2020 First International Conference on Power, Control and Computing Technologies (ICPC2T)

Class Imbalance Deep Learning for Bankruptcy Prediction

Shanmukha Vellamcheti
Computer Science and Engineering
National Institute of Technology Raipur, India
shannu31051999@gmail.com

Pradeep Singh
Computer Science and Engineering
National Institute of Technology Raipur, India
psingh.cs@nitrr.ac.in

***Abstract*— This paper addresses one of the most burning issues among financiers namely bankruptcy prediction. For Bankruptcy prediction many researchers have used various methods ranging from Statistical Modeling to Machine Learning. Training a robust Machine Learning model to accurately predict whether a company goes bankrupt or not is a challenging issue in the sense that real life data is often imbalanced. Hence, we first reduce the imbalance nature of data and then train a Deep Neural Network with the balanced data. The dataset used is that of Polish companies which consists of five years of data corresponding to five different tasks. We reduce class imbalance using an Oversampling method known as Synthetic Minority Oversampling Technique (SMOTE). Our model significantly outperforms the previous neural network models and weak learners trained on this dataset in terms of Area Under Receiver Operator Characteristic Curve (AUC).**

Keywords*— *neural network; bankruptcy prediction; imbalanced data;

I. Introduction

Bankruptcy prediction has been a pressing issue among investors and policy makers from quite some time. It is essential because of it's influence on economic decisions. It helps in the prediction of business distress [1] of a firm well in advance which gives ample time for the concerned people to take preventive measures. Hence this issue has grabbed the eyeballs of many researchers around the globe.

However this issue demands the need for large amount of historical data which can be processed for drawing insights. Often firms have different indicators which are mathematically modeled in order to describe it's financial state [2]. This data can prove to be very useful, especially for predicting bankruptcy.

Despite the data, accurate prediction of bankruptcy is still a challenge because of the class imbalance problem. Class imbalance is a situation where the distribution of data corresponding to the classes is uneven. This makes the model biased which further leads to inaccurate predictions. This paper addresses this issue and improves the area under curve (AUC) for neural networks. The rest of this paper is organised as follows. Section 2 gives outline of literature review and Section 3 describes the proposed methodology using neural networks. Section 4 contains details about dataset and preprocessing. Whereas Section 5 analyses results and Section 6 draws out the conclusion.

II. Literature Review

The first attempts for bankruptcy prediction were started by Beaver [3] who applied statistical models. Later, Altman [4] used multi dimensional analysis to predict bankruptcy. But since the nineties, with the rise of Artificial Intelligence, researchers started employing Machine Learning methods to tackle this problem.

Support Vector Machine (SVM) by Shin et al [5] was one of the most successful machine learning model. But the inability of these to comprehend non linear data led to Neural Networks implementation by Zhang et al [6]. Zieba et al [7] used Extreme Gradient Boosting for predicting bankruptcy. L. Shi and L. Xi [8] used an ensemble of Neural networks and G. P. Naidu and K. Govinda [9] used Neural Networks for bankruptcy prediction.

In [13] S. Karlos and N. Fazakis studied the effectiveness of semi-supervised learning on bankruptcy prediction.
M. Wagle proposed different data mining techniques including Bayesian Networks, Support Vector Machines and Neural Networks for Bankruptcy prediction [14]. A new method hybridizing LogitBoost algorithm with correlation based feature subset selection was introduced by G. Kumar and S. Roy [15]. S. Fan et al [16] used Isolation forest to detect anomaly in bankruptcy prediction. S. Joshi et al [17] used a combination of Genetic algorithm and Random forest for prediction of bankruptcy. However, these papers didn't address the class imbalance challenge.

III. Proposed Methodology

Deep Neural Network (DNN) is a type of statistical model which is inspired by the functioning of biological human brain. The major difference between conventional Machine Learning model and a Deep Neural Network is that the former demands the features to be hand picked from the dataset whereas the latter can extract the features all by itself. This makes them very powerful and flexible. Researchers over a period have developed many variants of NN such as Multi

Layer Perceptron (MLP), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) etc.

In this experiment we use an MLP with an input layer, two hidden layers along with an output layer. The distribution of neurons in each layer is shown in Table 1.

TABLE I
DISTRIBUTION OF NEURONS

Layer Type	No. of Neurons
Input Layer	64
Hidden Layer 1	256
Hidden Layer 2	20
Output Layer	2

Input layer consists of 64 neurons as the dataset consists of 64 feature columns. The Output layer consists of 2 neurons which represents the possibility of two classes: Bankrupt and Not Bankrupt. The dataset is described more extensively in the further sections. Hidden layers and Neurons are selected after several attempts of trail and error for the best combination. A sample Neural Network architecture is shown in Fig. 1.

