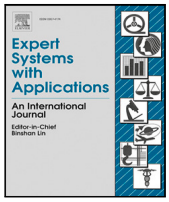
Expert Systems With Applications 258 (2024) 125133

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Enhancing corporate bankruptcy prediction via a hybrid genetic algorithm and domain adaptation learning architecture

T. Ansah-Narh [a,b,c,*], E.N.N. Nortey [b], E. Proven-Adzri [a], R. Opoku-Sarkodie [d]

[a] Ghana Space Science and Technology Institute, Ghana Atomic Energy Commission, P. O. Box LG 80, Legon-Accra, Ghana
[b] Department of Statistics and Actuarial Science, University of Ghana, P. O. Box LG 115, Legon-Accra, Ghana
[c] School of Technology, Ghana Institute of Management and Public Administration, P. O. Box AH 50, Achimota-Accra, Ghana
[d] Department of Information Technology and Mathematical Sciences, Methodist University Ghana, P. O. Box DC 940, Dansoman-Accra, Ghana

ARTICLE INFO

Dataset link: Taiwanese Bankruptcy Prediction dataset, Polish Companies Bankruptcy data

Keywords:
Bankruptcy prediction
Financial ratios
Genetic algorithm
Domain adaptation learning
Data distribution shifts
Bayesian optimisation

ABSTRACT

In the contemporary business landscape, accurately evaluating a company's financial health is essential for stakeholders to mitigate risks and avert bankruptcy. This study presents an innovative approach to improving business bankruptcy prediction through the hybrid integration of Domain Adaptation Learning (DAL) and Genetic Algorithm (GA) techniques. The hybrid model harnesses DAL to address distributional changes in real-world scenarios and utilises GA's proficiency in feature selection. Six machine learning models are rigorously evaluated against the proposed hybrid model: Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), Gradient Boosting (GB), *k*-Nearest Neighbours (*k*-NN), and Stacking Ensemble (SE). Our hybrid model performs well on imbalanced target datasets using the Area Under the Precision–Recall Curve metric: 0.93 (RF), 0.93 (SVM), 0.89 (LR), 0.91 (GB), 0.88 (*k*-NN), and 0.92 (SE). These findings highlight the model's ability to overcome the limitations of traditional approaches, offering a more reliable predictive framework for stakeholders to make informed decisions and proactively manage financial stability. Future research directions may explore the applicability of this hybrid model across different industries and the integration of additional techniques to further enhance its performance.

1. Introduction

Examining a company's financial performance is an important task, as it plays a pivotal role in determining its strengths and weaknesses. A company's daily transaction records serve as a valuable source of information for decision-making, especially when focusing on scenarios that lead to bankruptcy. When a company experiences financial distress, it undergoes a gradual evolution, initially with limited liquidity and eventually leading to bankruptcy (Fahlevi & Marlinah, 2018). In today's business environment, there has been a marked increase in the number of companies facing financial failure and subsequent liquidation. A relevant example is the financial sector reforms that began in Ghana in 2017, which resulted in the central bank revoking the licenses of 23 universal banks and 388 microfinance and microcredit companies.[1] Also, because privately held enterprises frequently lack the trustworthy and open financial statements of publicly audited organisations, recent research by da Silva Mattos and Shasha (2024) has shown how difficult it is to predict insolvency for these types of businesses. Managing these less trustworthy reports presents special difficulties for stakeholders as a result. These case studies emphasise how vital it is to carry out in-depth research in order to offer information that will help pertinent stakeholders prevent business defaults. In fact, the Ghana case serves as a compelling illustration of the challenges faced by companies in dynamic economic environments, highlighting the critical need for predictive models that can adapt to evolving industry landscapes. While the data used in this study originates from the Taiwan bankruptcy prediction dataset, it also incorporates the Polish Companies Bankruptcy data and considers the Ghanaian context, highlighting the global issue of corporate bankruptcy and the need for adaptable predictive models across diverse financial environments. To address this urgent research need, it is important to advance our understanding of the complex dynamics that lead to corporate bankruptcy. This study aims to focus on the following primary research question: How can the accuracy and adaptability of bankruptcy prediction models be enhanced to effectively handle distributional changes in real-world scenarios? The recent financial crisis highlights the importance of continually re-evaluating and refining methods to improve predictive power. Given the complexity

* Corresponding author at: Ghana Space Science and Technology Institute, Ghana Atomic Energy Commission, P. O. Box LG 80, Legon-Accra, Ghana.
E-mail addresses: theophilus.ansah-narh@gaec.gov.gh (T. Ansah-Narh), ennnortey@ug.edu.gh (E.N.N. Nortey), emmanuel.proven-adzri@gaec.gov.gh (E. Proven-Adzri), rsarkodie@mucg.edu.gh (R. Opoku-Sarkodie).

[1] https://www.bog.gov.gh/wp-content/uploads/2019/08/Revocation-of-Licenses-of-SDIs-16.8.19.pdf.

https://doi.org/10.1016/j.eswa.2024.125133
Received 6 March 2024; Received in revised form 31 July 2024; Accepted 15 August 2024
Available online 20 August 2024

of the global business environment, a comprehensive investigation of the factors that influence a company's financial position is necessary. Additionally, the landscape of financial analysis is constantly changing due to technological advances, so it is essential to utilise innovative methodologies.

Previous research has mostly concentrated on using structural and statistical approaches to predict insolvency. The latter, which is the subject of this study, uses traditional machine learning models such as *k*-nearest neighbours (Chen et al., 2011; Li & Wang, 2017), discriminant analysis (Altman, 1968; Kliestik, Vrbka, & Rowland, 2018), logit models (Chi & Tang, 2006; Li, Lee, Zhou, & Sun, 2011), artificial neural networks (ANN) (Odom & Sharda, 1990; Zhang, Hu, Patuwo, & Indro, 1999), and decision trees (Olson, Delen, & Meng, 2012; Syed Nor, Ismail, & Yap, 2019). For instance, the study by Min, Lee, and Han (2006) discusses how bankruptcy prediction affects bank lending decisions and profitability. In contrast to logistic regression (LR) and neural networks, it emphasises the recent application of support vector machines (SVM) in this field and shows its promising outcomes. The article highlights the growing application of genetic algorithms (GA) in conjunction with other AI methods such as neural networks and case-based reasoning. It does, however, highlight the paucity of research on the combination of GA and SVM, in spite of its promise for useful applications. In order to improve bankruptcy prediction, the study uses GA to simultaneously optimise two factors–feature subset selection and parameter optimisation, in order to increase SVM performance. These models seek to pinpoint the relevant financial factors that directly impact bankruptcy prediction. The former strategy, on the other hand, entails complex accounting ratio forecasts and a thorough comprehension of the economic subtleties of the organisation being studied.

While the concept of ANN dates back approximately eight decades to McCulloch and Pitts (1943)'s threshold logic-based model designed to emulate the human brain, the modern environment is characterised by the incorporation of high-performance computing systems that are driving AI into the mainstream. The computational design of ANN is based on interconnected neurons, where each connection facilitates the transmission of signals from one neuron to another. The receiving neuron processes the signal and subsequently transmits the processed information to other interconnected neurons. The organisation of these neurons typically involves layers, with the first layer serving as the input and the last layer as the output. Sandwiched between these are hidden layers, which can be shallow or deep in design. The versatility of ANN architecture allows it to handle data ranging from single to multiple dimensions, making it applicable across a broad spectrum of cases. The efficiency of AI applications, particularly in bankruptcy prediction, has substantially increased due to the growth of large and diverse datasets, both organised and unstructured. Investigations by Awoyemi, Adetunmbi, and Oluwadare (2017), Kristóf and Virág (2020), Sharma, Banerjee, Tiwari, and Patni (2021), Tripathi, Edla, Cheruku, and Kuppili (2019) into bankruptcy prediction demonstrate this efficiency. In this domain, the term "prediction" is often used interchangeably with "classification" because the ultimate goal is to determine whether a company will likely face financial distress or bankruptcy. Table 1 provides a comparative analysis of various studies in bankruptcy prediction, highlighting the diversity in methods, datasets, and results.

