) is a method used to make training of artificial neural networks faster and more stable through normalization of the layers' inputs by re-centering and re-scaling, which was first presented by Joffe et al. (2015) [14]. It takes the layer inputs (x_i) as a batch and normalises it using its mean (E_{x_i}) and variance (Var_{x_i}). This resulting value is employed in scaling and shifting the inputs to obtain the output of the layer, which can be mathematically expressed as:

$$z_i = \frac{\gamma}{\sqrt{\left(Var_{x_i} + \epsilon\right)}} x_i + \left(\beta + \frac{\gamma E_{x_i}}{\sqrt{\left(Var_{x_i} + \epsilon\right)}}\right) \tag{4}$$

where z_i is the updated output of the layer and $\beta\ and\ \gamma$ are the learned offset and scaling factors respectively, which are learned during training along with the original parameters.

Dropout and dropout rate are another additional hyperparameters added to the model, through which if the dropout is found to be greater than 0.5, then a Dropout layer using the dropout rate is added to our model. Dropout is a regularization technique used to randomly drop out some neurons during training to prevent overfitting. The rest of the tuned hyperparameters for Model 2 are found to be: batch size = 852, dropout = 0.482, dropout rate = 0.021, epochs = 49, learning rate = 0.1877, normalization = 0.949 and the optimal activation function is softplus. Softplus can be imagined as a smooth layer of RELU, which for an layer input (x_i) can be given as:

$$f(x_i) = \log\left(1 + e^{x_i}\right) \tag{5}$$

2.4 Logistic Regression Model (Baseline Model)

Logistic Regression is an extensively employed approach that has also been used significantly in predicting bank bankruptcy and loan defaults, such as by Ohlson (1980) and Altman (1968). In this model, we use a sigmoid function to transform the input features into a probability value between 0 and 1, which represents the likelihood of the sample belonging to the positive class (default) or the negative class (no-default). So, for our conditional probability of observing a default for bank i, denoted as $p_i = Prob(y_i = 1|X)$ where X represents a vector of $x_1, x_2, \ldots., x_p$, for p features (95 features in the context of our dataset), we get our logistic regression equation as:

$$p_i = \frac{1}{1 + \exp\left(-\left(\beta_0 + x_1\beta_1 + x_2\beta_2 + \ x_3\beta_3 \ + \cdots x_p\beta_p\right)\right)} \tag{6}$$

where $\beta_0, \beta_1, \beta_2 \ldots \beta_p$ are the coefficients that determine the impact of features on predictor variable.

2.5 Synthetic Minority Oversampling Technique (SMOTE)

To overcome the issue of imbalanced dataset, two balancing methods can be applied: undersampling and oversampling methods. While oversampling entails taking new data points of minority classes up until the number of minority classes is about balanced, undersampling tries to eliminate some majority classes. Recent research has shown that the oversampling methodology is superior to the undersampling strategy. Furthermore, the undersampling strategy runs the danger of removing the relevant case from the collection, thus causing loss of valuable information. On the other side, oversampling seeks to maintain the dataset's original contents while preserving all data. Consequently, the oversampling method outperforms the undersampling strategy [6,8].

One such oversampling technique is Synthetic Minority Oversampling Technique (SMOTE). SMOTE generates synthetic examples for the minority class by creating new observations that are similar to, but not identical to, existing observations. In the context of our dataset, we assume that the minority class is the "default" class, and the majority class is the "non-default" class.

The SMOTE algorithm can be implemented using the following equations. Let x be an observation in the minority class, and let D be the set of all observations in the dataset. SMOTE selects k nearest neighbours from D for x by computing the Euclidean distance between x and all other observations in D and selecting the k observations with the smallest distance. For each of the k nearest neighbours, SMOTE generates a new example z by adding a fraction r (between 0 and 1) of the difference between x and y to x, as follows:

$$z \ = \ x \ + \ r \ * \ (y \ - \ x) \tag{7}$$

where r is a random number between 0 and 1. This results in the creation of new synthetic observations in the minority class that lie on the line segment between x and its k nearest neighbours. SMOTE repeats this process for all observations in the minority class, adding the new examples to the minority class. While this technique improves the performance of machine learning algorithms on the minority class, it can also lead to overfitting and decreased performance on the majority class if not balanced carefully for optimal performance.