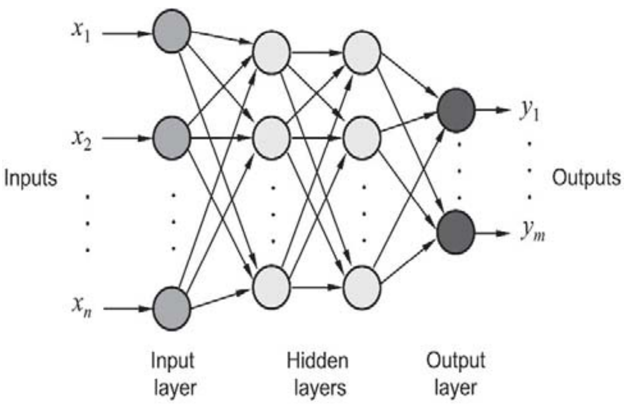

Fig. 1 A sample MLP with two hidden layers

Learning rate of 0.001, Number of epochs as 1800 and Dropout as 0.5 are the Hyperparameters used for training the model [18]. K-fold cross validation is used with K as 10. In order to optimize the loss, Adam optimizer is used. The MLP architecture was designed using Tensorflow library [19]. It is to be noted that fine-tuning of the proposed model is performed only on 1st year data and the obtained Hyperparameters are used for the remaining four-years datasets to predict the bankruptcy.

IV. DATASET AND PREPROCESSING

For this experiment the data of Polish companies analysed over 2002-2012 is used. This dataset was obtained from a popular public dataset repository maintained by University of California, Irvine (UCI) [20]. This data contains five different years of forecasting period. Each dataset consists of 64 different financial indicators which can act as features. There are 2 possible classes representing the bankruptcy status. The class distribution of data in each year is shown in Table 2. This clearly indicates class imbalance.

TABLE II
CLASS DISTRIBUTION

Year	Class 0 – Not Bankrupt	Class 1–Bankrupt
1st year	6756	271
2nd year	9773	400
3rd year	10008	495
4th year	9277	515
5th year	5500	410

Before passing the data into MLP, it first needs to be preprocessed. Firstly, the missing data values are handled by replacing them with the mean of that particular column. Secondly, the data is normalized such that, in each column the mean is 0 and standard deviation is 1. This is done by subtracting the mean from each value in each column and dividing that by the standard deviation of that column. These steps are performed using pandas and scikit-learn libraries [21].

Now to address the class imbalance problem we use sampling algorithms. These algorithms sample the original data in such a way that the distribution of both classes become nearly equal. There are 3 variants of sampling algorithms. Oversampling – increases the instances of minority class, Undersampling – decreases the instances of majority class and a combination of both – undersampling followed by oversampling.

These are implemented using imbalanced-learn library. In Oversampling there are various algorithms like Random Oversampler, Synthetic Minority Oversampling Technique (SMOTE) [10] and Adaptive Synthetic sampling method (ADASYN) [11]. In Undersampling there is Random Undersampler algorithm. As a combination of both SMOTETomek is presented in [12]. Here we experiment and report the results with SMOTE and Random Undersampler algorithms.

SMOTE generates synthetic data instances by considering underlying data instance with its k-nearest neighbours. The new synthetic data instance is generated along the line segment of the underlying data sample and its nearest neighbour by using the method of interpolation. Being a part of the oversampling class it increases the dataset size. Whereas, Random Undersampler decreases the size of dataset in order to balance both the classes. This may result in poor performance if the original dataset size itself is small.

The overall workflow of our methodology can be seen in Fig. 2.