However, the AI approach to bankruptcy prediction described above relies on assumptions inherent in classical machine learning. One key assumption is that the training and test sets come from the same distribution, meaning a model trained on labelled data is expected to perform effectively on test data. This assumption may not always hold in real-world applications where training and test data can come from different distributions. Discrepancies can arise from various factors, such as differences in the origins of the training and test sets or an outdated training set due to changes in data patterns over time. In instances where there is a disparity across domain distributions, blindly applying the trained model to a new dataset can lead to a decline in performance. Addressing this challenge falls within the realm of domain adaptation, a subfield within machine learning (Farahani, Voghoei, Rasheed, & Arabnia, 2021; Guan & Liu, 2021; Jiang & Zhai, 2007). The primary objective of domain adaptation is to mitigate issues arising from differing distributions by aligning them, thus enabling the trained model to generalise effectively within the domain of interest. This alignment process is crucial for ensuring the robustness and applicability of the predictive model in real-world scenarios.

In light of the complex issues mentioned, this study attempts to investigate a novel approach to reduce the effects of distributional changes. The principal aim is to facilitate the creation of bankruptcy prediction models with increased flexibility and robustness under dynamic and changing conditions. As a result, this work presents a hybrid model that combines the strengths of Genetic Algorithm (GA) and Domain Adaptation Learning (DAL). This strategy aims to create a more reliable and adaptable predictive model for bankruptcy analysis by combining the best features of both approaches. The GA employed utilises a heuristic search-based scheme to extract relevant financial features from the original dataset and feed them into the proposed DAL pipeline for bankruptcy prediction. Keep in mind that the computational method used by the optimisation solver in GA is based on biological evolution and follows the guidelines of the *natural evolution process* (Ghanea-Hercock, 2003). This methodology finds applications across diverse fields, showcasing its versatility. Notably, it has been extensively used in pattern recognition (Alsultanny & Aqel, 2003; Kim, Park, Yang, & Sim, 2006; Maulik & Bandyopadhyay, 2000; Pal & Wang, 1996), route optimisation (Inagaki, Haseyama, & Kitajima, 1999), network intrusion detection (Li, 2004), and image processing (Bhanu, Lee, & Ming, 1995; Saitoh, 1999). On the other hand, DAL is a machine learning paradigm that aims to address the challenges that arise when a model trained in one domain (source domain[2]) is deployed to another related domain.(target domain[3]) The core principle of domain adaptation pipelines is to leverage knowledge gained from the source domain to improve the generalisation and performance of the model in the target domain. This becomes very important when there are distributional changes between training data and deployment data. Therefore, for the purpose of this work, we seek to perform the following tasks:

i. Create a spatial distribution model to visually represent correlations between financial variables and gain a comprehensive understanding of hidden patterns within the original dataset.
ii. Mitigate biases in the source domain dataset through the use of simulation techniques, ensuring the robustness and reliability of the analysis.
iii. Systematically identify and select financial features considered crucial for predicting corporate bankruptcy within the source domain, thereby improving the precision of the subsequent modelling process.
iv. Utilise the financial features selected in Task (iii) within the source domain to apply domain adaptation techniques, ensuring the model's robustness in the face of variations in data distributions. This crucial step enhances the model's applicability to real-world scenarios beyond the training data by addressing potential disparities in data distribution.

We put together the following sections of the paper: Section 2 discusses the data acquisition process, emphasising the dataset's relevance, coverage of financial information, and the rigorous data-gathering procedures followed to ensure accuracy and applicability. Section 3 focuses on the classification measures used in the study to evaluate the performance of the proposed hybrid model for bankruptcy prediction.

[2] The source domain refers to the domain from which the training data is obtained to build a machine learning model.

[3] The target domain is the domain where the model's performance is evaluated. The target data is mostly imbalanced.

Table 1
Summary of various bankruptcy prediction models and their respective performance metrics.

Author	Dataset	Method[a]	ACC (%)	Obs. period	Attributes	Pred. type	Limitation
Almaskati, Bird, Yeung, and Lu (2021)	S & P firms	• ALT[b] • OHL[c] • ZMJ[d] • SHW[e] • BSH[f] • CHS[g] • PRM[h]	• 0.82 • 0.73 • 0.82 • 0.79 • 0.73 • 0.80 • 0.81	2005–2015	19	Bankruptcy	• Impact of specific governance variables • Comparison of different non-parametric methods • Temporal changes in governance impact
Liang, Tsai, Dai, and Eberle (2018)	• Taiwanese • Chinese • Australian • German	• SVM[i] • KNN[j] • MLP[k] • CART[l] • Bayes[m]	• 77.60–91.27 • 69.40–90.39 • 71.40–89.38 • 73.20–93.00 • 70.70–88.82	2005–2015	• 95 • 45 • 14 • 24	• Bankruptcy • Bankruptcy • Credit • Credit	• Exploration of new classifier ensembles • Type I error reduction • Dataset Diversity
Barboza, Kimura, and Altman (2017)	North American firms	• MDA[n], LR[o], ANN[p] • Bagging[q], Boosting[r], RF[s], SVM[i]	• 52–77 • 71–87	1985–2013	11	Bankruptcy	• Limited analysis of new financial indicators • Longitudinal changes in model performance • Study focused on North American firms
Heo and Yang (2014)	Korean construction companies	AdaBoost[t]	78.5	2008–2012	12	Bankruptcy	• Limited analysis of new financial indicators • Exploration of other algorithms • Study focused on Korean construction firms

[a] Method: The approach or algorithm used for bankruptcy prediction.
[b] ALT: Altman Z-score Model.
[c] OHL: Ohlson O-score Model.
[d] ZMJ: Zmijewski Model.
[e] SHW: Shumway Model.
[f] BSH: Bharath–Shumway Model.
[g] CHS: Campbell, Hilscher, and Szilagyi Model.
[h] PRM: Premachandra Model.
[i] SVM: Support Vector Machines.
[j] KNN: K-Nearest Neighbors.
[k] MLP: Multi-Layer Perceptron.
[l] CART: Classification and Regression Trees.
[m] Bayes: Bayesian Classifiers.
[n] MDA: Multiple Discriminant Analysis.
[o] LR: Logistic Regression.
[p] ANN: Artificial Neural Networks.
[q] Bagging: Bootstrap Aggregating.
[r] Boosting: Ensemble Technique for Improving Weak Models.
[s] RF: Random Forest.
[t] AdaBoost: Adaptive Boosting.

It outlines the metrics and evaluation criteria employed to assess the model's predictive accuracy, adaptability, and generalisation capabilities. The section provides insights into how the model's performance is measured and analysed in the context of imbalanced target datasets. The Section 4 delves into the implications of the results, providing a detailed analysis of how the hybrid model addresses the challenges of traditional approaches in bankruptcy prediction. The final section (in Section 5) summarises the key findings and contributions of the study, emphasising the significance of the hybrid model in enhancing corporate bankruptcy prediction. It discusses the implications of the research for financial risk management and decision-making, highlighting the potential impact of the hybrid model on stakeholders in the business landscape. The conclusion also outlines future research directions and areas for further exploration to enhance the performance and usefulness of the hybrid model in different industries and scenarios.

2. Data and methods

2.1. Data acquisition

In the present study, we utilised the Taiwan bankruptcy prediction dataset from the University of California, Irvine machine learning repository, originally compiled from the Taiwan Economic Journal and covering the period from 1999–2009. Additionally, the study incorporated the Polish Companies Bankruptcy data to evaluate the effectiveness of the proposed hybrid model. The selection of these datasets was based on their comprehensive coverage of financial information and provided a solid foundation for training and evaluation of the hybrid model. For instance, in the case of the Taiwan data, a rigorous procedure was followed during the data-gathering phase to guarantee the accuracy and applicability of the dataset (Liang, Lu, Tsai, & Shih, 2016). Two fundamental standards were utilised in the process of collecting the data:

i. The selected companies were required to disclose their financial information for at least three years before the start of the financial crisis. This criterion ensured that the dataset contained sufficient temporal context so that the model could capture pre-crisis trends and patterns.
ii. Another important criterion was to consider similar companies within the same industry. This step was important for a nuanced analysis of the financial picture, allowing the model to identify industry-specific dynamics. The goal was to improve the model's ability to generalise insights across companies with comparable economic conditions.

Table 2
Descriptive statistics of some financial ratios in the Taiwan bankruptcy data.

