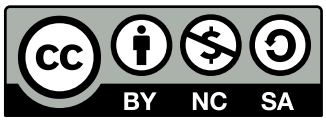
Aalto University
School of Science

Bachelor's Programme in Bachelor's Programme in Science and Technology

Machine Learning in Bankruptcy Prediction: A Literature Review

Tran Quang Anh Tuan

Bachelor's Thesis
2024



Aalto University
School of Science

Author Tran Quang Anh Tuan

Title Machine Learning in Bankruptcy Prediction: A Literature Review

Degree programme Bachelor's Programme in Science and Technology

Major Data Science

Teacher in charge Professor Maarit Korpi-Lagg

Advisor Docent Gozaliasl Ghassem

Date 6 September 2024 **Number of pages** 52+10 **Language** English

Abstract

Substantial research efforts have focused on the topic of bankruptcy prediction. Researchers have analyze bankruptcy and default events using various statistical and machine learning techniques for risk management. Academics have also employed various data sources and processing techniques. In this thesis, various proposed models since 2017 in the literature are evaluated and compared. A literature review is conducted to compare the use of data and processing techniques. The models are then implemented and compared in a uniform testing environment to determine the most optimal method. The data from 2009 to 2021 on US firms from the Compustat database is used for the experiment. Several features sets, both from the literature and from widely used feature selection methods, are generated and applied in the experiment to determine the most suitable set of features for predicting bankruptcy. The experiment results have highlighted several insights in relation to the application of machine learning models in bankruptcy prediction. The models trained on Altman's original features set outperforms those of the the other sets, including recently proposed features sets. This may be due to the diverse set of firms in the experiment, which are from various industries with varying financial conditions. Regarding machine learning models, ensemble methods, random forest, categorical boosting, and gradient boosted decision tree, outperform the other techniques in almost every evaluation metric. Most of the recently proposed methods show lackluster performances compared to previously employed models. The results encourage future research in a more focused manner, which focuses on firms in a single field or scope, to avoid introducing noises and affecting classifying ability. Furthermore, more research on the interpretability of the models would be beneficial to professionals in the field.

Keywords Bankruptcy prediction , financial distress prediction , Machine Learning , review

Contents

Abstract 3

Contents 4

1 Introduction 5

2 Literature review 7

3 Research material and methods 9
3.1 Research methods . . . 9
3.2 Machine learning methods . . . 15
3.2.1 E-SMOTE-ADASVM-TW . . . 15
3.2.2 CatBoost . . . 16
3.2.3 LSEOFOA-KELM and GWO-KELM . . . 17
3.2.4 Gaussian process . . . 19
3.2.5 Clustering-based under-sampling . . . 20
3.3 Experiment . . . 20
3.3.1 Data collection . . . 20
3.3.2 Data preparation . . . 21
3.3.3 Evaluation criteria . . . 22
3.3.4 Experiment design . . . 23

4 Results 25

5 Discussion 43

6 Conclusions 45

References 46

A Appendix 53

1 Introduction

Corporate financial distress or default of corporations can have a significant economic impact. Corporations are characterized by their diverse sources of capitals. Shareholders are those who hold a number of shares in the business, who are subject to possible market value gains and dividend income. In exchange, the companies will receive capital to fund their operations. Another source of capital is from loans. In a loan exchange, the companies will receive capital for their operations, while the lenders, often referred to as creditors, will be paid back the original amount plus interest in a set amount of time. In short, the companies offer possible monetary gains in exchange for immediate capital. Therefore, the performances of these companies, and the resulting ability to pay off obligations, are of serious concern to creditors and shareholders. If these companies are subject to serious financial distresses, the stakeholders risk losing their investments. Furthermore, corporate failures are also of significant importance to the governments. Large companies are often an important part in complex supply chains. Their operational suspension would, therefore, cause shortages that have serious economic consequences. For this reason, the ability to monitor and forecast corporate bankruptcy accurately and efficiently proves essential. Academic researchers, governments, and financial institutions have sought to develop forecasting tools to assist financial professionals in foreseeing crises and making informed decisions.

One of the domains that address prediction tasks is machine learning (ML), particularly supervised machine learning. Supervised ML involves leveraging mathematical models and existing data to predict or label unknown cases. The models are typically trained on labeled existing data. This step ensures that the model can learn the patterns in the data. Next, these models are evaluated on unseen data to test their abilities to generalize to new data. Their performances are analyzed and compared using suitable evaluation metrics, which can highlight their strengths and weaknesses. Therefore, performing ML requires various preparation steps. First, researchers have to propose a suitable model for the prediction task from consulting the literature. Next, researchers need to collect data, perform the necessary processing, and subsequently select the most relevant data and features based on empirical experiment. Finally, they have to choose suitable evaluation metrics to analyze the performance of the model.

There have been various research papers on the application of ML in predicting bankruptcy. In 2017, Barboza, Kimura, and Altman conducted a thorough review of traditional statistical and new machine learning approaches to the prediction of financial distress [1]. Using a large data set, the paper applied the features proposed in previous studies [2] and reviewed methods to conduct a uniform comparative experiment. Since then, researchers have proposed a variety of techniques and models. Most of these proposed methods have been stated to provide improved performance. However, these papers employ different datasets and experiment settings. In addition, these studies conduct unique data processing, feature selection, and imbalance treatment techniques. These varying conditions complicate the comparison process, which causes practitioners to have a hard time deciding which method to employ for optimal performance. Thus, this thesis aims to determine the most optimal method

for predicting bankruptcy. The purpose of this thesis is to assess the effectiveness of recently proposed methods in bankruptcy prediction. The thesis also aims to explore the differences in data processing, feature selection, and model implementations among recent papers in the literature. Therefore, this thesis develops a comparative analysis by conducting a literature review and empirical research on the models and datasets in recent papers.

The thesis is divided into 6 parts. Section 2 discusses the current state of research in the application of ML in bankruptcy prediction. Section 3 presents the process of researching the literature and provides a brief overview of machine learning models and related auxiliary techniques. It also explains the data and presents the process used in empirical analysis. Section 4 presents the results of the analysis. Section 5 discusses the results. Finally, Section 6 discusses the implications of the results, provides future research suggestions, and presents the final thoughts.

2 Literature review

Traditionally, statistical methods have been applied to estimate the solvency. The earliest method used was Beaver's univariate model in 1966 [3]. In 1968, Altman proposed the Z-score model with the application of financial ratios and discriminant analysis [4]. In 1980, logistic regression was applied to predict bankruptcy [5]. In subsequent years, researchers have changed their focus to ML methods such as neural networks (NN) [6, 7, 8, 9], support vector machines (SVM) [10, 11, 12, 13], and decision trees (DT) [14, 15], which provided improved performance. Machine learning methods are also less restricted by statistical assumptions and more effective in fitting non-linear patterns in data. Furthermore, researchers have used ensemble methods and obtained satisfactory results [16, 17, 18, 19]. Notable research interests in applying ensemble methods include XGBoost [20]and AdaBoost [21, 22]. Researchers also utilize various algorithms to optimize ML models, such as genetic algorithms (GA) [23, 24, 25], self-organizing maps (SOM) and k-means [26]. There have been various studies and experiments focusing on regional firms, such as in the US [27], France [28, 29], Greece [30], Hungary [31], Sweden [32], Poland [33], Vietnam [34] and India [35]. The large volume of research studies above shows that there has been a continuous effort in forecasting business bankruptcy, employing a variety of methods and datasets.

Researchers often employ financial ratios to predict bankruptcy. The financial ratios are divided into nine categories. Altman's, profitability, operating capacity and efficiency, solvency and liquidity, structure, cashflow, growth, leverage, and equity. The Altman ratios are the original ratios used in the Altman 1968 [4] paper. Profitability ratios measure the ability of a company to make a profit from its operations. Operating ratios measure operation size and efficiency, which signify operation smoothness. Solvency and liquidity ratios are used to analyze the payment ability of debt and other liabilities. Structure ratios estimate the capital structure. For example, they reveal how a firm is funded, whether it is primarily through loans or equity. Cash flow ratios evaluate the availability of usable cash and other liquid capital. Growth ratios estimate profit, assets, operation growth. Leverage ratios analyze the impact of long-term obligations on overall financial health. Equity ratios are statistics on the trade market. In corporate bankruptcy prediction, data sets are naturally unbalanced. There are usually a much higher number of financially healthy companies compared to the number of bankrupted companies. This has been shown to significantly alter prediction performance [36]. This has led researchers to employ various balancing techniques. A simple solution is to limit the training set so that the classes are balanced. However, this leads to small training sets that may not optimally train the classification models. Many models, such as neural networks, require an adequately sized dataset. Some studies have proposed the creation of synthetic data. The most notable is the use of the synthetic minority oversampling technique (SMOTE) [37, 38]. In 2019, Hosaka also suggested producing weighted average synthetic data [39].

Another problem in predicting bankruptcy is feature selection. Traditionally, features are chosen according to existing studies in finance. Features that have been useful in traditional analysis are also employed in statistical and machine learning models.

However, many machine learning and deep learning models are able to capture information in non-numerical data. Thus, in recent years, more data sources and previously unused features have been taken into account. These include corporate governance measures [40, 41], relational management data [42], and textual financial statements data [43, 44, 45]. In addition, other sources of numerical data have also been shown to be useful as predictors, such as data from the equity market [46]. However, collecting textual data requires expert manual analysis. In addition, corporate governance data lack a reliable and standardized data source. As a result, only financial and equity data are included in the scope of this thesis. Given the large number of useful predictors, practitioners and academics must employ effective feature selection methods to avoid producing high-dimensional problems. Many researchers have proposed filter [47], wrapper [48, 49, 50], and embedded [16] feature selection methods. Other statistical methods, such as the partial dependence plot [51], the normality test and the collinearity test [38], have also been devised.

Research studies on this topic also differ in model evaluation. Although many articles utilize classification accuracy (ACC) for evaluation [37, 52, 53], they also consider other evaluation methods that are better suited for bankruptcy prediction. Firstly, misclassifying a bankrupt case is much more costly than misclassifying a healthy firm. The inability to recognize troubled companies beforehand can cause significant monetary loss to shareholders. However, a healthy company that is classified as potentially bankrupt can reassess its financial performance to avoid negative situations without incurring substantial cost. Furthermore, the bankruptcy prediction problem also has highly unbalanced data, which can affect evaluation results. Therefore, more suitable metrics are suggested, such as true positive rate (TPR), true negative rate (TNR) [38, 39], and customized cost matrix [41]. Some studies also praise the reliability of the receiver operating characteristic (ROC) curve [54, 39] and the area under the ROC curve (AUC) [51, 46] to analyze the trade-off between performance and sensitivity. Matthews correlation coefficient (MCC) [55], which considers all four categories of the confusion matrix, is also applied, due to its high quality of evaluation with respect to unbalanced data.

3 Research material and methods

3.1 Research methods

In 2017, Barboza et al. [1] conducted experiments on techniques derived up until 2016. This thesis thus focuses on reviewing techniques and features proposed in the literature since 2017. The aim of this thesis is on popular papers researching bankruptcy prediction published in reputable journals during the period from 2017 to 2024. This thesis assume a correlation between popularity and good qualities. It is noted that most papers included are from 2017 to 2021 due to the low popularity of more recent papers.

All the initial papers are collected from Google Scholar and Scopus database search engine. Google scholar search engine provides a wider array of initial selections and clearer picture of the research focuses in the community. Google Scholar also provide more relevant search results, which help improving search terms and process. To ensure good quality, papers collected from Google Scholar are double checked on the Scopus database. DOIs, authors' names, published year, citation statistics are collected from Scopus for uniformity and better accuracy.

The initial search term used is "bankruptcy prediction". On further investigation, more search terms are included based on high usage frequency by authors of published articles. These include "bankruptcy forecast", "financial distress forecast", "financial distress prediction". More specified search terms, such as "Machine learning in bankruptcy prediction", "Deep learning in bankruptcy prediction", and "Artificial intelligence in bankrupty prediction", are also used to search for articles focused on method proposal.

The papers extracted are subjected to four main criteria. Firstly, the papers must be published in reputable journals and by reputable publishers. All the papers chosen in the final selection are published by Elsevier and accessible through ScienceDirect. These papers are all indexed in the Scopus database. Secondly, the papers must be cited by more than 50 articles as stated on the Scopus database. The citation process guarantees that the papers have been well peer-reviewed. High citation statistic also indicate promising or valuable discoveries. Thirdly, only papers discussing usage of methods are included in the review and experiment. Papers excluded are review articles, industry-specific case studies. Review articles do not propose any new methods. Industry-specific articles instead usually conduct experiments with variables or methods which are only available or relevant with regards to the industry being studied. It is also noted that this topic follows closely with the applications of machine learning in credit evaluation. However, these studies are excluded to simplify the experiment process. These excluded sources are instead used as researching materials. Lastly, the papers included must be reproducible. This includes having clear method explanation. The methods proposed must be well cited, having clear pseudocode, and firm theoretical background. Finally, the reference list of chosen papers are then studied to collect more relevant papers. These papers are also subjected to the same criteria as the initial papers. The final papers are those with the highest cite number. The search process is shown in Figure 1.

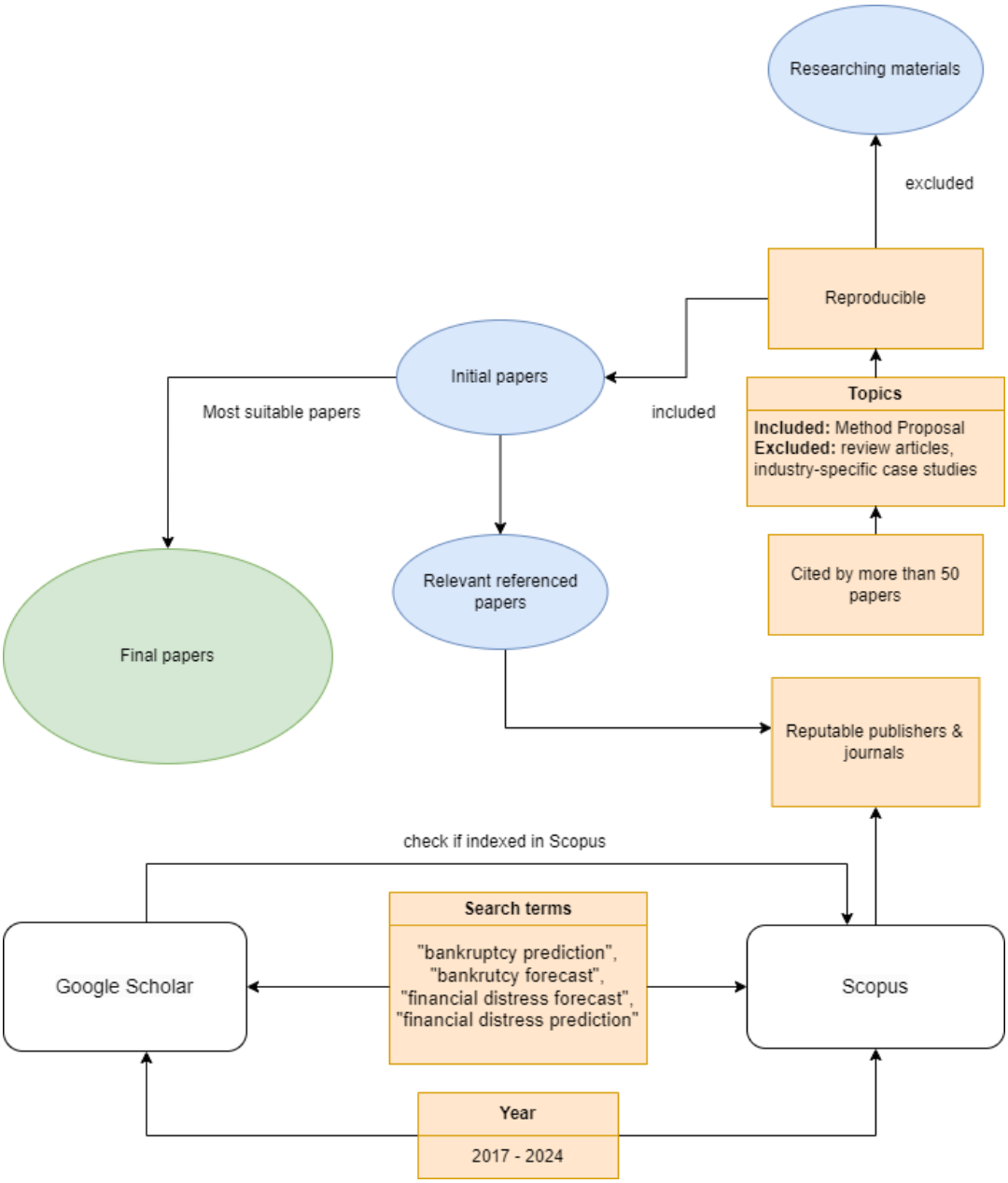


Figure 1: Literature search process

The reviewed papers general summary in Table 1. The methods used are summarized in Table 3. Firstly, it is seen that the datasets range between 150 firms to 2000 firms. However, these papers do not document their datasets clearly, or the cited links are outdated, which lead to difficulties in comparing the datasets. Secondly, it can be seen that the reviewed papers differ substantially in their preprocessing step. Two of the reviewed papers do not implement any preprocessing. Regarding missing data, two of the reviewed papers perform average filling, while others decide to remove the missing data altogether. Only one among the reviewed articles deal with outliers. Most of the

papers either do not address the imbalanced dataset problem, or decide to train on a balanced dataset, or under-sampling. Sun's paper [38] is the only one that utilize SMOTE. Regarding the features used, the reviewed papers utilized varying sets of financial ratios as predictors. The formulas for these ratios are include in table A1. Table A3 shows the types of the ratios. Table A2 shows which ratios are used by each reviewed papers. All of these tables are included in the Appendix, section A.
Table 5 provide a summary of what type of ratios are used by each papers. It can be seen that research papers in this domain favour the use of profitability, solvency, and liquidity ratios. The operation capacity and efficiency, as well as the capital structure, are also of interest. However, growth ratios are rather under utilized. It is argued that growth also plays a crucial role in measuring a company success, or failure [1]. Nevertheless, the level of growth also varies across industries, which makes it a difficult metric to assess. The most popular features, which are used by 3 or more papers, are included in table 7. Again, profitability ratios account for a large proportion of the most used features. Among these features, only interest coverage (IC) and current ratio (CR) are not of the profitability ratios type.