2.6 Borderline Synthetic Minority Oversampling (BSM)

BSM (Borderline-SMOTE) is an extension of SMOTE that addresses the issue of generating noisy synthetic examples near the decision boundary between the minority and majority classes. The algorithm first identifies the minority class samples that are near the decision boundary, called borderline samples. These samples are then oversampled using the SMOTE algorithm.

Mathematically, to identify that a minority class observation x with k-nearest neighbours is near the decision boundary, it must satisfy the condition:

$$\frac{k}{2} \leq k' \leq k \tag{8}$$

where k' is the no. of samples out of k that are actually majority class observations $D - x$, where D again is the set of all observations. This set of x satisfying the above condition are called borderline samples. These samples are then oversampled using the SMOTE algorithm, using the same equation (7) which thus adds new minority samples to match the total number of majority class observations, giving us a balanced dataset.

In general, BSM may be more effective in cases where the decision boundary between the minority and majority class is not well-defined, while SMOTE may be more effective in cases where the minority class instances are well separated from the majority class. For Model 1, which is our 5-layer-ANN, we consider all 95 ratios for the binary classification model where we perform oversampling using SMOTE. However, after performing PCA and reducing the dimensionality to almost half in Model 3, the dataset may have become more separable and the decision boundary between the minority and majority classes has shifted. To preserve the decision boundary even when the dataset is more separable, it might be more appropriate to use BSM rather than SMOTE [9]. This also supported in avoiding overfitting and improved the performance of our deep learning algorithm on the minority class.

2.7 Proposed Methodology

The research begins by downloading the Taiwan bankruptcy data from Kaggle [7]. After acquiring the dataset, we perform exploratory data analysis before and after splitting the data into training and testing sets. The EDA steps include checking for missing and duplication data, outlier treatment and data normalization.

To acquire comprehensive results, we take two initial versions of datasets: with and without PCA. The PCA technique is applied to perform dimensionality reduction and to compare its performance with other non-PCA models in order to analyse the extent of overfitting on them and gauge its overall performance. According to Hair et al. (2013), the number of Principle Components (PCs) can be determined based on the number of factors that can meet a given proportion of the stated variance that will be researched, which is typically as high as 60% of the total number of components or more [10]. We follow this by reducing our total number of components by around 40%, by choosing to retain 90% of the total variance of the data. After experimenting with the no. of layers along with hyperparameter tuning, we decide to select 5 layers which optimised our model performance for Models 1 and 3. We also performed a more extensive hyperparameter tuning by also tuning the no. of layers in the simulation, which gives us our ANN Model 2. The research steps taken in the proposed methodology are illustrated in Figure 2.

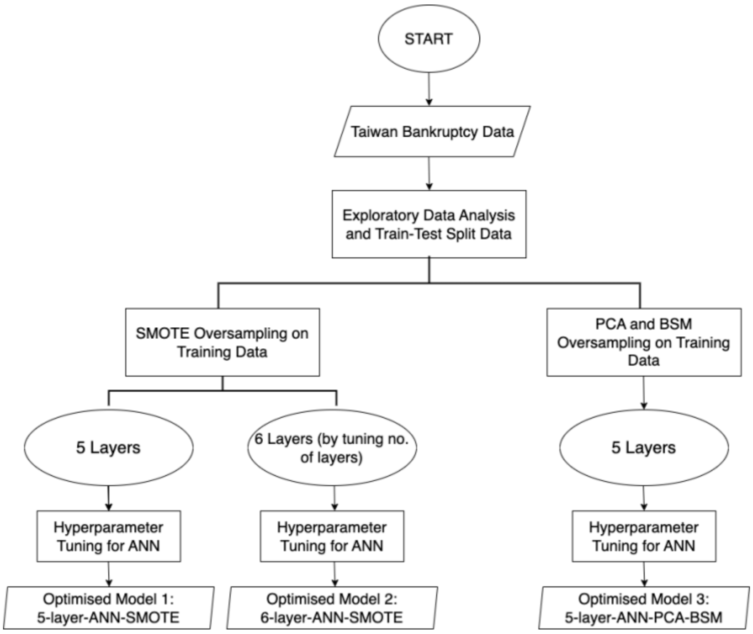
START
Taiwan Bankruptcy Data
Exploratory Data Analysis and Train-Test Split Data
SMOTE Oversampling on Training Data
PCA and BSM Oversampling on Training Data
5 Layers
6 Layers (by tuning no. of layers)
5 Layers
Hyperparameter Tuning for ANN
Hyperparameter Tuning for ANN
Hyperparameter Tuning for ANN
Optimised Model 1: 5-layer-ANN-SMOTE
Optimised Model 2: 6-layer-ANN-SMOTE
Optimised Model 3: 5-layer-ANN-PCA-BSM