Variables	Count	Mean	Std	Min	25%	50%	75%	Max
Roa(C) Before Interest And Depreciation Before ...	6819.00	0.51	0.06	0.00	0.48	0.50	0.54	1.00
Roa(A) Before Interest And % After Tax	6819.00	0.56	0.07	0.00	0.54	0.56	0.59	1.00
Roa(B) Before Interest And Depreciation After Tax	6819.00	0.55	0.06	0.00	0.53	0.55	0.58	1.00
Operating Gross Margin	6819.00	0.61	0.02	0.00	0.60	0.61	0.61	1.00
Realised Sales Gross Margin	6819.00	0.61	0.02	0.00	0.60	0.61	0.61	1.00
Operating Profit Rate	6819.00	1.00	0.01	0.00	1.00	1.00	1.00	1.00
Pre-Tax Net Interest Rate	6819.00	0.80	0.01	0.00	0.80	0.80	0.80	1.00
After-Tax Net Interest Rate	6819.00	0.81	0.01	0.00	0.81	0.81	0.81	1.00
Non-Industry Income And Expenditure/Revenue	6819.00	0.30	0.01	0.00	0.30	0.30	0.30	1.00
Continuous Interest Rate (After Tax)	6819.00	0.78	0.01	0.00	0.78	0.78	0.78	1.00
Operating Expense Rate	6819.00	1,995,347,312.80	3,237,683,890.52	0.00	0.00	0.00	4,145,000,000.00	9,990,000,000.00
Research And Development Expense Rate	6819.00	1,950,427,306.06	2,598,291,554.00	0.00	0.00	509,000,000.00	3,450,000,000.00	9,980,000,000.00
Cash Flow Rate	6819.00	0.47	0.02	0.00	0.46	0.47	0.47	1.00
Interest-Bearing Debt Interest Rate	6819.00	16,448,012.91	108,275,033.53	0.00	0.00	0.00	0.00	990,000,000.00
Tax Rate (A)	6819.00	0.12	0.14	0.00	0.00	0.07	0.21	1.00
Net Value Per Share (B)	6819.00	0.19	0.03	0.00	0.17	0.18	0.20	1.00
Net Value Per Share (A)	6819.00	0.19	0.03	0.00	0.17	0.18	0.20	1.00
Net Value Per Share (C)	6819.00	0.19	0.03	0.00	0.17	0.18	0.20	1.00
Persistent Eps In The Last Four Seasons	6819.00	0.23	0.03	0.00	0.21	0.22	0.24	1.00
Cash Flow Per Share	6819.00	0.32	0.02	0.00	0.32	0.32	0.33	1.00
Revenue Per Share (Yuan ¥)	6819.00	1,328,640.60	51,707,089.77	0.00	0.02	0.03	0.05	3,020,000,000.00
Operating Profit Per Share (Yuan ¥)	6819.00	0.11	0.03	0.00	0.10	0.10	0.12	1.00
Per Share Net Profit Before Tax (Yuan ¥)	6819.00	0.18	0.03	0.00	0.17	0.18	0.19	1.00
Realised Sales Gross Profit Growth Rate	6819.00	0.02	0.01	0.00	0.02	0.02	0.02	1.00
Operating Profit Growth Rate	6819.00	0.85	0.01	0.00	0.85	0.85	0.85	1.00
After-Tax Net Profit Growth Rate	6819.00	0.69	0.01	0.00	0.69	0.69	0.69	1.00
Regular Net Profit Growth Rate	6819.00	0.69	0.01	0.00	0.69	0.69	0.69	1.00
Continuous Net Profit Growth Rate	6819.00	0.22	0.01	0.00	0.22	0.22	0.22	1.00
Total Asset Growth Rate	6819.00	5,508,096,595.25	2,897,717,771.17	0.00	4,860,000,000.00	6,400,000,000.00	7,390,000,000.00	9,990,000,000.00
Net Value Growth Rate	6819.00	1,566,212.06	114,159,389.52	0.00	0.00	0.00	0.00	9,330,000,000.00
Total Asset Return Growth Rate Ratio	6819.00	0.26	0.01	0.00	0.26	0.26	0.26	1.00
Cash Reinvestment %	6819.00	0.38	0.02	0.00	0.37	0.38	0.39	1.00
Current Ratio	6819.00	403,284.95	33,302,155.83	0.00	0.01	0.01	0.02	2,750,000,000.00
Quick Ratio	6819.00	8376594.82	244,684,748.45	0.00	0.00	0.01	0.01	9,230,000,000.00
Interest Expense Ratio	6819.00	0.63	0.01	0.00	0.63	0.63	0.63	1.00
Total Debt/Total Net Worth	6819.00	4,416,336.71	168,406,905.28	0.00	0.00	0.01	0.01	9,940,000,000.00
Debt Ratio %	6819.00	0.11	0.05	0.00	0.07	0.11	0.15	1.00
Net Worth/Assets	6819.00	0.89	0.05	0.00	0.85	0.89	0.93	1.00
Long-Term Fund Suitability Ratio (A)	6819.00	0.01	0.03	0.00	0.01	0.01	0.01	1.00
Borrowing Dependency	6819.00	0.37	0.02	0.00	0.37	0.37	0.38	1.00
Contingent Liabilities/Net Worth	6819.00	0.01	0.01	0.00	0.01	0.01	0.01	1.00
Operating Profit/Paid-In Capital	6819.00	0.11	0.03	0.00	0.10	0.10	0.12	1.00
Net Profit Before Tax/Paid-In Capital	6819.00	0.18	0.03	0.00	0.17	0.18	0.19	1.00
Inventory And Accounts Receivable/Net Value	6819.00	0.40	0.01	0.00	0.40	0.40	0.40	1.00
Total Asset Turnover	6819.00	0.14	0.10	0.00	0.08	0.12	0.18	1.00
Accounts Receivable Turnover	6819.00	12,789,705.24	278,259,836.98	0.00	0.00	0.00	0.00	9,740,000,000.00
Average Collection Days	6819.00	9,826,220.86	256,358,895.71	0.00	0.00	0.01	0.01	9,730,000,000.00
Inventory Turnover Rate (Times)	6819.00	2,149,106,056.61	3,247,967,014.05	0.00	0.00	0.00	4,620,000,000.00	9,990,000,000.00
Fixed Assets Turnover Frequency	6819.00	1,008,595,981.82	2,477,557,316.92	0.00	0.00	0.00	0.00	9,990,000,000.00
Net Worth Turnover Rate (Times)	6819.00	0.04	0.04	0.00	0.02	0.03	0.04	1.00
Revenue Per Person	6819.00	2,325,854.27	136,632,654.39	0.00	0.01	0.02	0.04	8,810,000,000.00
Operating Profit Per Person	6819.00	0.40	0.03	0.00	0.39	0.40	0.40	1.00
Allocation Rate Per Person	6819.00	11,255,785.32	294,506,294.12	0.00	0.00	0.01	0.02	9,570,000,000.00
Working Capital To Total Assets	6819.00	0.81	0.06	0.00	0.77	0.81	0.85	1.00
Quick Assets/Total Assets	6819.00	0.40	0.20	0.00	0.24	0.39	0.54	1.00
Current Assets/Total Assets	6819.00	0.52	0.22	0.00	0.35	0.51	0.69	1.00
Cash/Total Assets	6819.00	0.12	0.14	0.00	0.03	0.07	0.16	1.00
Quick Assets/Current Liability	6819.00	3,592,902.20	171,620,908.61	0.00	0.01	0.01	0.01	8,820,000,000.00
Cash/Current Liability	6819.00	37,159,994.15	510,350,903.16	0.00	0.00	0.00	0.01	9,650,000,000.00
Current Liability To Assets	6819.00	0.09	0.05	0.00	0.05	0.08	0.12	1.00
Operating Funds To Liability	6819.00	0.35	0.04	0.00	0.34	0.35	0.36	1.00
Inventory/Working Capital	6819.00	0.28	0.01	0.00	0.28	0.28	0.28	1.00
Inventory/Current Liability	6819.00	55,806,804.53	582,051,554.62	0.00	0.00	0.01	0.01	9,910,000,000.00
Current Liabilities/Liability	6819.00	0.76	0.21	0.00	0.63	0.81	0.94	1.00
Working Capital/Equity	6819.00	0.74	0.01	0.00	0.73	0.74	0.74	1.00
Current Liabilities/Equity	6819.00	0.33	0.01	0.00	0.33	0.33	0.33	1.00
Long-Term Liability To Current Assets	6819.00	54,160,038.14	570,270,621.96	0.00	0.00	0.00	0.01	9,540,000,000.00
Retained Earnings To Total Assets	6819.00	0.93	0.03	0.00	0.93	0.94	0.94	1.00
Total Income/Total Expense	6819.00	0.00	0.01	0.00	0.00	0.00	0.00	1.00
Total Expense/Assets	6819.00	0.03	0.03	0.00	0.01	0.02	0.04	1.00
Current Asset Turnover Rate	6819.00	1,195,855,763.31	2,821,161,238.26	0.00	0.00	0.00	0.00	10,000,000,000.00
Quick Asset Turnover Rate	6819.00	2,163,735,272.03	3,374,944,402.17	0.00	0.00	0.00	4,900,000,000.00	10,000,000,000.00
Working Capital Turnover Rate	6819.00	0.59	0.01	0.00	0.59	0.59	0.59	1.00
Cash Turnover Rate	6819.00	2471976967.44	2,938,623,226.68	0.00	0.00	1,080,000,000.00	4,510,000,000.00	10,000,000,000.00
Cash Flow To Sales	6819.00	0.67	0.01	0.00	0.67	0.67	0.67	1.00
Fixed Assets To Assets	6819.00	1,220,120.50	100,754,158.71	0.00	0.09	0.20	0.37	8,320,000,000.00
Current Liability To Liability	6819.00	0.76	0.21	0.00	0.63	0.81	0.94	1.00
Current Liability To Equity	6819.00	0.33	0.01	0.00	0.33	0.33	0.33	1.00
Equity To Long-Term Liability	6819.00	0.12	0.02	0.00	0.11	0.11	0.12	1.00