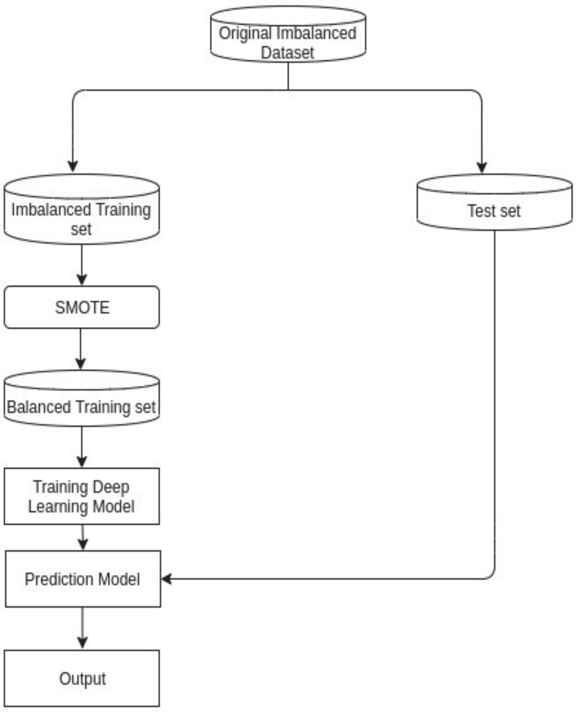

Fig. 2 Overall Flow of Our Methodology

V. Results

We use two metrics to evaluate the model. One is Accuracy and the other one is Area Under Receiver Operator Characteristic curve (AUC). Receiver Operator Characteristic (ROC) curve can be plotted using True Positive Rate and False Positive Rate. These rates are calculate using Sensitivity and Specificity. Both these metrics are calculated using True Positives, True Negatives, False Positives and False Negatives.

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

Where,

$$TP = Number\ of\ True\ Positives$$
$$TN = Number\ of\ True\ Negatives$$
$$FP = Number\ of\ False\ Positives$$
$$FN = Number\ of\ False\ negatives$$

$$Sensitivity = \frac{TP}{TP+FN}$$

$$Specificity = \frac{TN}{TN+FP}$$

$$TP\ Rate = Sensitivity = \frac{TP}{TP+FN}$$

$$FP\ Rate = 1 - Specificity = \frac{FP}{FP+TN}$$

AUC is a measure which determines how good a model can distinguish between classes. AUC of different ROCs can be compared to determine the classification capability of the models. If the AUC is more then the classification capability of the model is more.

All the results reported are using the Hyper parameters mentioned before. We take the mean of all the metrics in the ten-fold cross validation. Mean Accuracy and AUC of our model can be seen in Table 3. The difference in the performance before SMOTE and after SMOTE is evident.

TABLE III
Proposed Model's Accuracy and AUC Before and After SMOTE

Year	**Before SMOTE**		**After SMOTE**	
	AUC	***Accuracy***	***AUC***	***Accuracy***
1st year	0.929	0.989	0.942	0.988
2nd year	0.833	0.969	0.912	0.983
3rd year	0.879	0.978	0.919	0.983
4th year	0.880	0.973	0.917	0.979
5th year	0.931	0.984	0.938	0.984

TABLE IV
PROPOSED MODEL'S ACCURACY AND AUC COMPARISON WITH RU AND SMOTE

Year	With RU		With SMOTE	
	AUC	*Accuracy*	*AUC*	*Accuracy*
1st year	0.607	0.861	0.942	0.988
2nd year	0.579	0.816	0.912	0.983
3rd year	0.600	0.830	0.919	0.983
4th year	0.608	0.829	0.917	0.979
5th year	0.674	0.876	0.938	0.984

TABLE V
PROPOSED MODEL'S ACCURACY AND AUC BEFORE AND AFTER RU

Year	After RU		Before RU	
	AUC	*Accuracy*	*AUC*	*Accuracy*
1st year	0.607	0.861	0.929	0.989
2nd year	0.579	0.816	0.833	0.969
3rd year	0.600	0.830	0.879	0.978
4th year	0.608	0.829	0.880	0.973
5th year	0.674	0.876	0.931	0.984

From Tables 3,4 and 5 it is evident that the model will perform poorly when under sampled with Random Undersampler (RU) in comparison with SMOTE. We even compare the results of our model After RU and before RU in Table 5. In both the cases it is evident that our assumption is true.

TABLE VI
AUC COMPARISON OF PROPOSED MODEL WITH BASELINE MODELS

Model	1st year	2nd year	3rd year	4th year	5th year
LDA	.639	.660	.688	.714	.796
MLP	.543	.514	.548	.596	.699
JRip	.523	.540	.535	.538	.654
CJRip	.745	.774	.804	.799	.778
J48	.717	.653	.701	.691	.761
CJ48	.658	.652	.618	.611	.719
LR	.620	.513	.500	.500	.632
CLR	.704	.671	.714	.724	.821
AB	.916	.850	.861	.885	.925
AC	.916	.849	.859	.886	.928
SVM	.502	.502	.500	.500	.505
CSVM	.578	.517	.614	.615	.716
RF	.851	.842	.831	.848	.898
Our MLP	**.942**	**.912**	**.919**	**.917**	**.938**

In order to prove the worth of the proposed model we have compared our model with the thirteen baseline models as follows. LDA (Linear Discriminant Analysis); MLP (Multi Layer Perceptron); JRip (decision rules inducer); CJRip (cost-sensitive variation of JRip); J48 (decision tree model); CJ48 (cost-sensitive variation of J48); LR (Logistic Regression); CLR (cost-sensitive variation of Logistic Regression); AB (AdaBoost); AC (AdaCost); SVM (Support Vector Machines); CSVM (Cost-sensitive Support Vector Machines); RF (Random Forest) [7]. The comparison is shown in Table 6.