Figure 2: Flowchart of the Proposed Methodology

2.8 Exploratory Data Analysis and Class Distribution

After a deep check, no missing or duplicate values were found, indicating high-quality data and reducing the need for extensive cleaning. Outlier treatment is an essential step in data cleaning for machine learning modelling to ensure that the model learns meaningful patterns from the data and generalizes well to new data. For outlier treatment across our 98 ratios, we apply the Tukey's method for outlier detection. The method defines outliers as values that fall outside of the range of 1.5 times the interquartile range (IQR) below the lower quartile or above the upper quartile. Around 8% of the dataset was identified as outliers and removed from our dataset. The resulting dataset was split into two parts: training data (70% of banks examined) and testing data (30%). The following process is the data standardization process. It involves transforming the features so that they have similar ranges or magnitudes. This is important because many machine learning algorithms, including logistic regression and KNN, are sensitive to the scale of the features. To avoid data leakage and minimize outlier impact, we fitted Min Max Scaler solely to the training data before using it to scale both our training and testing datasets. A crucial step often mistaken during balancing the dataset, is to apply the oversampling technique to only the training data, rather than the entire dataset. When oversampling is applied to the entire dataset, there is a risk of information leakage from the test set to the training set. In other words, the oversampling algorithm may inadvertently create duplicate copies of some of the minority class, resulting in overestimating a model's performance on the test set. As discussed in the prior section, we apply SMOTE and BSM to our respective training datasets. The effect of the oversampling methods on the training data is illustrated in Figure 3.

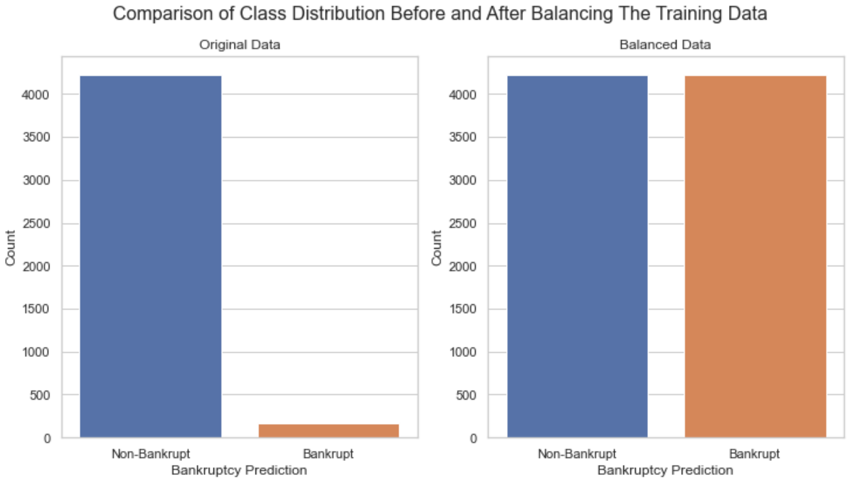

Figure 3: Comparison of Class Distribution Before and After Balancing The Training Data

3 Results

3.1 Evaluation Metrics

In this study, the performance of various models is evaluated using evaluation metrics based on the classification report. The metrics used are accuracy, precision, recall, and F1-scores, which are optimized using stratified k-fold cross-validation and hyperparameter tuning. Each metric can be expressed in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Accuracy is the measure of how often a model makes correct predictions. It is calculated by dividing the number of correctly predicted instances (true positives and true negatives) by the total number of instances. Precision measures how often the model's positive predictions are correct. It can be calculated as a ratio of true positives to the total predicted positives. Recall measures how well the model identifies true positive instances using the ratio of true positives to all possible positives. The F1-score is the harmonic mean of precision and recall, taking both into account. The formula for each metric is shown below.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1\,Score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{12}$$

In the context of imbalanced datasets, it is essential to assess the models' performance on both the majority and minority classes. While the minority class is often the class of interest and requires detection, evaluating the performance measures for both classes is crucial to obtain a comprehensive understanding of the model's performance. Therefore, we need to consider the performance of both classes to determine the effectiveness of the models.