The dataset covered a wide range of industries, such as the manufacturing sector (which includes industrial and electronics enterprises), the service sector (which includes shipping, tourist, and retail companies), and other non-financial industry entities. The dataset comprises a substantial sample of 6819 observations, each characterised by 96 attributes. Within this dataset, 220 observations have been identified as instances of bankruptcy. In Table 2, we present comprehensive descriptive statistics for the chosen financial ratios in the dataset. The statistical metrics employed encompass fundamental measures such as mean, standard deviation, minimum, maximum, and percentiles at 25%, 50%, and 75%. These metrics offer a general overview of the distribution and central tendencies of the selected financial ratios, providing valuable insights into their variability and the overall profile of the Taiwan bankruptcy data under examination. Three important factors in the current investigation necessitate domain adaptation.

First off, the dataset spans a sizable amount of time, from 1999 to 2009, and may include notable changes in industry dynamics, financial reporting standards, or economic situations. The training data (before the financial crisis) and possible test data (perhaps obtained after the financial crisis or in another economic environment) can become disconnected as a result of these temporal shifts, causing the model to perform poorly when applied to new data. In fact, the challenge of dataset shifts affecting the performance of supervised learning predictors has necessitated the development of a framework like `DetectShift1` to quantify and address these shifts (Maia Polo, Izbicki, Lacerda, Ibieta-Jimenez, & Vicente, 2023). There are three main types of data shifts that can affect model performance: Covariate Shift, where the input features' distribution changes between the training and testing datasets while the output variable's distribution remains the same, potentially leading to biased predictions; Concept Shift,

where the relationship between input features and the output variable changes due to factors like economic conditions or industry trends; and Prior Probability Shift, where the distribution of the target variable changes, affecting the model's predictive accuracy, particularly in cases of imbalanced data (Quiñonero-Candela, Sugiyama, Schwaighofer, & Lawrence, 2022).

Second, the diversity of the dataset across different industries highlights the necessity of understanding industry-specific dynamics. Each industry can experience unique shifts due to various factors such as technological advancements, regulatory changes, and evolving market conditions. These shifts might not be captured in the training set, leading to a model that performs well on historical data but poorly on new data reflecting current industry conditions. For example, a model trained on pre-2008 financial data might not account for post-crisis regulatory changes that significantly impact financial reporting and risk assessment. Similarly, a model trained on manufacturing data from an era of manual processes may struggle to predict outcomes in a modern, highly automated industry. The possibility of unanticipated shifts in industry conditions or trends underscores the importance of domain adaptability. Domain adaptation techniques allow models to adjust to new industry environments by learning from both historical and current data. This involves identifying industry-specific features that remain relevant over time, weighting instances to prioritise more recent and relevant data, and developing representations that are robust to changes in industry dynamics. By incorporating these techniques, models can generalise well across a range of industry situations, maintaining their accuracy and reliability even as industry conditions evolve.

Finally, a typical criticism of biased training data in the setting of machine learning has been identified, pointing out the possible influence of bias on the performance of the model, especially with respect to minority target labels.

The DAL approach proves invaluable in tackling data temporal shifts by strategically aligning features, weighting instances, and crafting domain-invariant representations. These techniques ensure that machine learning models adapt effectively to changes in the temporal distribution of data. By selecting and transforming features that remain stable across different time periods, assigning higher weights to instances that reflect the target domain's temporal characteristics, and learning representations insensitive to temporal variations, models become more resilient to temporal shifts. Additionally, harnessing transfer learning strategies such as pre-training on diverse temporal data and fine-tuning using domain adaptation methods enhances the model's ability to generalise and perform well across varying temporal contexts.

2.2. Handling outliers

Removing outliers from financial ratios before bankruptcy detection is essential to ensure the accuracy and reliability of the analysis. Outliers can skew statistical measures, distort trends, and mislead the interpretation of financial data, which can have significant implications, especially in critical decisions like bankruptcy prediction. A recent study conducted by Nyitrai and Virág (2019) highlighted the necessity of financial indicators in predicting bankruptcy and the challenges posed by outliers in these indicators. The authors explored different approaches to handling outliers, specifically focusing on winsorisation and the use of CHAID[4]-based categorisation of financial ratios.

In this work, we adopted a hybrid Bayesian change point and Hampel identifier (BCP-HI) method (Pehlivan, 2024). This amalgamation scheme can potentially identify outliers more precisely than winsorisation. Winsorisation replaces extreme values with values from the tails of the distribution, which can mask subtle outliers, especially when dealing with multiple change points. The first component, BCP analysis, helps pinpoint these potential change points, allowing the HI portion to better target outliers within those segments. Additionally, the hybrid method utilises the data itself to identify outliers. By modelling the data with a normal distribution before and after potential change points, it can compute unique probabilities for each data point, leading to a more data-driven approach to outlier detection compared to winsorisation's fixed threshold approach.

2.2.1. The integration of BCP-HI method

The BCP-HI outlier detection depicted in Algorithm 1 begins by preprocessing the financial data and determining initial parameters such as the window size (w) and the number of change points (cp). The notation w represents the size of the window used for outlier detection, while the number of cp indicates the expected number of shifts in the data distribution. These parameters are crucial for the effectiveness of the algorithm in identifying outliers accurately.

Algorithm 1 BCP-HI Outlier Detection Algorithm

```
Input: Financial ratio data
Output: Refined financial ratio with outliers corrected
procedure BCP-HI(data)
    Preprocess the input data
    Determine initial parameters: window size w, number of change points cp
    Perform Bayesian Change Point (BCP) analysis to identify change points
    Divide data into subsegments based on change points
    for each subsegment do
        Calculate median and MAD for the subsegment
        Apply Hampel Identifier (HI) for outlier detection
        Replace identified outliers with NaN values
        Apply median filter of size w to correct outliers
    end for
    return Refined time series data with outliers corrected
end procedure
```

Let denote the financial metric dataset over a given period as $\boldsymbol{D} = \{x_1, x_2, \ldots, x_n\}$ with n data points. The hybrid method first employs BCP analysis to detect significant shifts (cps) in the financial metric data. These change points represent instances where there is a notable change in the underlying distribution or behaviour of the financial metric. Consider the set of change points as $cps = \{cp_1, cp_2, \ldots, cp_k\}$, where k signifies the number of detected change points. The BCP analysis segments the financial data into subsegments with distinct distribution properties. Suppose S_i represent the ith subsegment, where $i = 1, 2, \ldots, k+1$. Each subsegment S_i is delineated by two adjacent change points cp_i and cp_{i+1}, such that $S_i = \{x_{cp_i}, x_{cp_i+1}, \ldots, x_{cp_{i+1}-1}\}$. This first approach incorporates Bayesian probabilistic modelling to estimate the likelihood of each data point belonging to a specific subsegment given the observed financial metric data as defined in Eq. (1);

$$P(S_i|x_j) = \frac{P(x_j|S_i) \cdot P(S_i)}{\sum_{m=1}^{k+1} P(x_j|S_m) \cdot P(S_m)}, \tag{1}$$

where $P(S_i|x_j)$ represents the posterior probability of data point x_j belonging to subsegment S_i, $P(x_j|S_i)$ signifies the likelihood of observing data point x_j under the distribution parameters of subsegment S_i, $P(S_i)$ denotes the prior probability of subsegment S_i, and $\sum_{m=1}^{k+1} P(x_j|S_m) \cdot P(S_m)$ calculates the weighted sum of likelihoods across all subsegments, ensuring that probabilities sum to 1. This Bayesian methodology facilitates the probabilistic detection of significant changes (change points) in financial metric data and enables the segmentation of the data into subsegments with distinct distributional characteristics, thereby enhancing the analysis and processing of each subsegment independently.