Table 1: Reviewed papers summary

Paper Title	Year Published	Source	Size	Bankrupted size	Bankruptcy criteria	Prediction horizon
Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting [38]	2020	Shanghai, Shenzhen stock exchange 2002 - 2016	2628 firms	438 firms	Negative net profit for 2 years + Net assets less than the registered capital most recent year	1 year
CatBoost model and artificial intelligence techniques for corporate failure prediction [51]	2021	French companies 2014-2016 from Orbis database	Unknown	133 firms	Bankruptcy filing	1-3 years
Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis [55]	2021	Wieslaw and Japanese (JPN-data) dataset	152 + 240 firms	112 + 76 firms	Bankruptcy filing	1-3 years
Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction [56]	2017	Wieslaw [57] and Japanese (JPNdata) dataset	152 + 240 firms	112 + 76 firms	Bankruptcy filing	1-3 years
Probabilistic modeling and visualization for bankruptcy prediction [58]	2017	DIANE, Australian, German, Japanese dataset 2005-2007	2000 firms	667 firms	Bankruptcy filing	1 year
CUS-heterogeneous ensemble-based financial distress prediction for imbalanced dataset with ensemble feature selection [59]	2020	China Stock Market and Accounting Research database (CSMAR) 2017-2018	6898 firm-years	256 firm-years	special treatment	1 year

Table 3: Reviewed papers method summary

Paper	Data processing	Feature selection	Features	Imbalance treatment	Model	Evaluation criteria
Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting [38]	Average filling, Removing outliers using triple SD test	Normality test and mean comparison, stepwise discriminant analysis, colinearity test	19	SMOTE	Embedded SMOTE adaptive boosted support vector machine with time weighting (ESMOTE-ADASVM-TW)	ACC, TPR, TNR, F-measure, G-measure
CatBoost model and artificial intelligence techniques for corporate failure prediction [51]	Replacing categorical with numerical representation	No selection	18	No treatment	Categorical Boosting (CatBoost)	ACC, AUC
Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis [55]	No processing	No selection	30,10	Train on a balanced dataset	Levy flight + slime mould algorithm + elite opposition-based fruit fly optimized kernel extreme learrning machine (LSEOFOA-KELM)	ACC, MCC, sensitivity, specificity
Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction [56]	No processing	No selection	30,10	Train on a balanced dataset	Grey wolf optimized kernel extrreme learning machine (GWO-KELM)	ACC, AUC, Type I, II errors
Probabilistic modeling and visualization for bankruptcy prediction [58]	Removing missing data, log transform, normalization, nominal to discrete integer conversion	No selection	30	Train on a balanced dataset	Gaussian process (GP)	ACC, Type I, Type II errors, F1-score, Precision, recall, ROC
CUS-heterogeneous ensemble-based financial distress prediction for imbalanced dataset with ensemble feature selection [59]	Average filling	Ensemble feature selection	21	No treatment	Gradient boosted decision tree (GDBT)	ACC, AUC, PPV

Table 5: Ratio types used by each papers

Ratio types	Self [38]	Orbis [51]	Wieslaw [55, 56]	DIANE [58]	Self [59]	Total papers
Altman's	y	n	y	n	y	3
Profitability	y	y	y	y	y	5
Operating Capacity and Efficiency	y	y	y	y	n	4
Solvency and Liquidity	y	y	y	y	y	5
Structure	y	n	y	y	y	4
Cash Flow	n	n	y	n	n	1
Growth	y	n	n	n	y	2
Leverage	y	n	n	n	n	1
Equity	y	n	n	n	n	1

Table 7: Most popular ratios by the reviewed papers

Ratios	Number of papers
Net profit margin (NPM)	4
Current ratio (CR)	4
Return on assets (ROA)	3
Return on equity (ROE)	3
Gross margin (GM)	3
Return on capital employed (ROCE)	3
Interest coverage (IC)	3

3.2 Machine learning methods

3.2.1 E-SMOTE-ADASVM-TW

E-SMOTE-ADASVM-TW works by embedding synthetic minority oversampling technique (SMOTE) [60] algorithm for balancing data at every iteration of adaptive boosting (AdaBoost) [61] with weak learners support vector machine (SVM) [62]. The model also incorporates time weighting to combat concept drift [63].
SMOTE is an oversampling technique often used to address imbalanced data classes. SMOTE generates synthetic samples of the minority class to balance the data. SMOTE first identifies the minority class instances. It then proceeds to pick random data points. For each of these data points, SMOTE determines their k nearest neighbors and choose one neighbor randomly. A synthetic example is then created in between the chosen datapoints and their neighbor using formula 1.

$$\mathbf{x}_{\text{synthetic}} = \mathbf{x} + \lambda \cdot (\mathbf{x}_{\text{neighbor}} - \mathbf{x}) \tag{1}$$

In the formula above, λ ranges between 0 and 1. SMOTE generates data points until a predefined condition, which is usually a desired ratio between the number of instances in each classes.
Adaptive boosting (AdaBoost) is a boosting algorithm which combines several learning algorithms, or weak learners, by weighted summing their output to form the final output. AdaBoost trains the weak learners sequentially. For every subsequent learner, the parameters are tweaked so that previously misclassified examples are given more attention. This is done by giving higher weights to misclassified training examples. Each weak learners are also given a coefficient representing their contribution to the final result, depending on their performance while training.
SVM is a supervised machine learning model used in classification and regression analyses. In classification, SVM aims to find the optimal hyperplane for separating data points of different classes. SVM solves the optimization problem:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i$$

subject to:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i$$

where $\mathbf{w}$ is the weight vector, b is the bias term, ξ_i are slack variables, and C is a regularization parameter.
Time weighting is a method for dealing with concept drift in model training. Concept drift is the distribution or concept changes in the data over time. In financial distress prediction, concept drift could be the changing nature of firms in each classes. For example, a bankrupted firm in 2009 will be very different from a bankrupted firm in 2024. Time weighting, thus, weights training examples by the time difference between the target prediction year and the year in which the data becomes available.
In E-SMOTE-ADASVM-TW, the embedded SMOTE works differently in that it generates a predefined number of synthetic data for each real data points. This number

is determined by sample weight given to each samples in AdaBoost training. This helps inflating the presence of misclassified training examples in the balanced dataset, which can further improves training performance on hard-to-classify samples. At each iteration u, E-SMOTE-ADASVM-TW determines the total number of synthetic examples, which the difference between the number of instances in the majority and the minority class. Then, for each real training example, a number of synthetic data points is calculated based on its weight. Further rounding steps are implemented so that the number of synthetic data are integers, and that the numbers sum up to the total calculated earlier. SMOTE is then applied to generate synthetic data. The weak learner is trained on the over-sampled dataset. The model is validated on the original dataset to calculate the error rate e_u. If the model shows sign of overfitting or lower than random guesses accuracy, it is disposed of and retrained. Otherwise, its voting weight is calculated using formula 2 and the same weights are updated using formulas 3 and 4. In formula 3, t^i represents the time difference between the training example i and the target prediction year. After training, E-SMOTE-ADASVM-TW makes prediction based on formula 5. For additional details and pseudo-code of the method, please consult the source paper [38].

$$\alpha_u = 0.5 \ln \left(\frac{1 - e_u}{e_u} \right) \tag{2}$$

$$w_i^{u+1} = w_i^u \exp \left(-\alpha_u l_i^u \exp \left(\lambda t^i l_u^i \right) \right), \quad (i = 1, 2, 3, \ldots, m) \tag{3}$$

$$l_i^u = \begin{cases} 1 & \text{if} \quad f_u(x_i) = y_i \\ -1 & \text{if} \quad f_u(x_i) \neq y_i \end{cases} \tag{4}$$

$$H(x) = \text{sign} \left(\sum_{u=1}^{U} \alpha_u f_u(x) \right) \tag{5}$$

3.2.2 CatBoost

Gradient boosting [64] in an ensemble learning method that sequentially adding weak learners, where each subsequent models are trained on predicting the error of the previous ones. Gradient boosting employs binary decision trees as weak learners. The final prediction is given in 6, where $h(x)$ is the base learner prediction and η_j is the learning rate. Gradient boosting algorithm was applied to bankruptcy prediction in 2020 [59].

$$Z = \sum_{j=1}^{M} \eta_j h(x) \tag{6}$$

CatBoost is a variation of the gradient boosting algorithm, which is proposed in 2018 [65]. CatBoost provides several performance enhancing features compared to other

gradient boosting algorithms. Firstly, it employs ordered boosting, which takes into account the order of the data. This method helps overcoming target leakage [66]. Secondly, CatBoost is designed to handle categorical features. At preprocessing phase, categorical features are replaced with one or more numerical values. Thirdly, CatBoost is shown to have good performance on small datasets [51]. Finally, CatBoost also employs random permuatation of tree structure, with can cope with overfitting. Its output follows formula 7, where R_j is the disjoint region corresponding to the leaves of the tree.

$$H(x_i) = \sum_{j=1}^{J} \eta_j \cdot 1\{x \in R_j\} \tag{7}$$

3.2.3 LSEOFOA-KELM and GWO-KELM

Kernel extreme learning machine (KELM) is a supervised machine learning method proposed in 2012 [67], derived from extreme learning machine (ELM), developed in 2006 [68]. ELM is a form of feedforward neural network with a single hidden layer. The model differs in that the weights and biases in the hidden layer are randomized. In ELM, only the weights and biases of the output layer are optimized, thus reduce the optimization problem to a simple linear problem. The training process is, therefore, significantly faster. KELM, instead, adopts the kernel matrix to define the hidden layer feature mapping. KELM also employs a penalty parameter C for calculating the output weight β as in formula 8. The kernel matrix for KELM is obtained via formula 9 where Gaussian radial basis function $K(u) = \exp\left(-\gamma\|u - v\|^2\right)$ is used. The output function for KELM is shown in formula 10.

$$\beta = H^T \left(\frac{I}{C} + HH^T\right)^{-1} T \tag{8}$$

$$\Omega_{\text{ELM}} = HH^T \text{ where } \Omega^{i,j}_{\text{ELM}} = h(x_i)h(x_j) = K(x_i, x_j) \tag{9}$$

$$f(x) = h(x)H^T \left(\frac{I}{C} + HH^T\right)^{-1} T = \begin{pmatrix} K(x; x_1) \\ K(x; x_2) \\ \vdots \\ K(x; x_N) \end{pmatrix}^T \left(\frac{I}{C} + X_{\text{ELM}}\right)^{-1} T \tag{10}$$

While penalty parameter C control the balance between error minimization and model complexity, parameter γ determines the value of the kernel function. These parameters are, therefore, very important and require leveraging suitable optimization method.
In 2017, swarm method grey wolf optimization (GWO) [69] is proposed for optimizing KELM parameters [56]. This algorithm simulates the hierarchical-based hunting behavior of wolf packs. A population of wolves are first initialized. The positions of each wolves are represented by parameter values. Average accuracy score of KELM cross validated on these parameters are used as fitness measure. The wolf population is then divided into four categories alpha (α), beta (β), delta (δ), and omega (ω). The first three fittest wolves are α), β, and δ. The rest of the wolves are ω. During the

optimizing loop, each of the wolves are updated based on the position of the three best wolves according to formula 11, where C_i is a randomly generated number between 0 and 2 and a_t is a number linearly decreasing from 2 to 0 at each iteration. After each iteration, the three best wolves are updated. At the end, the resulting α wolf position is returned as the best parameters setting.

$$\begin{aligned} D_\alpha &= |C_1 \cdot X_\alpha - X_t| \\ D_\beta &= |C_2 \cdot X_\beta - X_t| \\ D_\delta &= |C_3 \cdot X_\delta - X_t| \\ X_1 &= X_\alpha - A_1 \cdot D_\alpha \\ X_2 &= X_\beta - A_2 \cdot D_\beta \\ X_3 &= X_\delta - A_3 \cdot D_\delta \\ A_i &= 2 \times a_t \times rand(0,1) - a_t \\ X_{t+1} &= \frac{X_1 + X_2 + X_3}{3} \end{aligned} \tag{11}$$

Fruit fly optimization (FOA) algorithm is developed in 2012 [70]. FOA follows the foraging behavior of fruit flies. In the beginning, the positions of the flies are initialized randomly. At each iteration, the flies perform two actions. Firstly, they fly randomly within a defined range. Next, the fitness measures of each positions are determined as the average accuracy of KELM trained on the parameters. The position with the best fitness is determined. Then, the flies start moving towards this position. In 2021, an improved FOA is proposed [55], with the inclusion of Levy flight (LF), slime mould algorithm (SMA) [71], and elite opposition based learning (EBOL) [72]. The proposed model is referred to in brief as LSEOFOA. LF is random walk method in which the steps sizes follow Levy stable distribution. The potential new position of a fly is given by formula 12.

$$X_i' = X_i \times (1 + levy(\beta) \tag{12}$$

The step size of Levy distribution is given in formula 13, where ϕ is given in formula 14, and μ and ν are both drawn from standard normal distribution.

$$levy(\beta) = \frac{(\phi \times \mu)}{|\nu|^\beta} \tag{13}$$

$$\phi = \left(\frac{\Gamma(1+\beta) sin(\pi\beta)/2}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}} \right)^{1/\beta} \tag{14}$$

Fitness of the new position is calculated and compared to the original position. The position of the fly in the next iteration is defined as follows:

$$X_i^{t+1} = \begin{cases} X_i' & \text{if} \quad fitness(X_i') > fitness(X_i) \\ X_i & \text{if} \quad \text{otherwise} \end{cases} \tag{15}$$

SMA is a method for balancing between exploration and exploitation. SMA is inspired by the slime mould foraging behavior, in which its protoplasmic tubes' thickness

changes dynamically to explore and exploit the search space. The location update principle is given as follows:

$$X_i' = \begin{cases} rand \cdot (UB - LB) + LB, & \text{if} \quad rand < z \\ X_{best}(t) + vb \cdot (W \cdot X_A(t) - X_B(t)), & \text{if} \quad r < p \\ vc \cdot X_i & \text{if} \quad r \geq p \end{cases} \tag{16}$$

In formula 16, $rand$ and r are random variables between 0 and 1. UB and LB are upperbound and lowerbound respectively. $X_{best}(t)$ represents the best solution currently found. vb is a parameter with a range between $-a$ and a, where a is given in formula 17. W is the weight of the slime mould, which is given in formula 18, where condition is that the fitness of X_i rank in the first half of the population. $X_A(t)$ and $X_B(t)$ represent two randomly selected individuals from the population. vc decreases linearly from one to zero at each iteration. X_i is the current location of individual i. Parameter $p = tanh(|fitness(X_i) - best|)$. After X_i' is calculated, the next iteration position is also chosen based on formula 15.

$$a = arctanh(-\frac{t}{max_t} + 1) \tag{17}$$

$$W(Rank(fitness(X_i))) = \begin{cases} 1 + r \cdot log(\frac{best - fitness(X_i)}{best - worst} + 1), & condition \\ 1 - r \cdot log(\frac{best - fitness(X_i)}{best - worst} + 1), & otherwise \end{cases} \tag{18}$$

EBOL works by considering the alternative solution on the opposite side to the original one, with regards to the best position. This method ensures that the optimization process would consider diverse set of candidate, further enhancing its ability to find the global optimum. The candidate position is calculated using formula 19.

$$X_i' = rand \times (UB - LB) - X_{best}(t) \tag{19}$$

The position of the fly in the next iteration will be decided by a comparison in fitness between the original and the alternative solution, as in formula 15. It is noted that if the candidate position is outside of bounds, it will be randomized within range following formula 20.

$$X_i' = rand \times (UB - LB) + LB \tag{20}$$

Algorithm LSEOFOA goes through LF, SMA, and EBOL sequentially. The best solution is updated at each steps. After the optimization loop, the best solution is returned.