Oversampling Method	**Models**	**Accuracy**	**Precision**	**Recall**	**F1-score**
Balanced data using SMOTE Oversampling	Logistic Regression	89%	99%	90%	94%
	5-layer-ANN-SMOTE	95%	98%	97%	98%
	6-layer-ANN-SMOTE	90%	99%	90%	95%
Balanced data using BSM Oversampling	5-layer-ANN-PCA-BSM	96%	98%	98%	98%

Figure 4: Performance metrics table of four different models – Logistic Regression, 5-layer-ANN-SMOTE, 6-layer-ANN-SMOTE and 5-layer-ANN-PCA-BSM for the majority (non-bankrupt) class

Upon analysing the models trained on balanced data, it is evident that all three neural network models have achieved high accuracy rates, ranging from 90% to 96%. However, as the data was highly imbalanced, the evaluation statistics depicted in Figure 4 were biased towards the majority class. Though the baseline logistic regression model achieved high precision, the recall and F1-score were lower compared to all other models. The reason for this could be because precision does not account for true positive cases that were missed. This fact can be attributed to the lower recall and F1-score. Among all the neural network models, the 6-layer-ANN-SMOTE model had an underwhelming performance with recall and F1-score similar to the logistic regression. On the other hand, the 5-layer-ANN-PCA-BSM model stands out with the highest accuracy score, recall, and F1-score. A high accuracy score indicates that the model makes the least incorrect predictions, high recall indicates the greatest number of true positives, and F1-score indicates a good balance between recall and precision. Therefore, for the majority class, the 5-layer-ANN-PCA-BSM is currently the best-performing model amongst all.

Oversampling Method	**Models**	**Accuracy**	**Precision**	**Recall**	**F1-score**
Balanced data using SMOTE Oversampling	Logistic Regression	89%	17%	68%	27%
	5-layer-ANN-SMOTE	95%	29%	30%	34%
	6-layer-ANN-SMOTE	90%	19%	74%	31%
Balanced data using BSM Oversampling	5-layer-ANN-PCA-BSM	96%	35%	30%	32%

Figure 5: Performance metrics table of four different models – Logistic Regression, 5-layer-ANN-SMOTE, 6-layer-ANN-SMOTE and 5-layer-ANN-PCA-BSM for the minority (bankrupt) class

To assess the performance of our models on the minority class, we examine the evaluation statistics presented in Figure 5. The results are significantly less impressive than those for the majority class. This is because our original data was heavily imbalanced, with only 3.3% of the data pertaining to bankrupt companies, providing limited information about the minority class. As a consequence, neural network models are particularly valuable, as they can identify underlying trends and patterns from the limited bankrupt data available. As we can observe, these models show considerable improvement over the logistic regression model in predicting bankruptcies in the minority class. That is why we see these models being a wonderful improvement than logistic regression in the prediction of minority bankrupt cases. Both 5-layer-ANN models have precision, recall and F1-score equal to or exceeding 30% while the 6-layer-ANN-SMOTE has had tremendous increase in performance since majority class when compared to the other models. When working with highly imbalanced datasets, the F1-score is a more meaningful metric than accuracy [12]. The 5-layer-ANN-SMOTE model's performance in predicting minority cases was impressive based on all metrics, especially F1-score. The 5-layer-ANN-PCA-BSM model was a close runner-up, with good performance across all metrics. Its precision rate of 35% was more than double that of the logistic regression model (17%).

4 Discussion

With the first and second neural network model, we decided to perform oversampling of the minority class without performing principal component analysis. While this resulted in the models to display a satisfactory performance, we realise that our third model exceeded both of them on the basis of the aforementioned evaluation seen across the majority and the minority class. This is believed due to PCA’s ability in reducing unnecessary features, thus reducing chances of overfitting with the training data. It could also be linked to the use of Borderline SMOTE as the oversampling method, which is an extended version of SMOTE that is considered more efficient in dealing with data with high complexity and in cases, when the data points are close to the decision boundary of bankrupt and non-bankrupt classes.

Further, our logistic regression model while performing well in predicting majority class, showed underwhelming performance in the bankrupt minority class prediction. This could be attained due to the limitations of the model in identifying the underlying trends, especially for the minority class which represents only 3.23% of our total dataset. One important observation to notice is that the model rendered significant accuracy across the classes, however the evaluation of the other metrics: precision, recall and F1-score deemed its performance otherwise. This reemphasizes the limitation of accuracy score which fails to evaluate the wrong predictions made by the model and tends to display amazing score for models that overfit the training data. Thus, its crucial to always evaluate a model using multiple metrics such as precision, recall and F1-score.