[4] Chi-squared Automatic Interaction Detector.

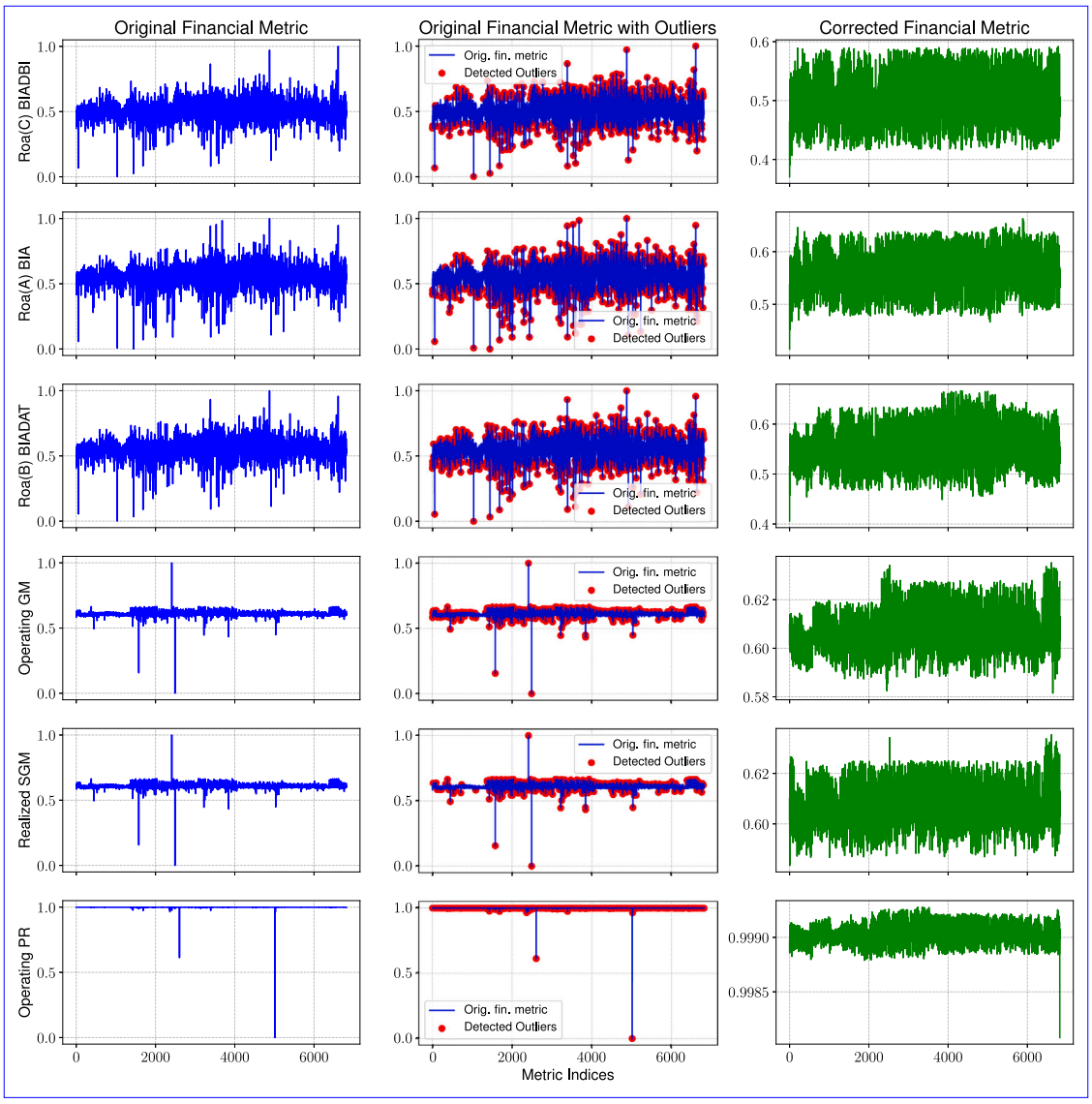


Fig. 1. Outlier detection and correction in financial metrics using a hybrid BCP-HI scheme. The first column shows the original time series data for each financial metric before outlier detection. The second Column highlights the outliers (in red circles) detected in the original data using the HI method. Column three shows the time series data after outliers have been removed.

Within each subsegment S_i, the algorithm calculates the median, denoted as median(S_i). The Median Absolute Deviation (MAD) is then computed, which measures the dispersion of the data points within the subsegment. The MAD is defined as the median of the absolute deviations from the median of the subsegment, mathematically expressed as $\text{MAD}(S_i) = \text{median}(|x_j - \text{median}(S_i)|)$ for all x_j in S_i. After computing the median and MAD for each subsegment, the algorithm applies the HI for outlier detection. The HI flags a data point x_j in subsegment S_i as an outlier if its absolute deviation from the median exceeds a specified threshold. This threshold is typically set as a multiple of the MAD. Specifically, a data point x_j is considered an outlier if $|x_j - \text{median}(S_i)| > k \times \text{MAD}(S_i)$, where k is a constant multiplier that determines the sensitivity of the outlier detection. By applying this criterion, the algorithm effectively identifies outliers within each subsegment based on the robust statistical properties of the median and MAD.

Identified outliers in the financial ratios are replaced with NaN values. Let $\{o_1, o_2, \ldots, o_m\}$ denote the indices of the identified outliers in the data D. For each outlier x_{o_i} in the dataset, we set $x_{o_i} = \text{NaN}$. To correct for these outliers, a median filter of size w is applied to the time series. The median filter processes the time series by sliding a window of size w across the data points. For each window position, the median value of the data points within the window is computed. If the window is centred at index j, the window includes data points $\{x_{j-\lfloor w/2 \rfloor}, x_{j-\lfloor w/2 \rfloor+1}, \ldots, x_{j+\lfloor w/2 \rfloor}\}$. The median of these data points, excluding NaN values, is used to replace the NaN value at index j. Mathematically, for each outlier index o_i, the corrected value is given by:

$$x_{o_i} = \text{median}(\{x_{o_i-\lfloor w/2 \rfloor}, x_{o_i-\lfloor w/2 \rfloor+1}, \ldots, x_{o_i+\lfloor w/2 \rfloor}\} \setminus \{\text{NaN}\}) \quad (2)$$

This process ensures that outliers are replaced with more representative values based on the local neighbourhood of data points, effectively smoothing the financial ratios while preserving important trends and patterns. The plots in Fig. 1 show the results of the hybrid outlier detection method applied to several financial metrics over the observed period of time. The y-axis represents the values of the financial metric, and the x-axis represents the metric indices. Take note of how severe outliers displayed in the figure are identified and calibrated leveraging on the BCP-HI statistical technique.