Our model significantly outperforms the baseline models on each of the five datasets.

VI. CONCLUSION

In this paper, we analyzed and examined the role of balancing techniques in hybridization with deep learning for bankruptcy prediction. The whole experimental work is performed over Polish companies data analyzed over 2002-2012. The paper also discussed the overall comparative analysis of the proposed method with thirteen different classifiers LDA, MLP, JRip, CJRip, J48, CJ48, LR, CLR, AB, AC, SVM, CSVM and RF. Fine-tuning of the proposed model is performed only on 1st year data and obtained hyperparameters are used for the remaining four-years datasets to predict the bankruptcy. All experiments were evaluated by 10 × 10-fold cross-validations. It is evident from the results that our model significantly outperforms the compared thirteen models trained on the same dataset in terms of Area Under Receiver Operator Characteristic Curve (AUC). Future work from this paper includes the investigation of other balancing techniques with modified deep learning models. In addition, it will be interesting to investigate the effect of parameter settings of the classifiers and imbalance learning in bankruptcy prediction.

REFERENCES

[1] Zhang, Y., Wang, S., Ji, G., 2013. A rule-based model for bankruptcy prediction based on an improved genetic ant colony algorithm. Mathematical Problems in Engineering 2013.

[2] Altman, E.I., Hotchkiss, E., 2010. Corporate financial distress and bankruptcy: Predict and avoid bankruptcy, analyze and invest in distressed debt. volume 289. John Wiley & Sons.

[3] Beaver, W.H., 1966. Financial ratios as predictors of failure. Journal of accounting research , 71–111.

[4] Altman, E.I., 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The journal of finance 23, 589–609.

[5] Shin, K.S., Lee, T.S., Kim, H.j., 2005. An application of support vector machines in bankruptcy prediction model. Expert Systems with Applications 28, 127–135.

[6] Zhang, G., Hu, M.Y., Patuwo, B.E., Indro, D.C., 1999. Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. European Journal of Operational Research 116, 16–32.

[7] M. Zięba, S. K. Tomczak, and J. M. Tomczak, “Ensemble Boosted Trees with Synthetic Features Generation in Application to Bankruptcy Prediction,” pp. 1–26, 2016.

[8] L. Shi and L. Xi, “Bagging of Artificial Neural Networks for Bankruptcy Prediction,” *2009 Int. Conf. Inf. Financ. Eng.*, pp. 154–156, 2009.

[9] G. P. Naidu and K. Govinda, “Bankruptcy prediction using neural networks,” *2018 2nd Int. Conf. Inven. Syst. Control*, no. Icisc, pp. 248–251, 2018.

[10] N. V Chawla, K. W. Bowyer, and L. O. Hall, “SMOTE : Synthetic Minority Over-sampling Technique,” vol. 16, pp. 321–357, 2002.

[11] H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning,” no. 3, pp. 1322–1328, 2008.

[12] M. C. Monard, “A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data.”, 2004.

[13] S. Karlos and N. Fazakis, “Effectiveness of semi-supervised learning in bankruptcy prediction,” *2016 7th Int. Conf. Information, Intell. Syst. Appl.*, pp. 1–6.

[14] M. Wagle, “Bankruptcy Prediction using Data Mining Techniques,” *2017 8th Int. Conf. Inf. Commun. Technol. Embed. Syst.*, pp. 1–4, 2017.

[15] G. Kumar and S. Roy, “Development of hybrid boosting technique for bankruptcy prediction,” *2016 Int. Conf. Inf. Technol.*, pp. 248–253, 2016.

[16] S. Fan, G. Liu, and Z. Chen, “Anomaly detection methods for bankruptcy prediction,” no. 17, pp. 1456–1460, 2017.

[17] S. Joshi, R. Ramesh, and S. Tahsildar, “Random Forest,” *2018 Second Int. Conf. Intell. Comput. Control Syst.*, no. Iciccs, pp. 1–6, 2018.

[18] G.Hinton, “Dropout : A Simple Way to Prevent Neural Networks from Overfiiting,” vol. 15, pp. 1929–1958, 2014.

[19] A. Agarwal et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” 2015.

[20] A. Frank and A. Asuncion, "UCI Machine Learning Repository [http:// archive. ics. uci. edu/ml]. Irvine, CA: University of California," School of information and computer science, vol. 213, 2010.

[21] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12 2825.