3.2.4 Gaussian process

In 2017, Gaussian process (GP) is proposed to model bankruptcy [58]. GP is a probabilistic framework used in machine learning for regression and classification analyses. The model is especially successful in modelling complex, non-linear

problems. GP model consists of mean and covariance functions, denoted as $m_f(x)$ and $k_f(x, x')$, where x and x' are input vectors. It is noted that the covariance function is a kernel function. GP assumes that the target variable y is given by $y = f(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $f(x) \sim GP(m_f(x), k_f(x, x'))$. To predict a new data point x_*, we compute the distribution of the latent variable corresponding to the new case [58],

$$p(f_* \mid \mathbf{X}, \mathbf{y}, x_*) = \int p(f_* \mid \mathbf{X}, x_*, f)\, p(f \mid \mathbf{X}, \mathbf{y})\, df \tag{21}$$

where $p(f|\mathbf{X}, y)$ is the posterior distribution over the latent variables. We then use this distribution for f_* to compute:

$$p(y_* = +1 \mid \mathbf{X}, \mathbf{y}, x_*) = \int \delta(f_*) p(f_* \mid \mathbf{X}, \mathbf{y}, x_*)\, df_* \tag{22}$$

where $\delta(.)$ is any sigmoid function. Formula 22 gives the probability of data point with feature vector x_* belonging to the positive class $y_* = +1$.

3.2.5 Clustering-based under-sampling

Clustering-based under-sampling (CUS) is a method of under-sampling data, which serves to balance the data classes. The algorithm was first used in financial domain in 2020 [59]. CUS is motivated by using smaller dataset without losing information. CUS achieves this with the assumption that clustered datapoints share the same information, thus, retaining datapoints from every clusters can help reducing information loss. CUS works by first clustering the data points into predefined k clusters. The algorithm then proceeds to sample randomly a portion p out of each clusters. The retained data are then used for model training and validation. CUS is highly beneficial in improving training time without significant performance loss, especially in an experimental setting.

3.3 Experiment

3.3.1 Data collection

The dataset is collected from Compustat database. The data contains financial and equity statistics for public US firms. Each rows in the dataset is a annual financial report made by a specific company. To gain information about bankruptcy filings, bankruptcy dataset from the UCLA-LoPucki Bankruptcy Research [73] is collected and matched with Compustat data using central index keys (CIK). It is noted that the UCLA-Lopucki dataset only consists of firms with over 100 millions in assets at least 3 years prior to bankruptcy filing. To avoid noisy data, several criteria are implemented to de-noise the data. Firstly, only data rows in which reported assets are over 50 millions dollars are queried. Secondly, only firms with over 100 millions dollars in average assets over the reported period are retained. For bankrupted firms, the average assets is calculated for the period up until 3 years prior to bankruptcy filings for bankrupted firms. For firms which the reported period starts less than three

years prior to bankruptcy filing, they are kept nonetheless. This is motivated by the assumption that large firms closing to bankruptcy may fail to maintain 100 millions dollars in assets. For the whole period, there are 9800 healthy and 716 bankrupted firms. The dataset consists of approximately 160000 firm-year data points. It is noted that financial reports are less likely to show signs of bankruptcy the further it is away from filing year. Therefore, the positive class only consist of data points for which the reported year are less than 4 years away from bankruptcy filing. There also exists firms which have emerged from bankruptcy. All of these firms reports start after bankruptcy filings. As a result, these firms are considered healthy firms.

3.3.2 Data preparation

An initial set consists of all the ratios in table A1 are formed using Compustat financial and equity data. Three other sets are formed including a set of six original Altman's features [4], a set of Barboza's features [1], and a set of most popular features from the reviewed papers. All of the sets are winsorized by three percents on both sides. Min-max normalization is also applied to the datasets according to formula 23.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{23}$$

All missing data points are removed. From the initial set, four other sets are formed using various feature selection methods, including correlation coefficient (CC), mutual information (MI), analysis of variance (ANOVA) [74], and GDBT feature importance (GDBT-FI). CC works by measuring Pearson's correlation coefficient [75] between the explanatory variables and the target variable, which is given by formula:

$$\rho(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y}, \tag{24}$$

where cov is the covariance, σ_X and σ_y are the standard deviations of x and y respectively.

MI is a measure of mutual dependency between two variables. The value is approximated by first discretizing the continuous variables x and y into bins, and using formula 25 from [76].

$$I(X,Y) \approx I_{binned}(X,Y) = \sum_{ij} p(i,j) log \frac{p(i,j)}{p_x(i)p_y(j)}, \tag{25}$$

where $p_x(i) = \int_i dx \mu_x(x)$, $p_y(j) = \int_j dy \mu_y(y)$, and $p(i,j) = \int_i \int_j dx dy \mu(x,y)$, and $\int_i$ means the integral over bin i.

ANOVA consists of several statistical models and estimation procedures used to analyze the difference among means of several variables [74]. Feature selection based on ANOVA is conducted by measuring ANOVA F-value, which measures how much the variance in the target variable can be explained by the explanatory variables. A high F-value indicates that the feature can significantly explain variance in the target variable, and vice versa. For all of the methods above, ten features with the best values

are selected to form datasets.
Finally, a feature set is formed using GDBT features importance. The importance of the features are derived from the contribution of each feature to the predictions of each underlying decision trees. This contribution is determined by the reduction of Gini impurity [77] with the inclusion of the feature. Gini impurity is calculated as follows:

$$\text{Gini impurity} = 1 - \sum_{i}^{C} p_i^2, \tag{26}$$

where C is the number of classes and p_i is the proportion of samples of class i in a specific node of a decision tree. All of the data preparation methods are implemented in Python library 'scikit-learn' [78]. For the experiment, the data up until 2016 are used as the training set, and those after the year 2016 are used as the test set. The feature sets used in the experiment are included in table A4. Tables A5 - A11 show the descriptive statistics of the datasets. All of the above tables are in the Appendix, chapter A. Tables 8 shows the classes distribution of all the sets. It is clear that the data is severely unbalanced. The ratios of the positive class to the negative class are in the range of 0.01. Therefore, CUS is applied to all the training sets. Parameters for CUS are chosen arbitrarily with the number of clusters $k = 100$ and retaining percentage of the majority class $p = 0.1$. The retaining percentage chosen would thus increase the classes ratios to approximately 0.1, reducing the effect of classes imbalance.

Table 8: Data sets class distribution

Feature set	Healthy	Bankrupted
Altman	76465	1377
Barboza	72635	1257
Popular	83022	1624
CC	75579	1370
MI	65102	1248
GDBT-FI	56669	1041
ANOVA	75579	1370

3.3.3 Evaluation criteria

To evaluate performances, several metrics are calculated. Firstly, ACC is calculated, which measures the proportion of correct predictions. Secondly, recall is employed, which measures the ability of a model to recognize all positive instances, which are bankrupted firms, in the dataset. A high recall value means the model is able to

recognize large number of bankrupted firms. This metric is especially crucial, due to the emphasized importance of realizing firms nearing bankruptcy. Thirdly, I also include precision to provide a balanced view of the models performances. Precision measures the ability of a model to return only true positive samples. A high precision value indicates that most of the predicted positive instances are true positive instances. F1 score provide a balanced view between recall and precision. The inclusion of precision and f1 score shows a complete picture of the models performances in acknowledging bankrupted firms. Next, MCC is also employed. MCC is praised for its ability in correctly evaluating models performances in the presence of classes imbalance. MCC takes into account true and false positives and negatives and is generally considered a balanced measure. Last, AUC is included to measure the model trade off between sensitivity and specificity. AUC is the area of the portion under the ROC curve, which is given by the true positve rate (TPR), or recall, and false positive rates (FPR) at different threshold of the prediction probabilities. The formulas for these metrics are given below, where TP, TN, FP, and FN means true positive, true negative, false positive, and false negative respectively.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{27}$$

$$Recall = TPR = \frac{TP}{TP + FN} \tag{28}$$

$$Precision = \frac{TP}{TP + FP} \tag{29}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{30}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{31}$$

$$FPR = \frac{FP}{FP + TN} \tag{32}$$

3.3.4 Experiment design

Of all the machine learning models presented above, GP and GDBT implmentations in Python library 'scikit-learn' are used. For CatBoost model, I employ the 'catboost' library implementation developed by Yandex [1]. The other models are implemented from scratch using the source papers specifications. Most parameters values are taken directly from the source papers. For other parameters which the source papers do not specify, hyper-parameters tuning using grid search is conducted. Furthermore, random forest (RF), multilayer perceptron (MLP), support vector machine (SVM), and logistic regression (LR) are also trained for comparison purpose. For these models, the 'scikit-learn' library implementations are used. The experiment is designed as follows.

[1] Accessed at catboost.ai

For each feature sets, all the models are trained and tested five times. The training set is formed using CUS with different random seeds, and the test set is kept untouched throughout the experiment. The averages of each of the performance metrics among the reruns are then calculated. The experiment design is given in figure 2.

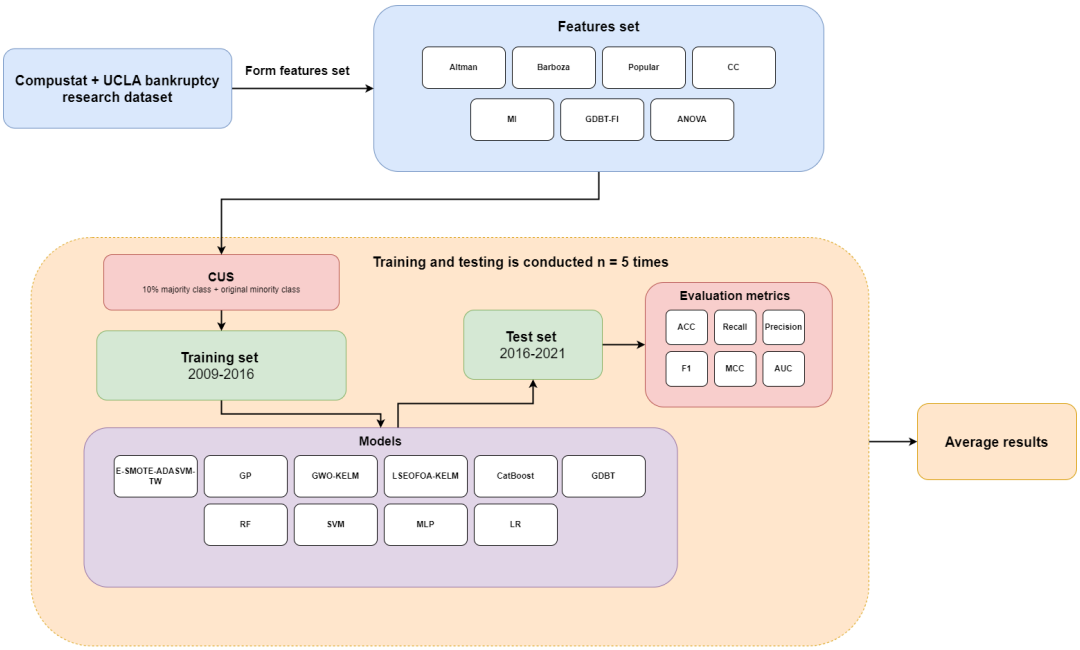


Figure 2: Experiment design

4 Results

Table 10: Results of using the Altman (1968) features

Model	ACC	Recall	Precision	F1	MCC	AUC
E-SMOTE-ADASVM-TW	0.803	**0.787**	0.054	0.100	0.171	0.892
CATBOOST	0.938	0.561	**0.123**	**0.202**	**0.242**	0.906
LSEOFOA-KELM	0.947	0.405	0.112	0.175	0.193	0.888
GWO-KELM	0.944	0.417	0.109	0.172	0.192	0.889
GP	0.942	0.457	0.111	0.179	0.205	0.900
GDBT	0.936	0.561	0.118	0.195	0.237	0.908
RF	0.940	0.534	0.121	0.198	0.234	0.903
MLP	0.933	0.553	0.114	0.188	0.229	**0.909**
SVM	**0.962**	0.193	0.090	0.123	0.114	0.852
LR	0.955	0.207	0.077	0.112	0.106	0.880
Mean	0.930	0.467	0.103	0.165	0.192	0.893

In table 10, average performances of the models while trained and tested using Altman's features set are shown. Overall, the models have varying performances. E-SMOTE-ADASVM-TW has the highest average recall at 0.787, while its ACC, precision, and f1 are the lowest of all models. CatBoost has the best precision, f1, and MCC, at 0.123, 0.202, and 0.242 respectively. Its recall ranks second at 0.561, while its AUC ranks third at 0.906. LSEOFOA-KELM and GWO-KELM are similar in performances in all categories. Their precision, f1, MCC, and AUC all rank below the ensemble methods, MLP, and GP. They do not have the best or worst scores in any metrics. GP has slightly better performance than the KELM models, while not exceptional in any categories. GDBT and RF are two other ensemble methods. Their scores are comparative to those of CatBoost, with their precision, f1, and MCC in the top three. The next model, MLP, has slightly poorer performance compared to the ensemble methods. However, its AUC score ranks first at 0.909. Regarding SVM, this model has the highest ACC score at 0.962, while scoring the lowest in recall and AUC. The performance of LR is close to that of SVM, with slightly better AUC and recall. Otherwise, the model does not excel in any metrics.

Table 12: Results of using the Barboza (2017) features

Model	ACC	Recall	Precision	F1	MCC	AUC
E-SMOTE-ADASVM-TW	0.796	**0.671**	0.042	0.079	0.131	0.849
CATBOOST	0.943	0.496	**0.113**	0.184	0.218	**0.906**
LSEOFOA-KELM	0.953	0.281	0.088	0.134	0.138	0.863
GWO-KELM	0.954	0.269	0.087	0.131	0.133	0.854
GP	0.945	0.331	0.085	0.136	0.147	0.879
GDBT	0.941	0.514	0.113	**0.185**	**0.221**	0.902
RF	0.943	0.477	0.109	0.178	0.209	0.901
MLP	0.929	0.492	0.091	0.152	0.188	0.885
SVM	**0.961**	0.208	0.087	0.122	0.117	0.857
LR	0.954	0.196	0.066	0.099	0.094	0.866
Mean	0.932	0.394	0.088	0.140	0.160	0.876

Table 12 shows the models results when trained and evaluated on the Barboza set. The models perform quite similar using Barboza set compared to Altman set. E-SMOTE-ADASVM-TW again has the best recall at 0.671. This is lower compared to the its score in the first features set. Its ACC, precision, and f1 again ranks last. The second model, CatBoost, boasts the best precision at 0.113, and the best AUC at 0.906. Its f1 and MCC also rank second. LSEOFOA-KELM and GWO-KELM again share similarities in their performances, with GWO-KELM perform slightly worse in recall and AUC. Otherwise, the models do not have noticeable performances in any categories. The next model, GP, has slightly better performance than the KELM models, but is poorer than the ensemble methods and MLP. Ensemble method GDBT has the best performance in f1 and MCC, at 0.185 and 0.221 respectively. Its precision is approximately equal to that of CatBoost, which ranks first. Its AUC ranks second at 0.902. GDBT also ranks second in recall at 0.514, below E-SMOTE-ADASVM-TW. RF, on the other hand, has poorer performance than GDBT in most categories, with the exception of ACC where it scores 0.943, minimally outperforms GDBT at 0.941. MLP, SVM, and LR do not have any noticeable scores, with SVM and LR ranking last in almost metrics. However, SVM again ranks first in ACC at 0.961, similar to that in the Altman set. Overall, the models all perform worse than when they are trained on the Altman features set, with all the best scores across metrics poorer than those of the first set.

Table 14: Results of using the popular features

Model	ACC	Recall	Precision	F1	MCC	AUC
E-SMOTE-ADASVM-TW	0.763	**0.645**	0.039	0.073	0.115	0.801
CATBOOST	0.941	0.323	0.087	0.137	0.145	**0.835**
LSEOFOA-KELM	0.944	0.245	0.074	0.113	0.111	0.791
GWO-KELM	0.947	0.250	0.079	0.120	0.118	0.793
GP	0.944	0.287	0.083	0.129	0.132	0.803
GDBT	0.941	0.345	0.091	**0.144**	**0.155**	0.829
RF	0.941	0.343	**0.091**	0.144	0.154	0.829
MLP	0.942	0.291	0.081	0.127	0.130	0.813
SVM	**0.956**	0.207	0.085	0.121	0.113	0.744
LR	0.939	0.186	0.052	0.081	0.073	0.743
Mean	0.926	0.312	0.076	0.119	0.125	0.798

The next table 14 shows the results of the models when trained on the most popular features set. According to the table , these models have notably poorer performances compared to models trained on the Altman and Barboza features, with significantly lower scores in all categories. SVM again has the best ACC at 0.956, while the best in recall score is from E-SMOTE-ADASVM-TW at 0.645. The ensemble methods are again the best performers in precision, f1, MCC, and AUC. In particular, GDBT leads in f1 and MCC at 0.144 and 0.155 respectively. RF lead in precision at approximately 0.091, while CatBoost ranks first in AUC at 0.835. The KELM models, LSEOFOA-KELM and GWO-KELM, have approximately similar performances, failing to distinguish themselves compared to the other models in any metrics. The next model, MLP, performs significantly worse when trained on this feature set compared to previously. The model ranks fifth in precision, f1, MCC, and AUC, below GP, despite outperforming this model in the previous two sets.