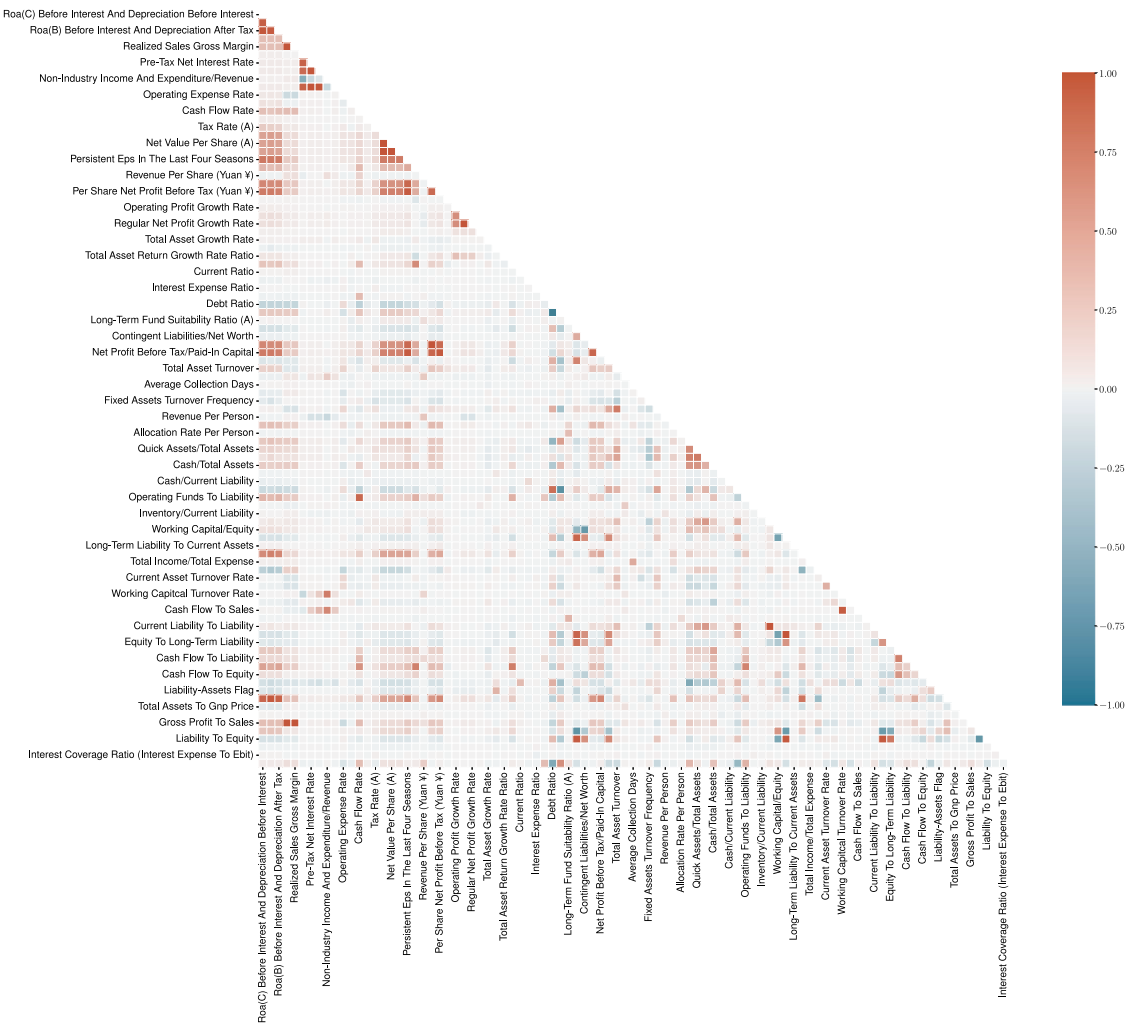


Fig. 2. A graph of correlation matrix describing the linear association between financial variables. Each element in the triangular matrix shows the correlation coefficient between two variables.

2.3. Bivariate analysis

We propose a correlation matrix (in Fig. 2) to thoroughly examine the links between financial ratios. A quantitative indicator of the relationships between these financial ratios is the Pearson correlation coefficient (r), which can be calculated using the following formula in Eq. (3) (Cohen et al., 2009):

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \cdot \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}. \tag{3}$$

The individual data points of the two financial ratios are represented by X_i and Y_i, their respective means are shown by $\bar{X}$ and $\bar{Y}$, and the total number of data points is indicated by n. We observe from the matrix plot that most of the financial variables have values that are approaching 0. This closeness suggests that there is little correlation between any two of the chosen variables, confirming that there is no discernible multicollinearity between the independent financial ratios. These observed characteristics underscore the necessity of utilising all financial ratios to ascertain their collective importance in determining relevant features.

2.4. Genetic algorithm for feature selection

Genetic algorithms, or GAs, have proven time and time again to be remarkably efficient at resolving a wide range of optimisation issues. (Nolfi, Floreano, Miglino, Mondada, et al., 1994) documented their achievements in a variety of applications, including sophisticated robot motion optimisation, control system parameter fine-tuning, and robotic system path planning. In addition to their conventional uses, GAs are flexible in machine learning, especially when it comes to feature selection. This versatility makes GA a valuable tool for systematic navigation in situations with complex combinations of features. The heuristic search algorithm uses principles inspired by natural selection and evolution to iteratively refine a subset of traits and gradually converge to an optimal set. Adopting this approach effectively balances model complexity and prediction metrics such as F1-score, precision, recall, and AUC-ROC, while also improving classifier performance. In the present work, we investigate the GA algorithm used to determine the ideal subset of features on the financial dataset that maximises the model classifier's accuracy.

Let $\mathbf{X}$ denote the feature matrix of a dataset with N instances and L features, and $\mathbf{y}$ represent the corresponding target label vector. The following characteristics describe the GA:

(i) Population size (N): Representing the number of individuals in each generation of the GA. The notation $N \in \mathbb{Z}^+$ is a key factor in determining population diversity and the trade-off between exploration and exploitation.
(ii) Offspring production (λ): The number of offspring produced in each generation, λ determines the rate at which new genetic

material is introduced into the population. Like N, λ is also a positive integer: $\lambda \in \mathbb{Z}^+$.

(iii) Crossover probability (P_c): Reflecting the likelihood of mating occurring between two individuals. The symbol P_c influences the exploration–exploitation balance by controlling the exchange of genetic material between parents. Mathematically, P_c is a probability value between 0 and 1: $P_c \in [0, 1]$.

(iv) Mutation probability (P_m): This parameter represents the likelihood that a bit in an individual's binary string will be flipped. Mutation introduces genetic diversity and prevents premature convergence. Mathematically, P_m is a probability value between 0 and 1: $P_m \in [0, 1]$.

(v) Number of generations (G): G signifies the total iterations or epochs for which the genetic algorithm will run, determining how many times the evolutionary process (selection, crossover, and mutation) is applied. Mathematically, G is a positive integer: $G \in \mathbb{Z}^+$.

Each individual in the population is represented by a binary string of length L. The binary string encodes the presence (1) or absence (0) of each feature in the subset. Mathematically, an individual I is represented as

$$I = [g_1, g_2, \ldots, g_L]. \tag{4}$$

Here, g_i is the ith gene in the binary string, indicating the presence or absence of the ith feature. The binary encoding provides a concise and flexible representation of feature subsets. Each gene g_i is a binary variable defined as $g_i \in \{0, 1\}$. The initial population is then formed by randomly generating binary strings of length L, ensuring diversity in the initial set of individuals. The initialisation of an individual I can be expressed as

$$I_i \sim \text{Bernoulli}(0.5) \quad \text{for } i = 1, 2, \ldots, L. \tag{5}$$

Algorithm 2 Genetic Algorithm for Feature Selection

Input: Dataset features **X**, target labels **y**, Population size N, Lambda λ, Crossover probability P_c, Mutation probability P_m, Number of generations G
Output: Selected features selected_features_ga
Initialise Population:
    Each individual I is represented as a binary string of length L, where L is the number of features.
Initialise Genetic Algorithm Set:
    Define fitness function fitness_function, crossover function, mutation function, and selection function
Initialise Population:
    Create a population of N individuals
Evaluate Initial Population:
    Evaluate the fitness of each individual using the fitness function
for $t = 1$ to G **do**
    Apply Crossover and Mutation:
        Generate offspring using crossover and mutation operations with probabilities P_c and P_m
    Evaluate Offspring:
        Evaluate the fitness of each offspring using the fitness function
    Select Individuals for the Next Generation:
        Use the selection function to choose individuals for the next generation based on their fitness
end for
Select Best Individual:
    Choose the individual with the highest fitness as the best individual
Extract Selected Features:
    Extract indices of selected features from the best individual: selected_features_ga