Thus, we conclude that our third model, 5-layer-ANN-PCA-BSM is the best amongst all 4 models on predicting bankruptcy on the Taiwan bankruptcy data from the 1999 to 2009 period.

In terms of modelling the bankruptcy performance, there is also significant potential in designing models that combine a traditional and advanced machine learning method, that will allow to utilize the best features of both types of models in optimising performance. Further, more sophisticated neural network models can be designed with the inclusion of autoencoders and CNNs in a network model architecture, which has already registered success in other fields such as in cancer research and other medical sciences.

This study aims to transcend the existing literature with the inclusion of many other financial ratios in order to predict a company’s financial situation. Most research papers, especially based on datasets originating from EMEA and Americas, have been constrained on utilizing few financial ratios which have been followed since decades. However, as economies continue to go through increasing dynamic changes due to financial crises, wars and pandemics, a need for change in existing methods of a company’s financial evaluation is required. Investors and creditors have become more concerned since COVID about betting on a company’s growth. Thus, our paper intends to shine light on newer datasets and methods to reimagine the focus of investors, creditors, financial analysts and other important shareholders concerning a company.

References

[1] Siegel, J. G. (1981). Warning signs of impending business failure and means to counteract such prospective failure. National Public Accountant, 26(4), 9-13.

[2] Gissel, J. L., Giacomino, D., & Akers, M. D. (2007). A Review of Bankruptcy Prediction Studies: 1930-Present. In Journal of Financial Education (Vol. 33). Publisher Link.

[3] Wilson, R. L., & Sharda, R. (1994). Bankruptcy prediction using neural networks. Decision support systems, 11(5), 545-557.

[4] Nasution, M. Z. F., Sitompul, O. S., & Ramli, M. (2018, March). PCA based feature reduction to improve the accuracy of decision tree c4. 5 classification. In Journal of Physics: Conference Series (Vol. 978, No. 1, p. 012058). IOP Publishing.

[5] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[6] Smiti, S., & Soui, M. (2020). Bankruptcy prediction using deep learning approach based on borderline SMOTE. Information Systems Frontiers, 22, 1067-1083.

[7] Fernando, S. (2019). Company Bankruptcy Prediction: Bankruptcy data from the Taiwan Economic Journal for the years 1999–2009. Kaggle

[8] Tumpach, M. (2020). Prediction of the bankruptcy of Slovak companies using neural networks with SMOTE. Ekonomický časopis.

[9] Sulistiani, I., & Nugraheni, M. (2022, August). Comparison of Bankruptcy Prediction Models Using Support Vector Machine and Artificial Neural Network. In 2022 11th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS) (pp. 316-321). IEEE.

[10] Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2013). Multivariate data analysis: Pearson new international edition PDF eBook. Pearson Higher Ed.

[11] Odom, M. D., & Sharda, R. (1990, June). A neural network model for bankruptcy prediction. In 1990 IJCNN International Joint Conference on neural networks (pp. 163-168). IEEE.

[12] Gupta, A. (2021, March 9). Problems with Precision and Recall. Analytics Vidhya.

[13] Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-Normalizing Neural Networks. In Advances in neural information processing systems (pp. 971-980)

[14] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456).

[15] Naidu, G., & Govinda, K. (2018). Bankruptcy prediction using neural networks. In 2018 2nd International Conference on Inventive Systems and Control (ICISC) (pp. 342-345). IEEE

[16] Durica, M., Frnda, J., & Svabova, L. (2019). Decision tree based model of business failure prediction for Polish companies. Oeconomia Copernicana, 10(3), 453-469.

[17] Adisa, J. A., Ojo, S. O., Owolawi, P. A., & Pretorius, A. B. (2019, November). Financial distress prediction: principle component analysis and artificial neural networks. In 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC) (pp. 1-6). IEEE.

[18] Horak, J., Vrbka, J., & Suler, P. (2020). Support vector machine methods and artificial neural networks used for the development of bankruptcy prediction models and their comparison. Journal of Risk and Financial Management, 13(3), 60.

[19] Wang, N. (2017). Bankruptcy prediction using machine learning. *Journal of Mathematical Finance*, *7*(04), 908.

[20] Wirgård WIklund, J. (2022). Predicting company bankruptcy using artificial neural networks.: Visualization and ranking of key features.