Table 16: Results of using the CC features

Model	ACC	Recall	Precision	F1	MCC	AUC
E-SMOTE-ADASVM-TW	0.824	**0.634**	0.049	0.091	0.141	0.847
CATBOOST	0.930	0.408	0.084	0.139	0.161	0.858
LSEOFOA-KELM	0.942	0.395	**0.099**	**0.158**	**0.176**	0.834
GWO-KELM	0.940	0.396	0.097	0.155	0.174	0.836
GP	0.934	0.423	0.091	0.150	0.174	0.858
GDBT	0.927	0.431	0.084	0.141	0.166	**0.86**
RF	0.932	0.399	0.085	0.139	0.160	0.857
MLP	0.924	0.431	0.082	0.137	0.163	0.858
SVM	**0.944**	0.377	0.099	0.157	0.172	0.761
LR	0.938	0.391	0.092	0.149	0.167	0.827
Mean	0.923	0.428	0.086	0.142	0.165	0.840

Regarding models trained on the CC features set, their average scores are similar to those of the models trained on the Barboza features. According to table 16 SVM again rank first in ACC at 0.944, while E-SMOTE-ADASVM-TW has the best performance in recall at 0.634. However, the rankings in precision, f1, and MCC are different. Although the average precision of the models are similar to that in the Barboza set, the individual models performances are vastly different. The best models regarding precision in the CC set, LSEOFOA-KELM and SVM, perform notably poorer than the best model in the Barboza set, GDBT, at around 0.099 and 0.113 respectively. GWO-KELM ranks third with a score of 0.097. LR ranks fourth with a score of 0.092. The ensemble methods all have scores at approximately 0.084, while E-SMOTE-ADASVM-TW ranks last at 0.049. Besides precision, LSEOFOA-KELM also has the best performance in f1 with a score of 0.158 and MCC with a score of 0.176. However, it is noted that these scores are much lower than those of the best models in the Barboza set. It is, thus, apparent that it is not LSEOFOA-KELM that has performed better, but the ensemble methods that have performed significantly worse using the CC set. However, the ensemble models have the best trade off with leading AUC at 0.86, 0.858, and 0.857 for GDBT, CatBoost, and RF respectively.

Table 18: Results of using the MI features

Model	ACC	Recall	Precision	F1	MCC	AUC
E-SMOTE-ADASVM-TW	0.823	**0.737**	0.060	0.111	0.176	0.888
CATBOOST	0.933	0.563	**0.12**	**0.198**	**0.238**	0.897
LSEOFOA-KELM	0.939	0.469	0.115	0.185	0.210	0.879
GWO-KELM	0.934	0.492	0.110	0.180	0.210	0.874
GP	0.932	0.529	0.113	0.186	0.222	0.890
GDBT	0.931	0.539	0.114	0.188	0.226	**0.9**
RF	0.937	0.505	0.118	0.192	0.223	0.893
MLP	0.931	0.558	0.117	0.193	0.233	0.898
SVM	0.939	0.448	0.110	0.177	0.200	0.848
LR	**0.948**	0.380	0.115	0.176	0.188	0.866
Mean	0.925	0.522	0.109	0.179	0.213	0.883

Regarding models trained on the MI features, the performances are more promising than those of the last two sets. LR leads in ACC with a score of 0.948, with models average of 0.925, while E-SMOTE-ADASVM-TW leads in recall at 0.737, with models average of 0.522. The two KELM models again have similar performances across categories. Regarding the other metrics, performances are again distributed similar to those in the Altman set, with the ensemble methods and MLP leading in precision, f1, MCC, and AUC. The precision scores of most models are between 0.11 and 0.12, while E-SMOTE-ADASVM-TW performs significantly worse at 0.06. CatBoost achieves the best f1 score at 0.198, outperforms the second best model, MLP, which boasts a score of 0.193. CatBoost also leads in MCC with a score of 0.238. Regarding AUC, GDBT obtains the best score of 0.9, with models average at

approximately 0.883. Most models perform above the range of 0.866, while SVM have the lowest score of 0.848.

Table 20: Results of using the GDBT-FI features

Model	ACC	Recall	Precision	F1	MCC	AUC
E-SMOTE-ADASVM-TW	0.845	**0.69**	0.065	0.119	0.179	0.884
CATBOOST	0.935	0.511	0.119	0.193	0.224	0.899
LSEOFOA-KELM	0.942	0.421	0.114	0.179	0.197	0.859
GWO-KELM	0.943	0.406	0.113	0.177	0.192	0.861
GP	0.939	0.482	0.120	0.192	0.218	0.886
GDBT	0.931	0.520	0.113	0.185	0.219	**0.899**
RF	0.939	0.506	**0.124**	**0.199**	0.229	0.899
MLP	0.930	0.553	0.119	0.194	**0.232**	0.896
SVM	**0.945**	0.407	0.117	0.182	0.197	0.843
LR	0.943	0.369	0.105	0.164	0.175	0.856
Mean	0.929	0.487	0.111	0.178	0.206	0.878

The next test results shown in table 20 focus on the application of the most important features as chosen by GDBT. LSEOFOA-KELM and GWO-KELM are similar in their performances across categories. SVM has the best ACC at 0.945, while best recall is from E-SMOTE-ADASVM-TW with a score of 0.69. The ensemble methods and MLP boast the best scores regarding precision, f1, MCC, and AUC. The models average performances are comparable to those of the models trained on MI features across metrics, with the exception of recall, in which models trained on MI features have an average score of 0.522 compared to 0.487 for GDBT-FI features.

Table 22: Results of using the ANOVA features

Model	ACC	Recall	Precision	F1	MCC	AUC
E-SMOTE-ADASVM-TW	0.830	**0.601**	0.049	0.090	0.135	0.839
CATBOOST	0.932	0.401	0.085	0.140	0.161	0.854
LSEOFOA-KELM	0.940	0.397	0.097	0.156	0.174	0.842
GWO-KELM	0.938	0.398	0.094	0.151	0.171	0.839
GP	0.935	0.407	0.090	0.147	0.168	0.855
GDBT	0.929	0.426	0.085	0.142	0.167	**0.856**
RF	0.934	0.367	0.082	0.134	0.149	0.849
MLP	0.927	0.415	0.083	0.138	0.161	0.853
SVM	**0.944**	0.379	**0.1**	**0.159**	**0.174**	0.772
LR	0.939	0.388	0.093	0.150	0.167	0.826
Mean	0.925	0.418	0.086	0.141	0.163	0.839

The test results for models trained on ANOVA features are shown in table 22. For this test, SVM ranks first in ACC, precision, f1, and MCC scores, with scores of 0.1, 0.159,

and 0.174 for each metrics respectively. E-SMOTE-ADASVM-TW achieves the best recall at 0.601, while GDBT achieves the best AUC score of 0.856. The ensemble methods have the worst performances in this set compared to all other sets. All the other models also perform significantly worse. Overall, the results are poorer than those of the previous sets, with the best scores highlighted and the models average being much lower across metrics.

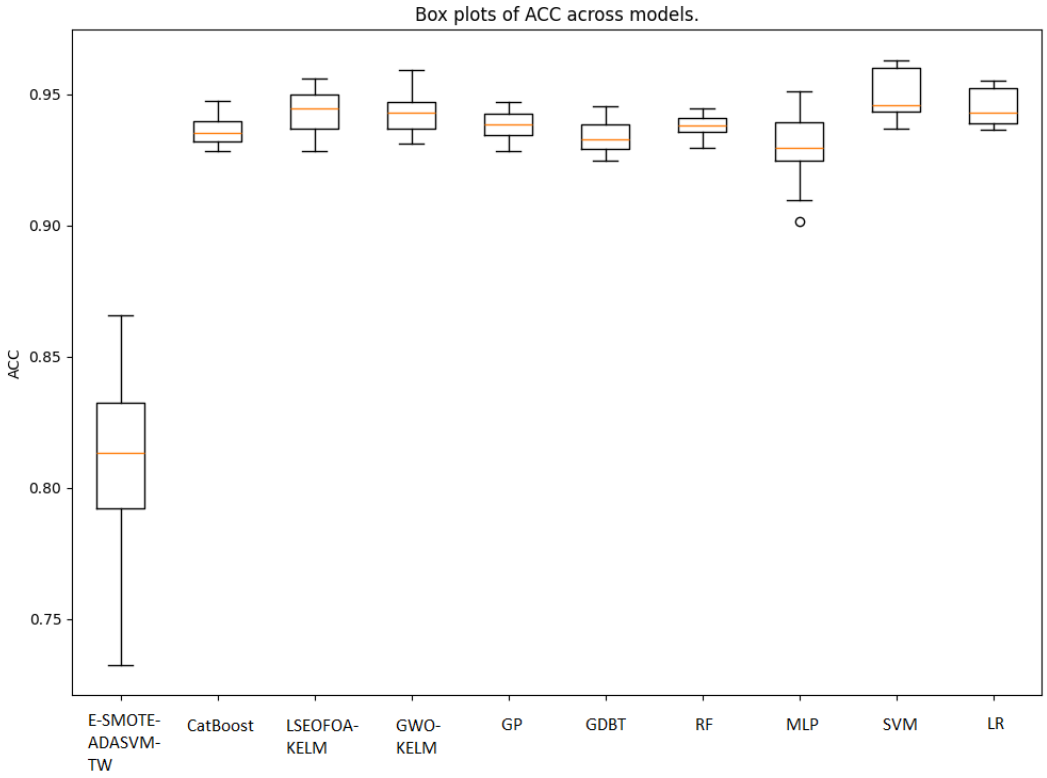


Figure 3: ACC box plots

Next, the consistencies of the models are analyzed using the box plots in figures 3 - 8, which show the min-max range, inter-quartile range, and the median of the models performances across of the tests. The results of training and testing on different datasets are all included to form the box plots. Figure 3 depicts the ACC scores distribution. According to this figure, E-SMOTE-ADASVM-TW has the worst overall ACC, while also having the most variation in its performances. Its ACC scores range between 0.73 and 0.87, and inter-quartile range from 0.78 to 0.83. All the other models' performances are always above 0.9. CatBoost, GP, GDBT, and RF have high consistencies in ACC, with the difference between maximum and minimum rather minimal. LSEOFOA-KELM and GWO-KELM have slightly more variation than the previously mentioned models, with maximum reaching above 0.95. The MLP model has the second largest variation range in its performances, from roughly 0.91 to 0.95. The model also has an outlying performance at roughly 0.9. SVM has the best overall ACC scores, which range from a minimum of 0.94 to a maximum of 0.97.

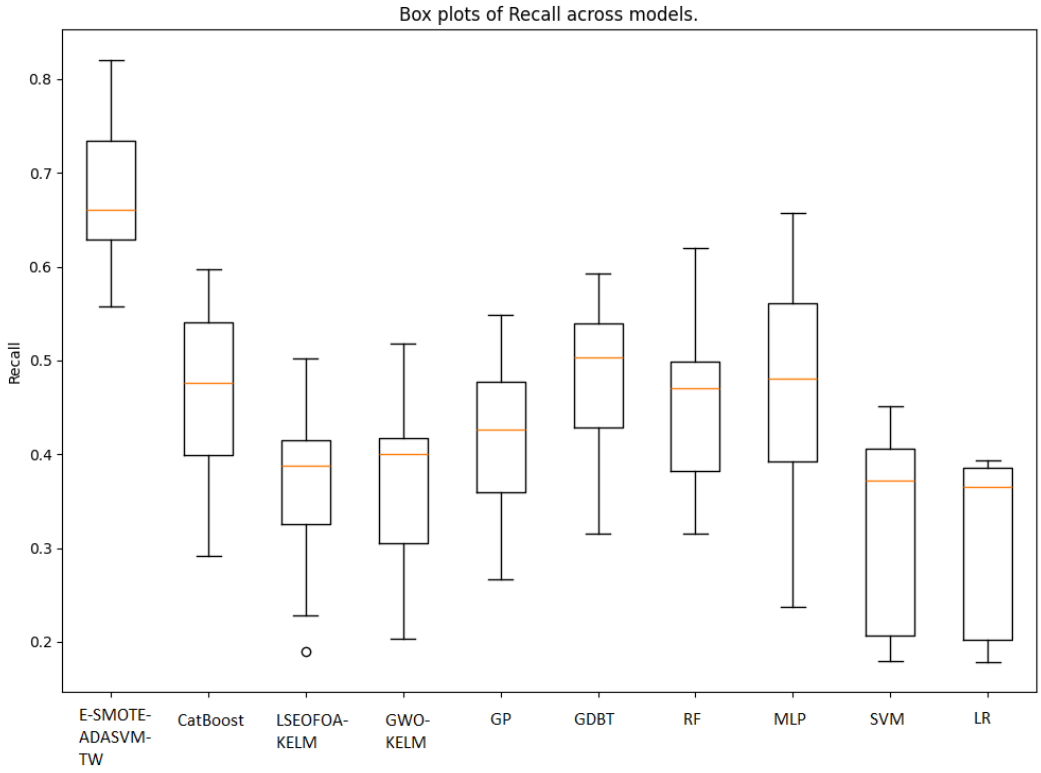


Figure 4: Recall box plots

Figure 4 shows the results distribution for the recall metric. E-SMOTE-ADASVM-TW has overall much higher recall than the other models, with inter-quartile range between 0.56 and 0.81. The ensemble methods and MLP have quite similar range. However, the performances of MLP vary widely, ranging from approximately 0.25 to 0.65, while the performances of ensemble methods only vary in the range between 0.3 and 0.6. The KELM models have very similar boxes shapes, with median at around 0.4, and variation between 0.22 and 0.52. The recall scores of SVM and LR vary in a lower range compared to other models, between a %25 quartile value of 0.21 and %75 quartile value of 0.4 . It is noted that, overall, recall scores vary more widely than the ACC score.

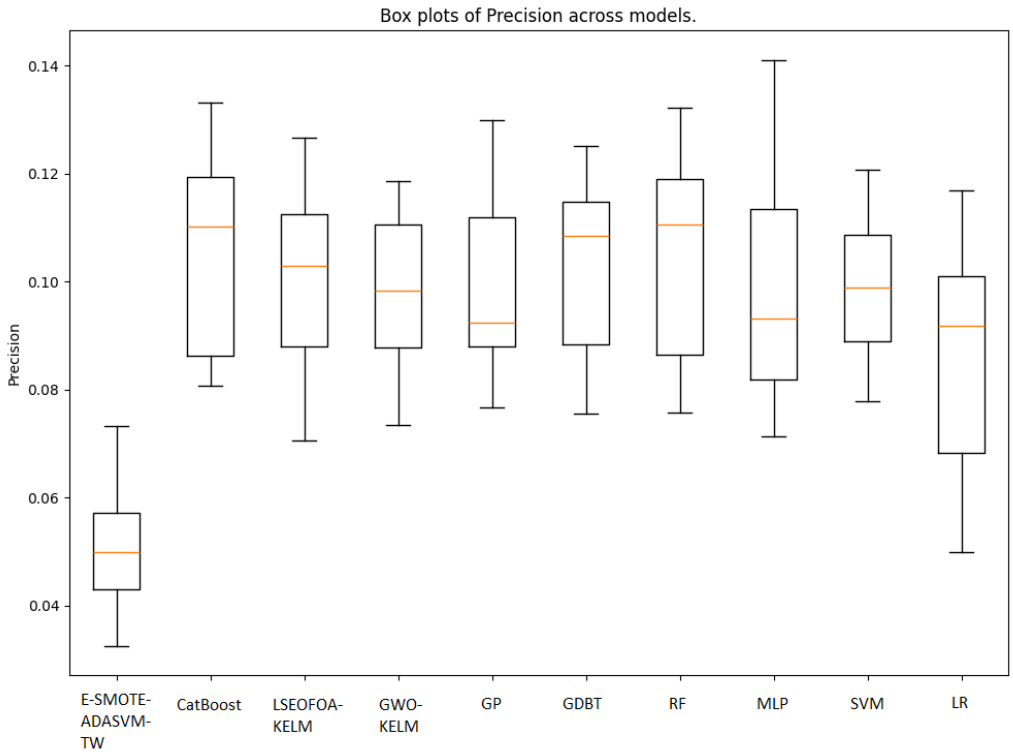


Figure 5: Precision box plots

Next, figure 5 presents the precision distributions of the models. Overall, the precision scores are relatively poor. E-SMOTE-ADASVM-TW has the lowest precision median, at approximately 0.05. Its performances range between 0.035 and 0.075. Meanwhile, most of the other models have similar range. With the exception of LR and E-SMOTE-ADASVM-TW, the models all have inter-quartile ranges fall roughly between 0.087 and 0.12. However, MLP has noticeably longer higher whisker with the maximum value of above 0.14, compared to roughly below 0.13 by the other models. Furthermore, their medians vary. The ensemble methods have the highest medians at roughly 0.11, while the lowest median is from GP and MLP at 0.09 LR, although having similar median to GP, has an extended lower whisker with the 25% quartile value at below 0.08 and minimum of 0.05.