The expression Bernoulli(p) represents a Bernoulli distribution with probability p, and I_i is the ith gene in the binary string. Keeping in mind that genetic operations (namely, crossover and mutation) are fundamental processes in genetic algorithms that shape the evolution of the population over generations. In the context of feature selection, these operations manipulate the binary string representations of individuals to explore new solutions. Crossover involves combining genetic material from two parent individuals to create offspring. The crossover point is randomly selected along the binary string, and genetic material beyond that point is swapped between parents. For instance, let I_1 and I_2 be two parent individuals with binary string representations:

$$\begin{aligned} I_1 &= [g_1^1, g_2^1, \ldots, g_i^1, \ldots, g_L^1], \\ I_2 &= [g_1^2, g_2^2, \ldots, g_i^2, \ldots, g_L^2]. \end{aligned} \tag{6}$$

The crossover point C is randomly selected, and offspring O_1 and O_2 are created:

$$\begin{aligned} O_1 &= [g_1^1, g_2^1, \ldots, g_C^1, g_{C+1}^2, g_{C+2}^2, \ldots, g_L^2], \\ O_2 &= [g_1^2, g_2^2, \ldots, g_C^2, g_{C+1}^1, g_{C+2}^1, \ldots, g_L^1]. \end{aligned} \tag{7}$$

The offspring replaces the genetic material beyond the crossover point with the parents. After the crossover operation, the next step is the application of mutation. Mutation involves randomly changing the value of one or more genes in an individual. This introduces genetic diversity into the population and prevents premature convergence to suboptimal solutions. In this work, the mutation is applied to each gene independently with a mutation probability P_m:

$$\text{MutatedGene}_i = \begin{cases} 1 - g_i, & \text{with probability } P_m \\ g_i, & \text{with probability } 1 - P_m \end{cases}$$

This mutated gene replaces the original gene in the binary string. If, for example, a mutation with P_m occurs at position $j = 5$ in O_1, the binary string might be updated as follows:

$$\begin{aligned} &\text{Original} \quad O_1 = [g_1^1, g_2^1, g_3^1, g_4^2, g_5^2, \ldots, g_L^2], \\ &\text{Mutated} \quad O_1 = [g_1^1, g_2^1, g_3^1, g_4^2, \mathbf{1}, \ldots, g_L^2] \end{aligned} \tag{8}$$

Note that after the crossover and mutation operations, the population is updated with the newly created individuals. These individuals have genetic material inherited from their parents, with potential variations introduced through mutation. Now, to provide a quantitative measure of the individual's performance, we introduce a fitness function. The fitness function evaluates how well an individual performs the task at hand. In the context of feature selection, the fitness function measures the effectiveness of a subset of features. The GA aims to find the subset of features S that maximises the accuracy of a machine learning model. The fitness function is typically defined as the accuracy achieved by the model trained on the dataset with the selected features:

$$\begin{aligned} \text{Fitness}(I) = \text{Accuracy}(&\text{Model}(X_{\text{train},S}, y_{\text{train}}), \\ &X_{\text{test},S}, y_{\text{test}}) \end{aligned} \tag{9}$$

Here, X_{train} and y_{train} are the training features and labels, and X_{test} and y_{test} are the test features and labels. Model is the machine learning model used in this case the Extra Trees classifier, and S represents the selected features according to the binary string I. The ultimate goal is to maximise the fitness function, thereby identifying the subset of features that results in the highest accuracy. Practical development and execution of the GA are depicted in Algorithm 2. The visual representation in Fig. 3 provides a comprehensive view of the key metrics crucial for assessing financial stability and forecasting potential financial distress. Additionally, the lollipop chart not only showcases the significance of each ratio but also emphasises their hierarchical importance in the context of bankruptcy prediction. Relative feature importance for each feature is determined by averaging its importance across multiple GA runs and normalising these averages.

Next, we discuss the various machine learning models used in the study.

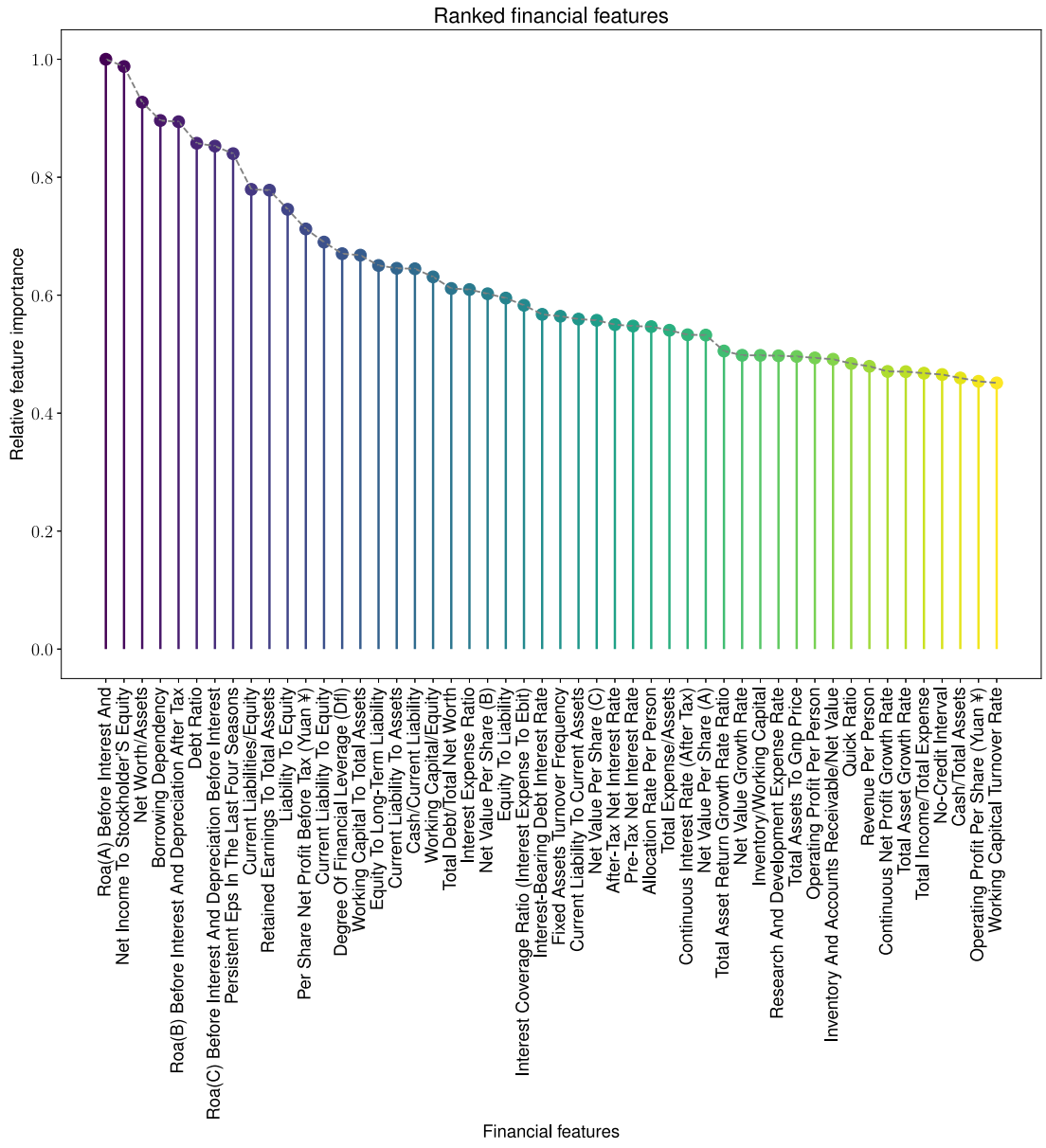


Fig. 3. A lollipop chart illustrating the distribution of the top 50 financial ratios, highlighting their relative importance in predicting the likelihood of bankruptcy.

2.5. Machine learning models

2.5.1. Random Forest (RF)

The RF algorithm is an effective ensemble learning method that avoids over-fitting by combining random feature selection with bagging to manipulate complex data patterns. The resulting mathematical framework illustrates the essential ideas that govern how successful random forests are as machine learning techniques.