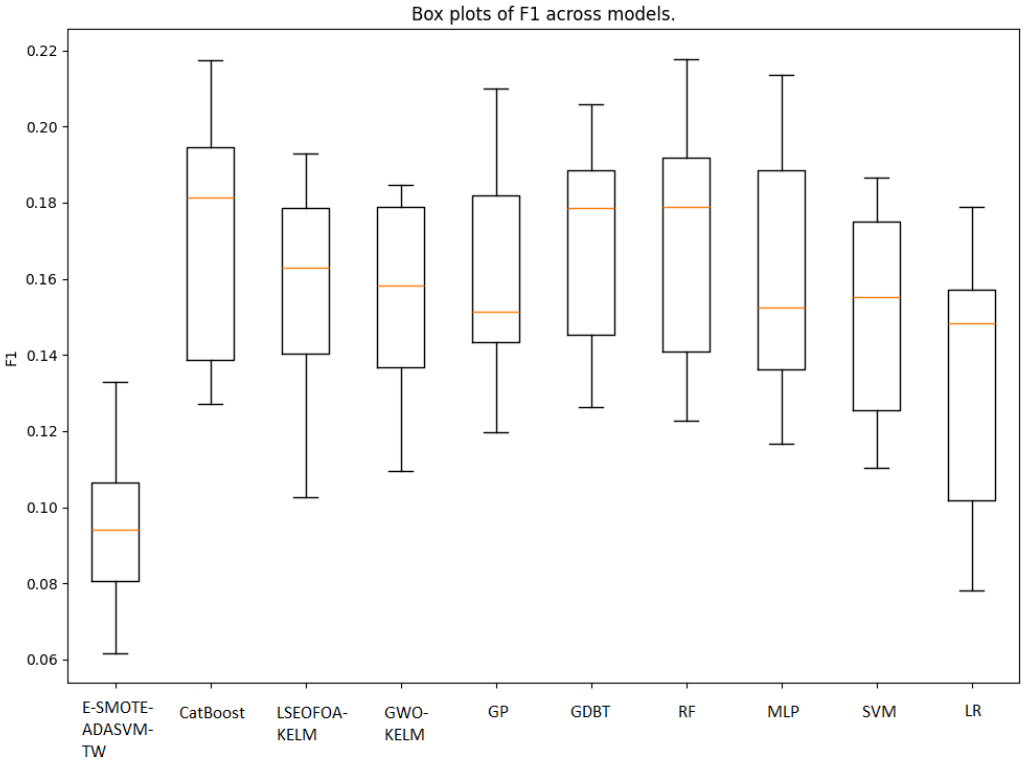


Figure 6: F1 box plots

Figure 6 presents the F1 scores distribution of the models. CatBoost and RF have the widest inter-quartile range, between 0.14 and 0.19, with the upper quartile reaching the highest f1 score of near 0.22. GDBT also have relatively high median f1 score. However, its maximum only reaches below 0.21. MLP has a slightly lower range compared to the CatBoost and RF, but a much lower median, at around 0.15. The KELM models show more consistent performances with a shorter inter-quartile range. Their ranges are, however, at a lower score range than the previously mentioned methods. E-SMOTE-ADASVM-TW displays the lowest f1 score, with both a low median and a small inter-quartile range, indicating consistently poor performances. LR has better median score than E-SMOTE-ADASVM-TW, but with a broader range, which reaches a minimum of 0.08.

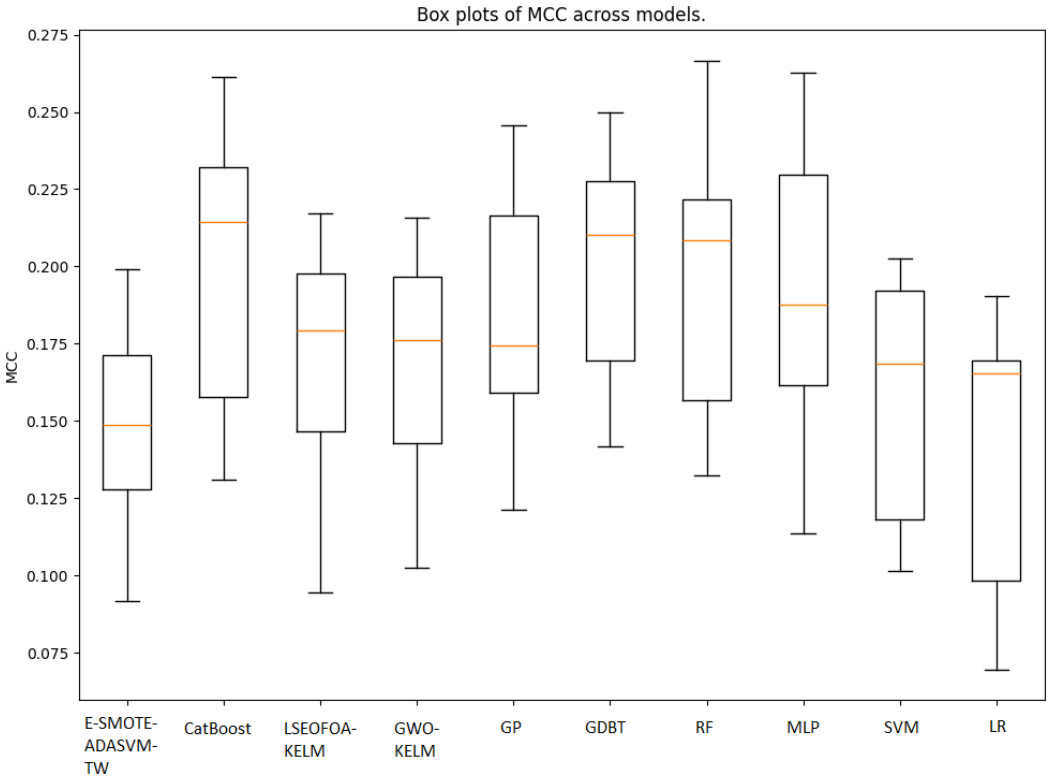


Figure 7: MCC box plots

Figure 7 presents the MCC scores distribution of the models. CatBoost, GDBT, and RF show the highest MCC values, with the highest median and a broad inter-quartile range. However, GDBT has a shorter min-max range, which indicate more consistent performances. MLP, although having similar inter-quartile range to the ensemble methods, has a much wider min-max range and lower median, which indicate that MLP is less consistent, with more of its performances on the lower side. LSEOFOA-KELM and GWO-KELM again show similar performances, with roughly equal median, inter-quartile range, and min-max range. E-SMOTE-ADASVM-TW has the lowest median score of all models. But LR has a much wider range, with a minimum reaching 0.075, the lowest of any recorded performances.

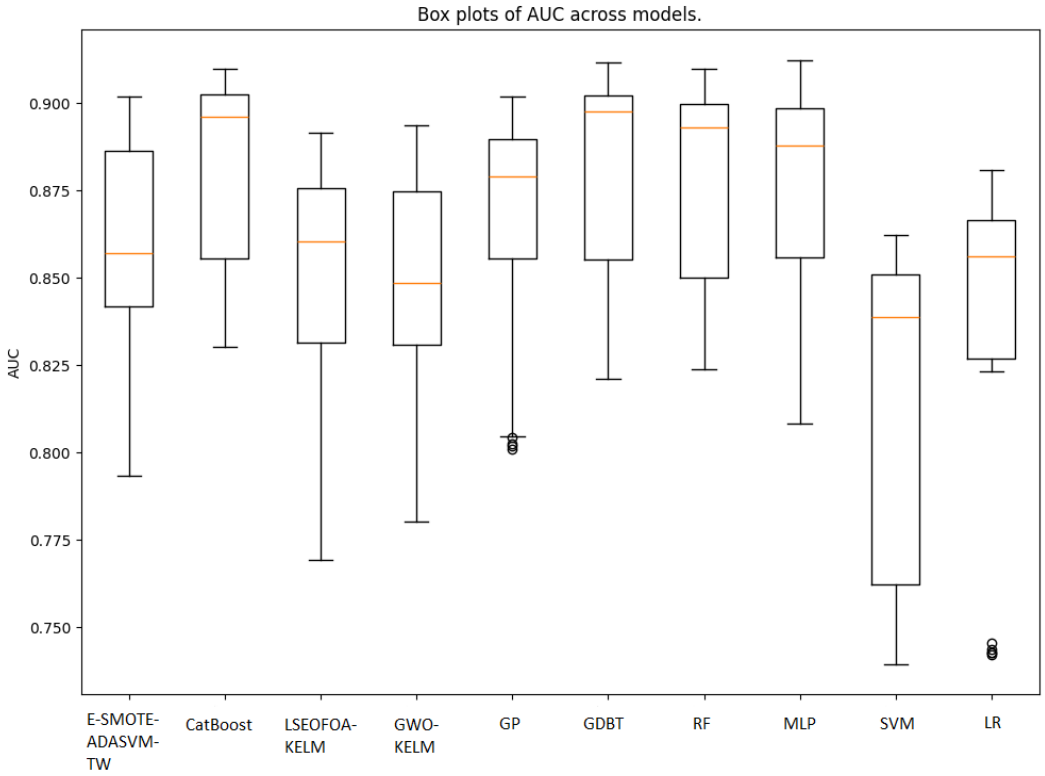


Figure 8: AUC box plots

Figure 8 presents the AUC scores distribution of the models. The highest median scores are from CatBoost, GDBT, and RF. These models also share similar inter-quartile and min-max range. MLP has a slightly lower median and much wider range, with the lower whisker extends further below. This again indicates that MLP is less consistent in its performances compared to the ensemble methods. SVM has the lowest median score and the widest inter-quartile range, which shows that this model has poor performances and lack consistency.

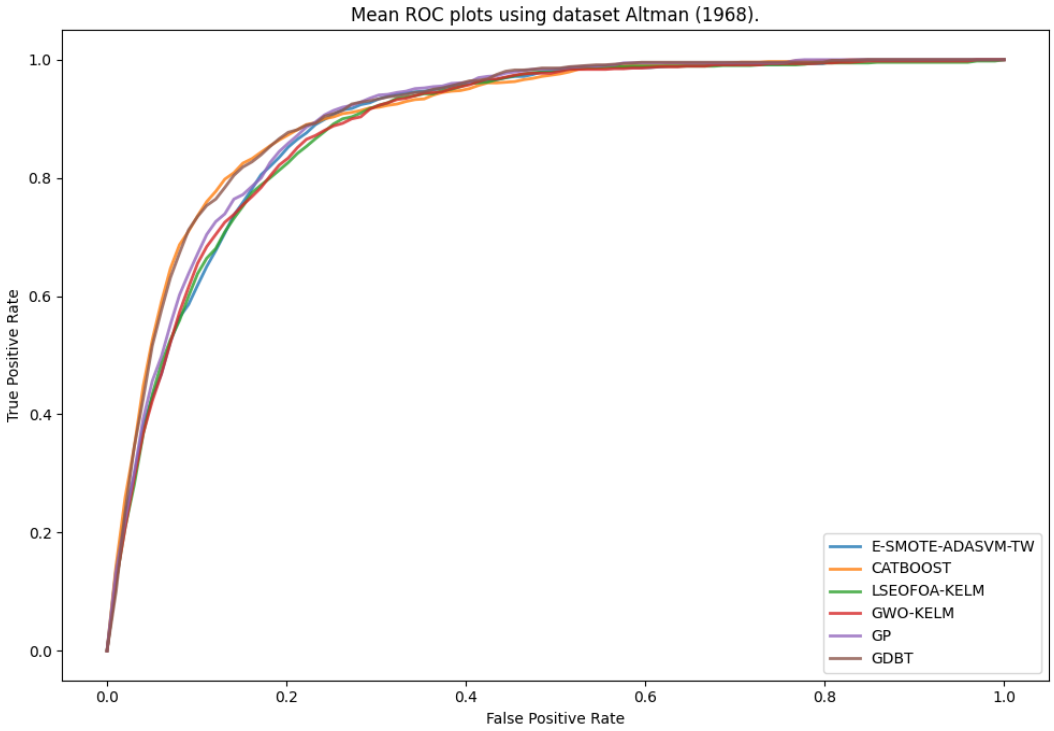


Figure 9: ROC plot using Altman features

Next, the ROC curves in figures 9 - 15 are analyzed. The average ROC for every model after five reruns are calculated for each datasets. For this part, only the models from the reviewed papers are included to avoid clustered plots. Figure 9 shows the average ROC of each models when trained and evaluated on the Altman features set. The models performances are mostly similar for points above the false positive rate of 0.3. For the area below that point, it is clear that CatBoost and GDBT have better curves than the other models. Their ROC curves tend to be closer to the top left corner, indicating better overall performance in balancing sensitivity and specificity. At higher false positive rates, between 0.3 and 0.4, E-SMOTE-ADASVM-TW and GP show very competitive performances compared to the ensemble methods. Overall, LSEOFOA-KELM and GWO-KELM are the worst performers with their curves furthest from the top left corner.

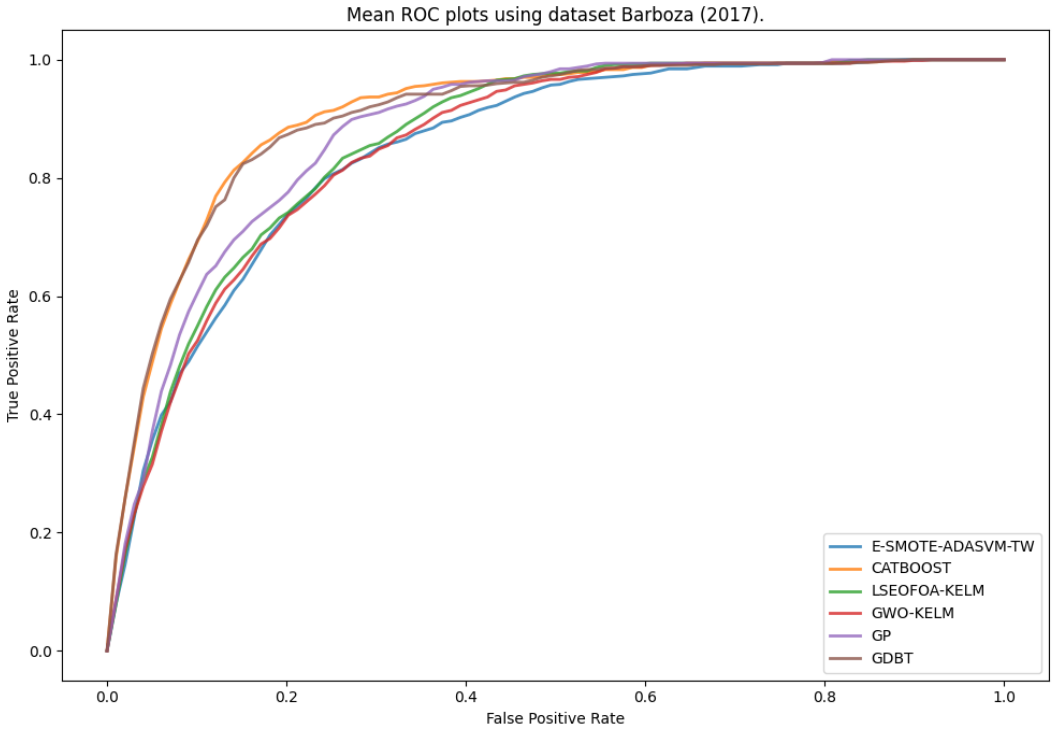


Figure 10: ROC plot using Barboza features

Next, figure 10 shows the average ROC of each models when trained and evaluated on the Barboza features set. Overall, there are larger differences between the curves of the models. CatBoost and GDBT again show the best performances, with their curves closer to the top left corner. GP ranks third with its curve below the ensemble models. The worst performers are LSEOFOA-KELM, GWO-KELM, and E-SMOTE-ADASVM-TW.

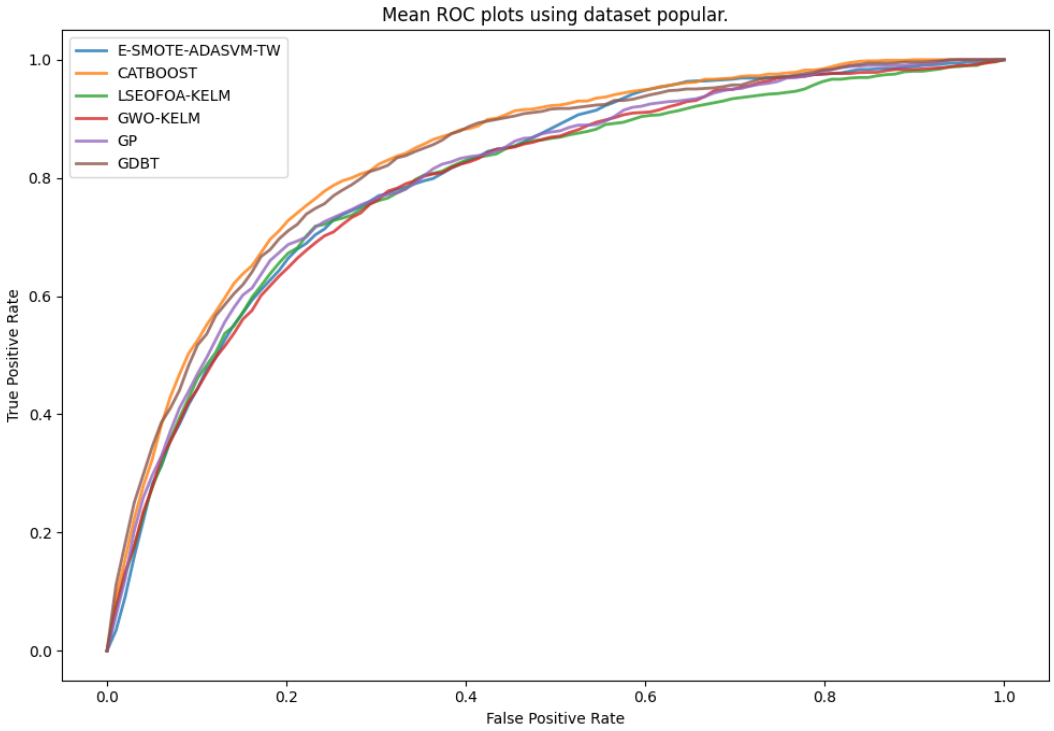


Figure 11: ROC plot using the popular features

Figure 11 shows the average ROC of each models when trained and evaluated on the popular features set. Overall, GDBT and CatBoost show the best performances with the highest curves. The other models have rather similar performances, with their curve indistinguishable from each other.

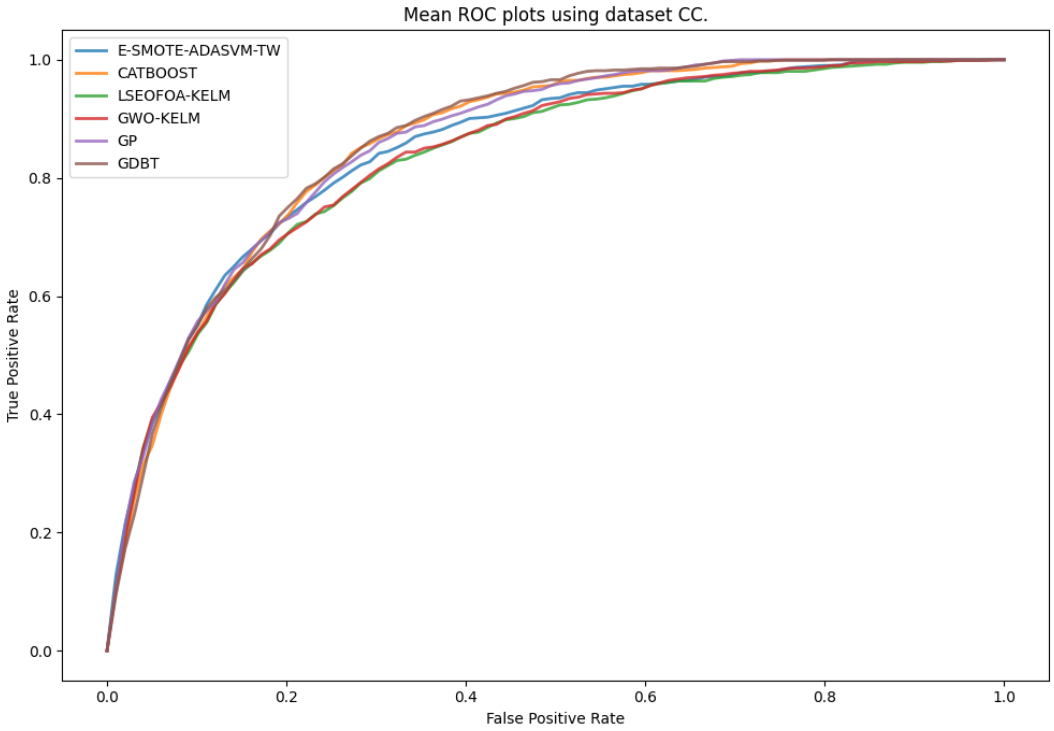


Figure 12: ROC plot using the CC features

Figure 12 shows the average ROC of each models when trained and evaluated on the CC features set. Models with highest curves are CatBoost, GDBT, and GP, which have very similar curves. E-SMOTE-ADASVM-TW performs slightly better than the KELM models. The KELM models have indistinguishable curves.

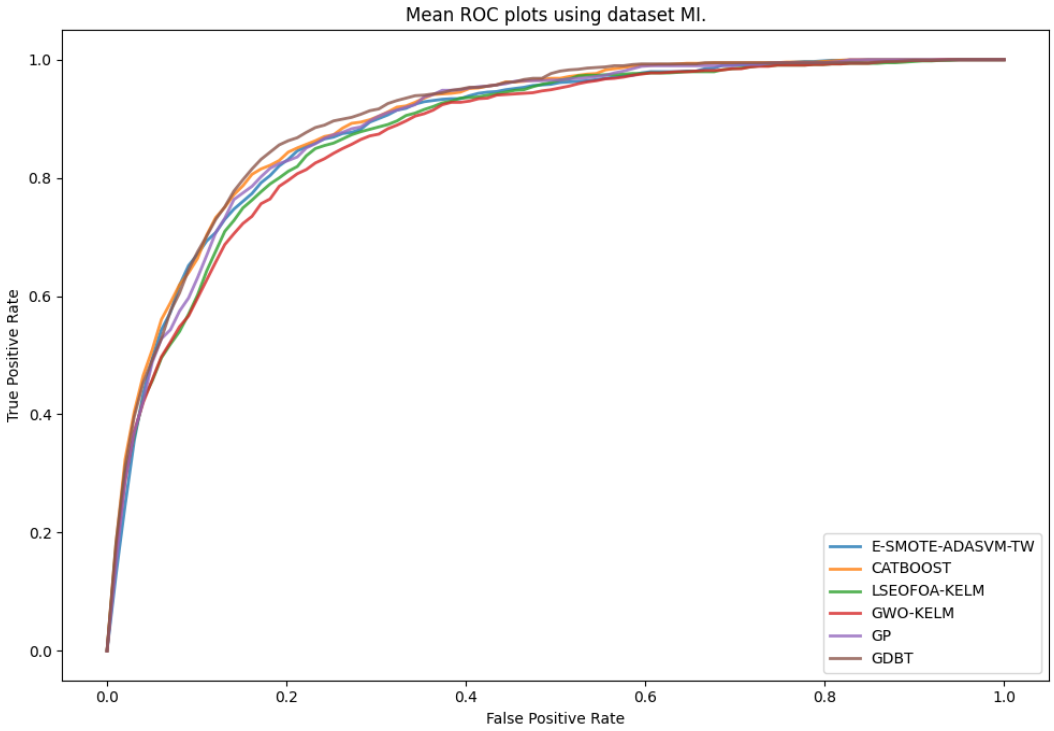


Figure 13: ROC plot using MI features

Figure 13 shows the average ROC of each models when trained and evaluated on the MI features set. The curves of the models are overall close in this set. GDBT is noticeably better in the false positive rate range between 0.2 and 0.4. On the other hand, GWO-KELM also appears to have lower performance compared to the other models in the same false positive range. It can be concluded that GDBT performs better than the other models in this set, althought the difference is not significant.

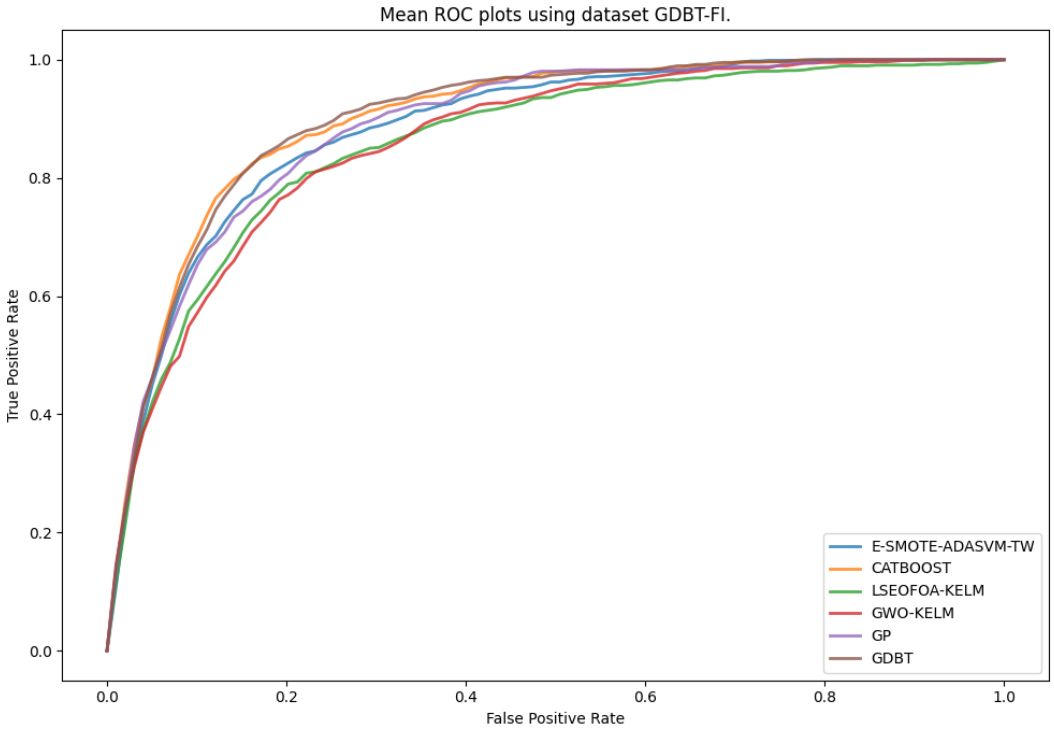


Figure 14: ROC plot using GDBT-FI features

Figure 14 shows the average ROC of each models when trained and evaluated on the GDBT-FI features set. Overall, the models are divided into three pairs with similar curves. CatBoost and GDBT have the best performances with the highest curves. E-SMOTE-ADASVM-TW and GP have similar performances, which are below those of the ensemble models. The KELM models are the worst performers with the lowest curves.

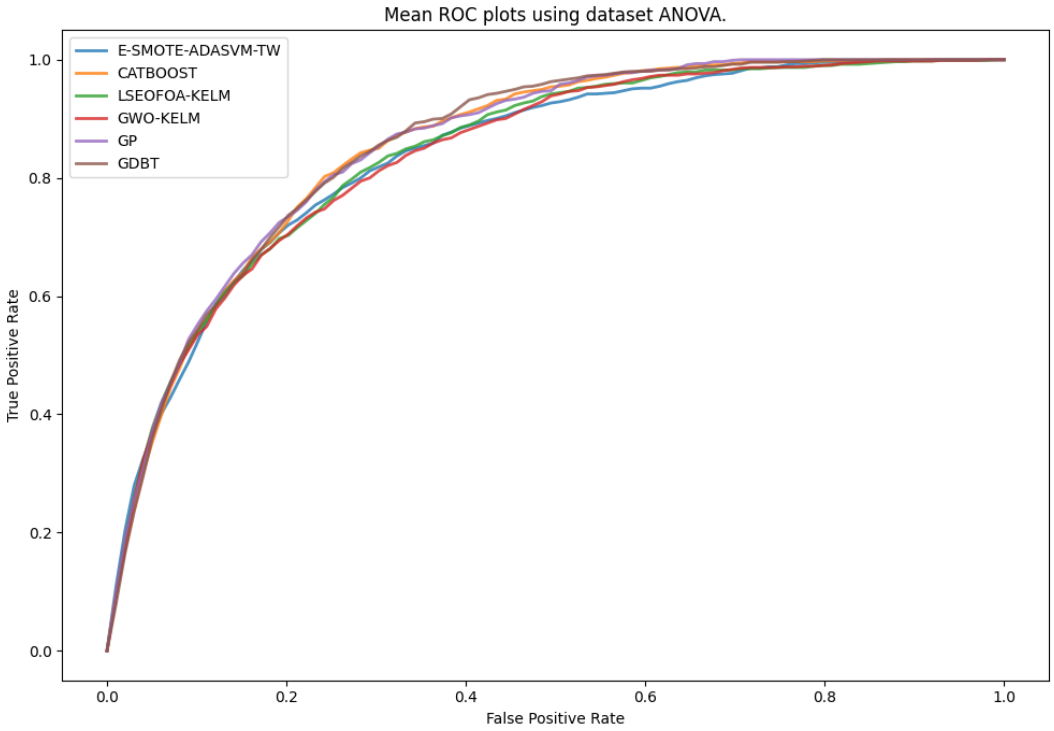


Figure 15: ROC plot using ANOVA features

Lastly, figure 15 shows the average ROC of each models when trained and evaluated on the ANOVA features set. Overall, the ROC curves are rather similar in the ANOVA set. However, CatBoost, GDBT, and GP perform slightly better than the other three models.

5 Discussion

Regarding the average results, SVM performs the best on ACC across features sets. However, ACC is a flawed metric in the presence of classes imbalance. The test set is severely unbalanced, which causes the ACC score to be biased towards correct predictions of the majority class. Regarding recall, E-SMOTE-ADASVM-TW shows dominant performances. This model is more likely to classify examples as belonging to the positive class. This may have been caused by the use of SMOTE. However, this model has the poorest performance in precision, f1, and MCC. It is clear that although E-SMOTE-ADASVM-TW recognize the highest number of true positive examples, it does so at the cost of making a large number of incorrect prediction. This is also a trait of SMOTE. The synthetic data often lower the ability to distinguish between the classes. LSEOFOA-KELM and GWO-KELM perform similarly on all the tests, despite the difference in optimization algorithm. MLP has good performance, but is mostly below those of the ensemble methods. SVM and LR are the worst performers in almost all categories, with the exception of ACC. Meanwhile, the ensemble methods always perform better than the other models in precision, f1, MCC, and AUC. It is clear that the ensemble methods are the most promising in their performances. These models have adequate results in recall, while still maintain better precision than the other models. Their f1 scores are the highest, which show that they are the most capable models in distinguishing between the classes. Their MCC scores are also the highest, which show that they are robust to classes imbalance. Their high AUC indicates high confidence in their predictions, and, as a result, better trade-off between sensitivity and specificity. Regarding the features sets, the Altman set produces the best overall results, as indicated by the highest best scores in all metrics. On the other hand, the ANOVA set produces the worst overall results. Besides the comparisons, it is noted that, overall, the models do not have adequate precision scores. This is caused by several factors. First, the lack of predictive ability may stem from the broad data. The data collected contains firms of varying industries. The industry-specific characteristics play a crucial role in determining what constitutes bad or good financial statistics. The lack of focus thus introduce noises to the training process. Regarding the box plots, overall, the ensemble methods are the most consistent in their performances. They also show higher median scores across metrics. The only model to have close performances is MLP, which is less consistent as indicated by its wide performances variation ranges. GP performs worse than MLP, with lower median scores and lower spanning ranges. LSEOFOA-KELM and GWO-KELM have very similar performances, despite the difference in their optimization algorithm. This may suggests that the optimization process does not have a crucial impact on the performances of the models. Their performances are also consistently poorer than the ensemble methods, with lower median and lower spanning ranges across categories. The worst performers are SVM and LR, with low median scores and often wide interquartile range. E-SMOTE-ADASVM-TW, although outperforms the ensemble methods in recall, performs poorly and lacks consistency in all other categories. In conclusion, the ensemble methods have consistently better performances while the other models often perform poorly and are less consistent.

Lastly, The analysis of the ROC curves has shown that the ensemble methods have the best performances among the reviewed methods. These models have higher ROC curves across all the features sets. Combined with the previous analyses of the average results tables and the box plots, it is clear that the ensemble models are the most optimal in predicting bankruptcy, providing the best and most consistent results.

6 Conclusions

In conclusion, the experiment has provided multiple valuable insights into the application of ML in bankruptcy prediction. First and most notable, the unbalanced nature of the data has a significant impact of the overall usability of the models. The ML models are unable to identify the positive class adequately. Although SMOTE have been applied, their precision are nevertheless low. This is speculated to be a results of the diverse dataset. Second, regarding the reviewed papers, most of the recently proposed methods, particularly E-SMOTE-ADASVM-TW, LSEOFOA-KELM, and GWO-KELM, do not have the reliability required to be put to practical use. The proposed models, although embedding varying methods and having increased complexity, can not be justified by their performance. This is evident from their poorer performance compared to previously proposed models such as RF and GDBT. Third, regarding the experiment results, it is clear that ensemble methods have the most reliable performances. These models are shown to have robustness against unbalanced dataset and overall consistent performance. Researchers and professionals can benefit from the employment of these ensemble methods in future research and practical usage. Furthermore, the Altman set also shows to be most suitable for the prediction task. The performances from the models when trained on this set is superior to those of the other generated sets. However, this may also be caused by the broad nature of the experiment data, where the simple Altman set can manage to capture the patterns without introducing noises.

The above discussion points out a possible future research direction. In particular, a more focused study, which address the problems with the broad datasets, is important in determining the applicability of the models. Besides, this thesis has not discussed the interpretability of the models, which is of great importance to financial analysts. Analysts and researchers alike often prefer models which can be interpreted easily. These models allow stakeholders to understand how decisions are made, securing trust and transparency in a high stake professional environment. It is also easier for professionals to communicate their findings to end-users and other professionals. Another possible future consideration is in the application of sentiment or other forms of non-financial analyses. Although financial analysis can provide an understanding of a firm underlying business, sentiment analysis can help stakeholders understand a firm's intangible values, such as brand recognition and customers reception.

In summary, while the current study emphasizes the challenges and limitations of applying machine learning to bankruptcy prediction, it also highlights promising areas for future research. By addressing these challenges and exploring new directions, researchers and professionals can make better choices in real world application of ML in finance.

References

[1] F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications*, vol. 83, p. 405 – 417, 2017. Cited by: 479.

[2] R. B. Carton and C. W. Hofer, *Measuring organizational performance: Metrics for entrepreneurship and strategic management research*. Edward Elgar Publishing, 2006.

[3] W. H. Beaver, "Financial ratios as predictors of failure," *Journal of Accounting Research*, vol. 4, pp. 71–111, 1966.

[4] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The journal of finance*, vol. 23, no. 4, pp. 589–609, 1968.

[5] J. A. Ohlson, "Financial ratios and the probabilistic prediction of bankruptcy," *Journal of accounting research*, pp. 109–131, 1980.

[6] M. D. Odom and R. Sharda, "A neural network model for bankruptcy prediction," in *1990 IJCNN International Joint Conference on neural networks*, pp. 163–168, IEEE, 1990.

[7] R. L. Wilson and R. Sharda, "Bankruptcy prediction using neural networks," *Decision Support Systems*, vol. 11, no. 5, pp. 545–557, 1994.

[8] G. Zhang, M. Y. Hu, B. Eddy Patuwo, and D. C. Indro, "Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis," *European Journal of Operational Research*, vol. 116, no. 1, pp. 16–32, 1999.

[9] Z. Yang, M. B. Platt, and H. D. Platt, "Probabilistic neural networks in bankruptcy prediction," *Journal of Business Research*, vol. 44, no. 2, pp. 67–74, 1999.

[10] K.-S. Shin, T. S. Lee, and H. jung Kim, "An application of support vector machines in bankruptcy prediction model," *Expert Systems with Applications*, vol. 28, no. 1, pp. 127–135, 2005.

[11] S.-H. Min, J. Lee, and I. Han, "Hybrid genetic algorithms and support vector machines for bankruptcy prediction," *Expert Systems with Applications*, vol. 31, no. 3, pp. 652–660, 2006.

[12] Y. Ding, X. Song, and Y. Zen, "Forecasting financial condition of chinese listed companies based on support vector machine," *Expert Systems with Applications*, vol. 34, no. 4, pp. 3081–3089, 2008.

[13] A. Chaudhuri and K. De, "Fuzzy support vector machine for bankruptcy prediction," *Applied Soft Computing*, vol. 11, no. 2, pp. 2472–2486, 2011. The Impact of Soft Computing for the Progress of Artificial Intelligence.

[14] S. Cho, H. Hong, and B.-C. Ha, "A hybrid approach based on the combination of variable selection using decision trees and case-based reasoning using the mahalanobis distance: For bankruptcy prediction," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3482–3488, 2010.

[15] S. Y. Kim and A. Upneja, "Predicting restaurant financial distress using decision tree and adaboosted decision tree models," *Economic Modelling*, vol. 36, pp. 354–362, 2014.

[16] G. Wang, J. Ma, and S. Yang, "An improved boosting based on feature selection for corporate bankruptcy prediction," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2353–2361, 2014.

[17] M.-J. Kim and D.-K. Kang, "Classifiers selection in ensembles using genetic algorithms for bankruptcy prediction," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9308–9314, 2012.

[18] C.-F. Tsai and J.-W. Wu, "Using neural network ensembles for bankruptcy prediction and credit scoring," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2639–2649, 2008.

[19] M.-J. Kim and D.-K. Kang, "Ensemble with neural networks for bankruptcy prediction," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3373–3379, 2010.

[20] S. Ben Jabeur, N. Stef, and P. Carmona, "Bankruptcy prediction using the xgboost algorithm and variable importance feature engineering," *Computational Economics*, vol. 61, no. 2, p. 715 – 741, 2023. Cited by: 31.

[21] E. Alfaro, N. García, M. Gámez, and D. Elizondo, "Bankruptcy forecasting: An empirical comparison of adaboost and neural networks," *Decision Support Systems*, vol. 45, no. 1, pp. 110–122, 2008. Data Warehousing and OLAP.

[22] J. Sun, H. Fujita, P. Chen, and H. Li, "Dynamic financial distress prediction with concept drift based on time weighting combined with adaboost support vector machine ensemble," *Knowledge-Based Systems*, vol. 120, p. 4 – 14, 2017. Cited by: 138.

[23] J. H. Min and C. Jeong, "A binary classification method for bankruptcy prediction," *Expert Systems with Applications*, vol. 36, no. 3, Part 1, pp. 5256–5263, 2009.

[24] H. Ahn and K. jae Kim, "Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach," *Applied Soft Computing*, vol. 9, no. 2, pp. 599–607, 2009.

[25] N. Gordini, "A genetic algorithm approach for smes bankruptcy prediction: Empirical evidence from italy," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6433–6445, 2014.

[26] C.-F. Tsai, "Combining cluster analysis with classifier ensembles to predict financial distress," *Information Fusion*, vol. 16, pp. 46–58, 2014. Special Issue on Information Fusion in Hybrid Intelligent Fusion Systems.

[27] P. Gandhi, T. Loughran, and B. McDonald, "Using annual report sentiment as a proxy for financial distress in u.s. banks," *Journal of Behavioral Finance*, vol. 20, no. 4, p. 424 – 436, 2019. Cited by: 53.

[28] N. Mselmi, A. Lahiani, and T. Hamza, "Financial distress prediction: The case of french small and medium-sized firms," *International Review of Financial Analysis*, vol. 50, p. 67 – 80, 2017. Cited by: 105.

[29] S. B. Jabeur and Y. Fahmi, "Forecasting financial distress for french firms: a comparative study," *Empirical Economics*, vol. 54, no. 3, p. 1173 – 1186, 2018. Cited by: 27.

[30] E. C. Charalambakis and I. Garrett, "On corporate financial distress prediction: What can we learn from private firms in a developing economy? evidence from greece," *Review of Quantitative Finance and Accounting*, vol. 52, no. 2, p. 467 – 491, 2019. Cited by: 36; All Open Access, Hybrid Gold Open Access.

[31] T. Kristóf and M. Virág, "A comprehensive review of corporate bankruptcy prediction in hungary," *Journal of Risk and Financial Management*, vol. 13, no. 2, 2020. Cited by: 19; All Open Access, Gold Open Access, Green Open Access.

[32] D. Yazdanfar and P. Öhman, "Financial distress determinants among smes: empirical evidence from sweden," *Journal of Economic Studies*, vol. 47, no. 3, p. 547 – 560, 2020. Cited by: 22.

[33] J. Kitowski, A. Kowal-Pawul, and W. Lichota, "Identifying symptoms of bankruptcy risk based on bankruptcy prediction models—a case study of poland," *Sustainability (Switzerland)*, vol. 14, no. 3, 2022. Cited by: 16; All Open Access, Gold Open Access.

[34] B. Pham Vo Ninh, T. Do Thanh, and D. Vo Hong, "Financial distress and bankruptcy prediction: An appropriate model for listed firms in vietnam," *Economic Systems*, vol. 42, no. 4, p. 616 – 624, 2018. Cited by: 38.

[35] S. Sehgal, R. K. Mishra, F. Deisting, and R. Vashisht, "On the determinants and prediction of corporate financial distress in india," *Managerial Finance*, vol. 47, no. 10, p. 1428 – 1447, 2021. Cited by: 17; All Open Access, Green Open Access.

[36] D. Veganzones and E. Séverin, "An investigation of bankruptcy prediction in imbalanced datasets," *Decision Support Systems*, vol. 112, p. 111 – 124, 2018. Cited by: 122.

[37] J. Sun, J. Lang, H. Fujita, and H. Li, "Imbalanced enterprise credit evaluation with dte-sbd: Decision tree ensemble based on smote and bagging with differentiated sampling rates," *Information Sciences*, vol. 425, p. 76 – 91, 2018. Cited by: 305.

[38] J. Sun, H. Li, H. Fujita, B. Fu, and W. Ai, "Class-imbalanced dynamic financial distress prediction based on adaboost-svm ensemble combined with smote and time weighting," *Information Fusion*, vol. 54, p. 128 – 144, 2020. Cited by: 221.

[39] T. Hosaka, "Bankruptcy prediction using imaged financial ratios and convolutional neural networks," *Expert Systems with Applications*, vol. 117, p. 287 – 299, 2019. Cited by: 160.

[40] D. Liang, C.-C. Lu, C.-F. Tsai, and G.-A. Shih, "Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study," *European Journal of Operational Research*, vol. 252, no. 2, pp. 561–572, 2016.

[41] D. Liang, C.-F. Tsai, H.-Y. R. Lu, and L.-S. Chang, "Combining corporate governance indicators with stacking ensembles for financial distress prediction," *Journal of Business Research*, vol. 120, p. 137 – 146, 2020. Cited by: 67.

[42] E. Tobback, T. Bellotti, J. Moeyersoms, M. Stankova, and D. Martens, "Bankruptcy prediction for smes using relational data," *Decision Support Systems*, vol. 102, p. 69 – 81, 2017. Cited by: 67; All Open Access, Green Open Access.

[43] G. Wang, G. Chen, and Y. Chu, "A new random subspace method incorporating sentiment and textual information for financial distress prediction," *Electronic Commerce Research and Applications*, vol. 29, p. 30 – 49, 2018. Cited by: 45.

[44] P. Gandhi, T. Loughran, and B. McDonald, "Using annual report sentiment as a proxy for financial distress in u.s. banks," *Journal of Behavioral Finance*, vol. 20, no. 4, p. 424 – 436, 2019. Cited by: 53.

[45] S. Li, W. Shi, J. Wang, and H. Zhou, "A deep learning-based approach to constructing a domain sentiment lexicon: a case study in financial distress prediction," *Information Processing and Management*, vol. 58, no. 5, 2021. Cited by: 53.

[46] F. Mai, S. Tian, C. Lee, and L. Ma, "Deep learning models for bankruptcy prediction using textual disclosures," *European Journal of Operational Research*, vol. 274, no. 2, p. 743 – 758, 2019. Cited by: 213.

[47] C.-F. Tsai, "Feature selection in bankruptcy prediction," *Knowledge-Based Systems*, vol. 22, no. 2, pp. 120–127, 2009.

[48] F. Lin, D. Liang, C.-C. Yeh, and J.-C. Huang, "Novel feature selection methods to financial distress prediction," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2472–2483, 2014.

[49] D. Liang, C.-F. Tsai, and H.-T. Wu, "The effect of feature selection on financial distress prediction," *Knowledge-Based Systems*, vol. 73, pp. 289–297, 2015.

[50] G. Kou, Y. Xu, Y. Peng, F. Shen, Y. Chen, K. Chang, and S. Kou, "Bankruptcy prediction for smes using transactional data and two-stage multiobjective feature selection," *Decision Support Systems*, vol. 140, p. 113429, 2021.

[51] S. B. Jabeur, C. Gharib, S. Mefteh-Wali, and W. B. Arfi, "Catboost model and artificial intelligence techniques for corporate failure prediction," *Technological Forecasting and Social Change*, vol. 166, 2021. Cited by: 137.

[52] S. Ben Jabeur, "Bankruptcy prediction using partial least squares logistic regression," *Journal of Retailing and Consumer Services*, vol. 36, p. 197 – 202, 2017. Cited by: 80.

[53] C.-H. Chou, S.-C. Hsieh, and C.-J. Qiu, "Hybrid genetic algorithm and fuzzy clustering for bankruptcy prediction," *Applied Soft Computing Journal*, vol. 56, p. 298 – 316, 2017. Cited by: 101.

[54] H. Son, C. Hyun, D. Phan, and H. Hwang, "Data analytic approach for bankruptcy prediction," *Expert Systems with Applications*, vol. 138, 2019. Cited by: 85.

[55] Y. Zhang, R. Liu, A. A. Heidari, X. Wang, Y. Chen, M. Wang, and H. Chen, "Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis," *Neurocomputing*, vol. 430, p. 185 – 212, 2021. Cited by: 222.

[56] M. Wang, H. Chen, H. Li, Z. Cai, X. Zhao, C. Tong, J. Li, and X. Xu, "Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction," *Engineering Applications of Artificial Intelligence*, vol. 63, p. 54 – 68, 2017. Cited by: 171.

[57] W. Pietruszkiewicz, "Dynamical systems and nonlinear kalman filtering applied in classification," in *2008 7th IEEE International Conference on Cybernetic Intelligent Systems*, pp. 1–6, IEEE, 2008.

[58] F. Antunes, B. Ribeiro, and F. Pereira, "Probabilistic modeling and visualization for bankruptcy prediction," *Applied Soft Computing Journal*, vol. 60, p. 831 – 843, 2017. Cited by: 63; All Open Access, Green Open Access.

[59] X. Du, W. Li, S. Ruan, and L. Li, "Cus-heterogeneous ensemble-based financial distress prediction for imbalanced dataset with ensemble feature selection," *Applied Soft Computing Journal*, vol. 97, 2020. Cited by: 51.

[60] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, p. 321 – 357, 2002. Cited by: 19748; All Open Access, Gold Open Access.

[61] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 904, p. 23 – 37, 1995. Cited by: 2724.

[62] V. N. Vapnik, "The support vector method," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1327, p. 264 – 271, 1997. Cited by: 42.

[63] J. C. Schlimmer and R. H. Granger, "Incremental learning from noisy data," *Machine Learning*, vol. 1, no. 3, p. 317 – 354, 1986. Cited by: 468; All Open Access, Bronze Open Access.

[64] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[65] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," *Advances in neural information processing systems*, vol. 31, 2018.

[66] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.

[67] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, p. 513 – 529, 2012. Cited by: 5179.

[68] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006. Neural Networks.

[69] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[70] W.-T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.

[71] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

[72] Y. Zhou, R. Wang, and Q. Luo, "Elite opposition-based flower pollination algorithm," *Neurocomputing*, vol. 188, pp. 294–310, 2016. Advanced Intelligent Computing Methodologies and Applications.

[73] L. M. Lopucki, "Ucla-lopucki bankruptcy research," 2022. Accessed: 2024-07-15.

[74] E. R. Girden, *ANOVA: Repeated measures*. No. 84, Sage, 1992.

[75] D. Freedman, R. Pisani, and R. Purves, "Statistics (international student edition)," *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

[76] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6, p. 066138, 2004.

[77] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, pp. 81–106, 1986.

[78] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

A Appendix

Table A1: Financial ratios formulas

Name	Formula
working capital to total assets (A)	working capital / total assets
retained earnings to assets (B)	retained earnings/ total assets
bep ratio (C)	EBIT / total assets
market capitalization to liabilities (D)	market capitalization / liabilities
revenue on assets (E)	revenue / total assets
z-score (Z)	Z = 1.2A + 1.4B + 3.3C + 0.6D + 1.0E
net profit margin (NPM)	net profit / revenue
return on assets (ROA)	net profit / total assets
return on equity (ROE)	net profit / equity
gross margin (GM)	gross profit / revenue
operating margin (OM)	EBIT / revenue
EBITDA margin (EBITDAM)	EBITDA / revenue
operating cashflow to assets (OCFA)	operating cashflow / total assets
working capital to sales (WCS)	(current assets - current liabilities) / revenue
return on capital employed (ROCE)	EBIT / (fixed assets + current assets - current liabilities)
net cash flow to revenue (CFR)	net cashflow / revenue
net op cash flow to revenue (OCFR)	operating cashflow / revenue
revenue to inventory (ROIN)	revenue / inventory
net profit to current assets ratio (NPCA)	net profit / current assets
EBITDA to total assets (EBITDAA)	EBITDA / total assets
net profit to fixed assets ratio (NPFA)	net profit / fixed assets
gross on equity (GOE)	revenue / equity
pretax income on equity (PIOE)	pretax income / equity
return on invested capital (ROIC)	(EBIT - tax) / invested capital
current assets turnover (CAT)	revenue / average current assets

Name	Formula
total assets turnover (TAT)	revenue / average total assets
capital intensity ratio (CIR)	total assets / revenue
inventory turnover (IT)	cost of goods sold / average inventory
research development efficiency (RDE)	revenue growth / research development expenses
fixed assets turnover (FAT)	revenue / average fixed assets
current ratio (CR)	current assets / current liabilities
quick ratio (QR)	(current assets - inventory) / current liabilities
solvency ratio (SR)	equity / assets
interest coverage (IC)	EBIT / interest expense
debt on assets (DOA)	debt / total assets
gearing (G) net profit on liabilities (NPOL)	debt / equity net profit / liabilities
EBITDA on liabilities (EBITDAOL)	EBITDA / liabilities
fixed assets on debt (FAOD)	fixed assets / debt
liabilites on fixed assets (LOFA)	liabilities / fixed assets
working capital on current assets (WCOCA)	working capital / current assets
current assets on assets (CAOA)	current assets / total assets
capital employed on fixed assets (CEOFA)	(fixed assets + current assets - current liabilities) / fixed assets
fixed assets on total assets (FAOA)	fixed assets / total assets
equity on fixed assets (EOFA)	equity / fixed assets
change on equity (C-ceq)	equity (ending) - equity (beginning)
current liabilities on assets (CLOA)	current liabilities / total assets
current assets minus inventory on assets (CAIA)	(current assets - inventory) / total assets
assets on liabilities (AOL)	total assets / liabilities
equity on current liabilities (EOCL)	equity / current liabilities
equity on non current liabilities (EONCL)	equity / (liabilities - current liabilities)

Name	Formula
equity on liabilities (EOL)	equity / liabilities
current liabilities on liabilities (CLOL)	current liabilities / liabilities
net cashflow on assets (CFOA)	net cashflow / total assets
net cashflow to current liabilities (CFOCL)	net cashflow / current liabilities
net cashflow on debt (CFOD)	net cashflow / debt
net profit growth (G-ni)	(net profit (ending) - net profit (beginning)) / net profit (beginning)
cash reinvestment ratio (CRR)	(change in fixed assets + change in working capital) / net cashflow
assets growth (G-at)	(assets (ending) - assets (beginning)) / assets (beginning)
revenue growth (G-revt)	(revenue (ending) - revenue (beginning)) / revenue (beginning)
employees number growth (G-emp)	(employees (ending) - employees (beginning)) / employees (beginning)
ROE change (C-ROE)	ROE (ending) - ROE (beginning)
PB change (C-PB)	PB (ending) - PB (beginning)
research development intensity (RDI)	research development expense / revenue
operating leverage coefficient (OLC)	EBIT growth / revenue growth
financial leverage coefficient (FLC)	EBIT / EBT (EBIT - I)
total leverage coefficient (LC)	FLC x OLC
earning per share (epspi)	net profit / number of shares outstanding
cash flow per share (CFPS)	net cashflow / number of shares outstanding
price to earning ratio (PE)	equity price / epspi
price to cash flow ratio (PCF)	equity price / CFPS
price to book (PB)	market capitalization / equity

Table A2: Financial ratios usage in the reviewed papers

Financial ratio names / Database, Dataset	Self [38]	Orbis [51]	Wieslaw [55, 56]	DIANE [58]	Self [59]	Total
working capital to total assets (A)	0	0	1	0	1	2
retained earnings to assets (B)	0	0	0	0	1	1
bep ratio (C)	1	0	1	0	0	2
market capitalization to liabilities (D)	1	0	0	0	0	1
revenue on assets (E)	0	0	1	0	0	1
z-score (Z)	0	0	0	0	0	0
net profit margin (NPM)	1	1	1	1	0	4
return on assets (ROA)	0	1	1	1	0	3
return on equity (ROE)	1	1	1	0	0	3
gross margin (GM)	0	1	1	1	0	3
operating margin (OM)	0	1	0	1	0	2
EBITDA margin (EBITDAM)	0	1	0	1	0	2
operating cashflow to assets (OCFA)	0	0	0	0	1	1
working capital to sales (WCS)	0	0	1	1	0	2
return on capital employed (ROCE)	0	1	1	1	0	3
net cash flow to revenue (CFR)	0	1	0	0	0	1
net op cash flow to revenue (OCFR)	0	0	0	1	0	1
revenue to inventory (ROIN)	0	0	1	0	0	1
net profit to current assets ratio (NPCA)	1	0	1	0	0	2

Financial ratio names / Database, Dataset	Self [38]	Orbis [51]	Wieslaw [55, 56]	DIANE [58]	Self [59]	Total
EBITDA to total assets (EBITDAA)	0	0	0	0	0	0
net profit to fixed assets ratio (NPFA)	1	0	0	0	0	1
gross on equity (GOE)	0	0	0	0	0	0
pretax income on equity (PIOE)	0	0	0	0	0	0
return on invested capital (ROIC)	0	0	0	0	0	0
current assets turnover (CAT)	0	0	1	1	0	2
total assets turnover (TAT)	0	1	0	1	0	2
capital intensity ratio (CIR)	1	0	0	0	0	1
inventory turnover (IT)	0	0	0	0	0	0
research development efficiency (RDE)	0	0	0	0	0	0
fixed assets turnover (FAT)	0	0	0	0	0	0
current ratio (CR)	0	1	1	1	1	4
quick ratio (QR)	0	1	0	1	0	2
solvency ratio (SR)	0	1	0	0	1	2
interest coverage (IC)	0	1	0	1	1	3
debt on assets (DOA)	0	0	0	0	0	0
gearing (G)	0	1	0	0	0	1
net profit on liabilities (NPOL)	0	0	1	0	0	1
EBITDA on liabilities (EBITDAOL)	0	0	0	0	1	1
fixed assets on debt (FAOD)	0	0	0	0	1	1
liabilites on fixed assets (LOFA)	1	0	0	0	0	1
working capital on current assets (WCOCA)	0	0	0	1	1	2

Financial ratio names / Database, Dataset	Self [38]	Orbis [51]	Wieslaw [55, 56]	DIANE [58]	Self [59]	Total
current assets on assets (CAOA)	0	0	1	0	0	1
capital employed on fixed assets (CEOFA)	0	0	0	1	0	1
fixed assets on total assets (FAOA)	1	0	0	0	0	1
equity on fixed assets (EOFA)	0	0	0	0	1	1
change on equity (C-ceq)	0	0	0	0	0	0
current liabilities on assets (CLOA)	0	0	0	0	0	0
current assets minus inventory on assets (CAIA)	0	0	0	0	0	0
assets on liabilities (AOL)	0	0	1	0	0	1
equity on current liabilities (EOCL)	0	0	1	0	0	1
equity on non current liabilities (EONCL)	0	0	1	0	0	1
equity on liabilities (EOL)	0	0	1	0	0	1
current liabilities on liabilities (CLOL)	0	0	0	0	0	0
net cashflow on assets (CFOA)	0	0	1	0	0	1
net cashflow to current liabilities (CFOCL)	0	0	1	0	0	1
net cashflow on debt (CFOD)	0	0	0	0	0	0
net profit growth (G-ni)	1	0	0	0	0	1
cash reinvestment ratio (CRR)	0	0	0	0	1	1
assets growth (G-at)	0	0	0	0	0	0
revenue growth (G-revt)	0	0	0	0	0	0

Financial ratio names / Database, Dataset	Self [38]	Orbis [51]	Wieslaw [55, 56]	DIANE [58]	Self [59]	Total
employees number growth (G-emp)	0	0	0	0	0	0
ROE change (C-ROE)	0	0	0	0	0	0
PB change (C-PB)	0	0	0	0	0	0
research development intensity (RDI)	0	0	0	0	0	0
operating leverage coefficient (OLC)	1	0	0	0	0	1
financial leverage coefficient (FLC)	0	0	0	0	0	0
total leverage coefficient (LC)	1	0	0	0	0	1
earning per share (epspi)	1	0	0	0	0	1
cash flow per share (CFPS)	0	0	0	0	0	0
price to earning ratio (PE)	1	0	0	0	0	1
price to cash flow ratio (PCF)	0	0	0	0	0	0
price to book (PB)	0	0	0	0	0	0

Table A3: Financial ratios types

Name	Type
working capital to total assets (A) retained earnings to assets (B) bep ratio (C) market capitalization to liabilities (D) revenue on assets (E) z-score (Z)	Altman's ratios [4]
net profit margin (NPM) return on assets (ROA) return on equity (ROE) gross margin (GM) operating margin (OM) EBITDA margin (EBITDAM)	Profitability

Name	Type
operating cashflow to assets (OCFA) working capital to sales (WCS) return on capital employed (ROCE) net cash flow to revenue (CFR) net op cash flow to revenue (OCFR) revenue to inventory (ROIN) net profit to current assets ratio (NPCA) EBITDA to total assets (EBITDAA) net profit to fixed assets ratio (NPFA) gross on equity (GOE) pretax income on equity (PIOE) return on invested capital (ROIC)	Profitability
current assets turnover (CAT) total assets turnover (TAT) capital intensity ratio (CIR) inventory turnover (IT) research development efficiency (RDE) fixed assets turnover (FAT)	Operating capacity and efficiency
current ratio (CR) quick ratio (QR) solvency ratio (SR) interest coverage (IC) debt on assets (DOA) gearing (G) net profit on liabilities (NPOL) EBITDA on liabilities (EBITDAOL) fixed assets on debt (FAOD) liabilites on fixed assets (LOFA)	Solvency and liquidity
working capital on current assets (WCOCA) current assets on assets (CAOA) capital employed on fixed assets (CEOFA) fixed assets on total assets (FAOA) equity on fixed assets (EOFA) change on equity (C-ceq) current liabilities on assets (CLOA) current assets minus inventory on assets (CAIA) assets on liabilities (AOL) equity on current liabilities (EOCL) equity on non current liabilities (EONCL) equity on liabilities (EOL) current liabilities on liabilities (CLOL)	Structure

Name	Type
net cashflow on assets (CFOA) net cashflow to current liabilities (CFOCL) net cashflow on debt (CFOD)	Cashflow
net profit growth (G-ni) cash reinvestment ratio (CRR) assets growth (G-at) revenue growth (G-revt) employees number growth (G-emp) ROE change (C-ROE) PB change (C-PB) research development intensity (RDI)	Growth
operating leverage coefficient (OLC) financial leverage coefficient (FLC) total leverage coefficient (LC)	Leverage
earning per share (epspi) cash flow per share (CFPS) price to earning ratio (PE) price to cash flow ratio (PCF) price to book (PB)	Equity

Table A4: Datasets used in the experiment

Ratios	Altman	Barboza	Popular	CC	MI	ANOVA	GDBT-FI
A	1	1	0	0	0	0	0
B	1	1	0	1	0	1	0
C	1	1	0	0	0	0	0
D	1	1	0	0	1	0	1
E	1	1	0	0	0	0	0
Z	1	0	0	1	0	1	0
NPM	0	0	1	1	0	1	0
ROA	0	0	1	1	0	1	0
ROE	0	0	1	0	0	0	0
GM	0	0	1	0	0	0	0
OM	0	1	0	0	0	0	0
OCFA	0	0	0	1	0	1	0

Ratios	Altman	Barboza	Popular	CC	MI	ANOVA	GDBT-FI
ROCE	0	0	1	0	0	0	0
OCFR	0	0	0	1	0	1	1
NPCA	0	0	0	1	0	1	0
NPFA	0	0	0	0	1	0	0
GOE	0	0	0	0	0	0	1
CAT	0	0	0	0	0	0	1
FAT	0	0	0	0	0	0	1
CR	0	0	1	0	0	0	0
SR	0	0	0	1	1	1	0
IC	0	0	0	0	1	0	0
DOA	0	0	0	0	1	0	1
G	0	0	0	0	1	0	1
NPOL	0	0	0	1	0	1	0
EOFA	0	0	0	0	0	0	1
AOL	0	0	0	0	1	0	0
EOL	0	0	0	0	1	0	0
G-at	0	1	0	0	0	0	0
G-revt	0	1	0	0	0	0	0
G-emp	0	1	0	0	0	0	0
C-ROE	0	1	0	0	0	0	0
C-PB	0	1	0	0	0	0	0
FLC	0	0	0	0	1	0	1
epspi	0	0	0	1	0	1	1
PE	0	0	0	0	1	0	0

Table A5: Descriptive statistics of the Altman (1968) set.

	A	B	C	D	E	Z
count	77842	77842	77842	77842	77842	77842
mean	0.392	0.703	0.592	0.149	0.304	0.262
std	0.251	0.206	0.203	0.219	0.243	0.208
min	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.197	0.652	0.511	0.027	0.123	0.134
50%	0.345	0.739	0.603	0.066	0.244	0.206
75%	0.548	0.824	0.703	0.159	0.419	0.313
max	1.000	1.000	1.000	1.000	1.000	1.000

Table A6: Descriptive statistics of the Barboza (2017) set.

	A	B	C	D	E	OM
count	73892	73892	73892	73892	73892	73892
mean	0.390	0.701	0.592	0.150	0.304	0.736
std	0.252	0.206	0.203	0.220	0.242	0.178
min	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.194	0.649	0.512	0.027	0.124	0.710
50%	0.343	0.737	0.603	0.067	0.245	0.756
75%	0.547	0.823	0.703	0.161	0.419	0.815
max	1.000	1.000	1.000	1.000	1.000	1.000

	G-at	G-revt	G-emp	C-roe	C-pb
count	73892	73892	73892	73892	73892
mean	0.395	0.392	0.353	0.502	0.477
std	0.172	0.173	0.164	0.150	0.161
min	0.000	0.000	0.000	0.000	0.000
25%	0.335	0.330	0.301	0.485	0.450
50%	0.374	0.374	0.330	0.502	0.477
75%	0.427	0.429	0.373	0.519	0.498
max	1.000	1.000	1.000	1.000	1.000

Table A7: Descriptive statistics of the popular set.

	CR	NPM	ROA	ROE	GM	ROCE	IC
count	84646	84646	84646	84646	84646	84646	84646
mean	0.259	0.742	0.660	0.550	0.431	0.443	0.181
std	0.224	0.181	0.198	0.166	0.257	0.180	0.179
min	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.108	0.735	0.611	0.518	0.237	0.368	0.119
50%	0.196	0.774	0.689	0.563	0.392	0.420	0.130
75%	0.336	0.817	0.765	0.602	0.599	0.502	0.159
max	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table A8: Descriptive statistics of the CC set.

	B	Z	NPM	ROA	OCFA	OCFR
count	76949	76949	76949	76949	76949	76949
mean	0.703	0.262	0.777	0.666	0.546	0.570
std	0.206	0.208	0.182	0.201	0.212	0.181
min	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.652	0.135	0.780	0.616	0.434	0.491
50%	0.739	0.206	0.814	0.698	0.545	0.555
75%	0.824	0.313	0.850	0.774	0.671	0.653
max	1.000	1.000	1.000	1.000	1.000	1.000

	NPCA	SR	NPOL	epspi
count	76949	76949	76949	76949
mean	0.598	0.566	0.545	0.488
std	0.188	0.238	0.179	0.192
min	0.000	0.000	0.000	0.000
25%	0.549	0.417	0.494	0.402
50%	0.615	0.567	0.544	0.479
75%	0.684	0.733	0.609	0.571
max	1.000	1.000	1.000	1.000

Table A9: Descriptive statistics of the MI set.

	D	NPFA	EOFA	EOL	AOL	DOA
count	66350	66350	66350	66350	66350	66350
mean	0.177	0.452	0.123	0.256	0.254	0.404
std	0.223	0.160	0.206	0.222	0.222	0.165
min	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.040	0.416	0.027	0.111	0.109	0.338
50%	0.098	0.433	0.045	0.193	0.191	0.370
75%	0.210	0.472	0.110	0.322	0.318	0.428
max	1.000	1.000	1.000	1.000	1.000	1.000

	SR	IC	FLC	PE
count	66350	66350	66350	66350
mean	0.563	0.248	0.485	0.500
std	0.237	0.179	0.163	0.172
min	0.000	0.000	0.000	0.000
25%	0.412	0.174	0.457	0.430
50%	0.576	0.196	0.475	0.504
75%	0.729	0.250	0.515	0.557
max	1.000	1.000	1.000	1.000

Table A10: Descriptive statistics of the GDBT-FI set.

	D	OCFR	GOE	FAT	CAT	EOFA
count	57710	57710	57710	57710	57710	57710
mean	0.182	0.514	0.426	0.165	0.321	0.173
std	0.223	0.189	0.172	0.219	0.230	0.213
min	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.045	0.418	0.350	0.029	0.161	0.056
50%	0.104	0.487	0.396	0.091	0.274	0.089
75%	0.218	0.597	0.470	0.200	0.420	0.191
max	1.000	1.000	1.000	1.000	1.000	1.000

	DOA	G	FLC	epspi
count	57710	57710	57710	57710
mean	0.436	0.436	0.497	0.481
std	0.162	0.162	0.163	0.193
min	0.000	0.000	0.000	0.000
25%	0.374	0.374	0.470	0.393
50%	0.407	0.407	0.488	0.468
75%	0.464	0.464	0.530	0.562
max	1.000	1.000	1.000	1.000

Table A11: Descriptive statistics of the ANOVA set.

	OCFR	Z	B	OCFA	NPOL	epspi
count	76949	76949	76949	76949	76949	76949
mean	0.570	0.262	0.703	0.546	0.545	0.488
std	0.181	0.208	0.206	0.212	0.179	0.192
min	0.000	0.000	0.000	0.000	0.000	0.000
25%	0.491	0.135	0.652	0.434	0.494	0.402
50%	0.555	0.206	0.739	0.545	0.544	0.479
75%	0.653	0.313	0.824	0.671	0.609	0.571
max	1.000	1.000	1.000	1.000	1.000	1.000

	NPM	ROA	SR	NPCA
count	76949	76949	76949	76949
mean	0.777	0.666	0.566	0.598
std	0.182	0.201	0.238	0.188
min	0.000	0.000	0.000	0.000
25%	0.780	0.616	0.417	0.549
50%	0.814	0.698	0.567	0.615
75%	0.850	0.774	0.733	0.684
max	1.000	1.000	1.000	1.000