Let $\mathcal{D}$ denote the original dataset housing $\mathcal{N}$ samples. For every decision tree within the ensemble (comprising a total of $\mathcal{T}$ trees), a bootstrap sample $\mathcal{D}_b$ of size $\mathcal{N}$ is created by iteratively selecting samples with replacement from $\mathcal{D}$ such that $\mathcal{D}_b = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$. Each decision tree undergoes training on its respective bootstrap sample, yielding $\mathcal{T}$ independently trained trees. This algorithm introduces an additional level of randomness by including only a subset of features at each node when building the decision tree. This deliberate selection of random features aims to decorrelate the trees, prevent excessive similarity, and allow us to capture different aspects of the data. Mathematically, at each node j of a decision tree, a random subset of features m is chosen from the complete feature set M such that $m \leq M$. The purpose of this stochastic selection is to increase the tree's diversity, fortify the algorithm's resilience, and enhance its capacity for generalisation. Taking into account the ensemble of decision trees $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_T(\mathbf{x})$, the final prediction $F(\mathbf{x})$ for a new input $\mathbf{x}$ is determined through a combination mechanism, typically involving voting for classification as given in Eqs. (10).

$$F(\mathbf{x}) = \text{mode}(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_T(\mathbf{x})) \quad \text{(for classification)} \tag{10}$$

2.5.2. Support Vector Machine (SVM)

The basic idea behind SVM is finding the best hyperplane in the feature space to separate different classes in the data effectively. This becomes especially important when dealing with binary classification issues, where the classes are usually designated as 0 and 1. Since its introduction by Vapnik (1982), SVM has been a major force in machine learning systems, outperforming many of its competitors in a short amount of time. Its ascendancy is attributed to the dual factors of simplicity and superior performance, as evidenced by studies such

as Peng and Xu (2013). The widespread adoption of SVM is underscored by its successful application across diverse research domains. Noteworthy fields where SVM has demonstrated its efficacy include finance, as exemplified by Luo, Yan, and Tian (2020), Tay and Cao (2001); chemistry, as explored by Li, Liang, and Xu (2009); renewable energy prediction, with contributions from Zendehboudi, Baseer, and Saidur (2018); medicine, as demonstrated by Wang, Zheng, Yoon, and Ko (2018); text classification, a domain addressed by Tong and Koller (2001); and face recognition, with seminal work by Osuna, Freund, and Girosit (1997).

Given a set of training data points $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, where $\mathbf{x}_i$ is the feature vector for the ith data point, and y_i is the corresponding class label such that $y_i \in \{0, 1\}$. The decision function of an SVM is given by Eq. (11):

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \tag{11}$$

Here, $\mathbf{w}$ is the weight vector, $\mathbf{x}$ is the input feature vector, and b is the bias term. The goal of SVM is to find the optimal hyperplane that maximises the margin between the two classes. The margin is defined as the distance between the hyperplane and the nearest data point from each class. Mathematically, the margin (M) is given by the formula in Eq. (12):

$$M = \frac{2}{\|\mathbf{w}\|}, \tag{12}$$

where $\|\mathbf{w}\|$ is the Euclidean norm of the weight vector. To ensure that the SVM correctly classifies the training data and maximises the margin, it must satisfy the following constraints:

i. For each positive training example ($\mathbf{x}_i$ with $y_i = 1$):

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1$$

ii. For each negative training example ($\mathbf{x}_i$ with $y_i = 0$):

$$\mathbf{w} \cdot \mathbf{x}_i + b < 0$$

The above constraints can be combined into a single expression to get Eq. (13):

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \tag{13}$$

This is the standard formulation of the linear SVM optimisation problem. Nonetheless, when confronted with intricate systems, such as the one under consideration, we enhance the foundational principles established in linear scenarios by incorporating a kernel function.

Employing SVMs with kernel methods involves working in high-dimensional feature spaces, allowing for the construction of non-linear decision boundaries. The decision function for SVM with a kernel K can be expressed as Eq. (14):

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{14}$$

Here, α_i are the Lagrange multipliers obtained during the optimisation process, and $K(\mathbf{x}_i, \mathbf{x})$ is the kernel function. The optimisation problem for SVM with kernel methods is given by:

$$\text{Minimise} \quad \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{n} \alpha_i$$

subject to the constraints:

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C \quad \text{for } i = 1, 2, \ldots, n$$

where C is the regularisation parameter that controls the trade-off between achieving a low training error and a large margin. The decision boundary is determined by the support vectors, which are the data points $\mathbf{x}_i$ corresponding to non-zero Lagrange multipliers α_i. The kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ implicitly computes the dot product of the data points in a higher-dimensional space, allowing SVMs to capture complex, non-linear relationships in the data. Commonly used kernel functions include:

i. Linear Kernel ($K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$): Corresponds to the standard linear SVM described in Eq. (11).
ii. Polynomial Kernel ($K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + c)^d$): Introduces non-linearity through polynomial terms.
iii. Radial Basis Function or Gaussian Kernel $\left(K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)\right)$ (Ding, Liu, Yang, & Cao, 2021): Provides a smooth, non-linear decision boundary.
iv. Sigmoid Kernel ($K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i \cdot \mathbf{x}_j + \theta)$): Represents a hyperbolic tangent function, introducing non-linearities.

These expressions capture the essence of SVM with kernel methods, which leverage the mathematical concept of the kernel to implicitly operate in a high-dimensional space, enabling the modelling of non-linear relationships in the data.

2.5.3. k-nearest neighbours (k-NN)

The basic idea underlying the nearest neighbour algorithm is rather simple: instances are grouped based on the class of their nearest neighbours. It is frequently advantageous to take into account not just one but several neighbours in order to increase the robustness of this method. Therefore, the commonly known approach is the k-Nearest Neighbour (k-NN) algorithm, where the class is determined based on the consensus of k nearest neighbours. The algorithm requires the availability of training examples during runtime, meaning they must be stored in memory at the time of execution. Consequently, it can also be referred to as a memory-based algorithm (Cunningham & Delany, 2020). As the real learning or model construction is deferred until runtime when predictions are needed, this technique is considered a form of lazy learning, making it flexible and adaptive to varying data patterns encountered during runtime.

Analytically, we can express the k-NN algorithm by considering a given dataset D with n data points in a feature space, where each data point i is represented by a feature vector $\mathbf{x}_i$ and is associated with a class label y_i (for classification). For a new data point $\mathbf{x}$ that we want to classify or predict, the k-NN algorithm operates as follows:

i. Distance Metric:- Let $d(\mathbf{x}_i, \mathbf{x})$ be a distance metric that measures the distance between data point $\mathbf{x}_i$ and the query point $\mathbf{x}$. Common distance metrics include Euclidean distance, Manhattan distance, or other suitable measures based on the problem at hand.
ii. Nearest Neighbours:- Identify the k nearest neighbours of the query point $\mathbf{x}$ from the dataset D based on the chosen distance metric. Let $N(\mathbf{x})$ represent the set of indices of these k nearest neighbours.
iii. For k-NN classification:- For classification tasks, assign the class label y to the query point based on majority voting among the class labels of its k nearest neighbours as given in Eq. (15):

$$y = \arg\max_{c} \sum_{i \in N(\mathbf{x})} \delta(y_i, c) \tag{15}$$

where $\delta(y_i, c)$ is the Kronecker delta function that equals 1 if $y_i = c$ and 0 otherwise.

The k-NN algorithm essentially relies on the assumption that points nearby in the feature space are likely to have similar labels or target values. The choice of the distance metric and the value of k are critical parameters that influence the algorithm's performance.

2.5.4. Logistic Regression (LR)

The LR is a widely used statistical method in machine learning for binary classification tasks. This model predicts the probability that an instance belongs to a particular class, and it is particularly useful in scenarios where the dependent variable is categorical and binary, such as predicting whether an event will occur or not. Statistically, the LR model is expressed in Eq. (16):

$$P(Y=1)=\frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n)}} \tag{16}$$

Here, $P(Y=1)$ represents the probability of the positive class, β_0 is the intercept, $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients, and $X_1, X_2, \ldots, X_n$ are the feature values. Keep in mind that each coefficient represents the change in the log-odds of the dependent variable for a one-unit change in the corresponding independent variable, holding other variables constant. Also, the intercept represents the log-odds of the event when all independent variables are zero.

This method is advantageous in machine learning for several reasons. Firstly, LR inherently produces probabilities, making the model interpretable and allowing decision-makers to understand the confidence level of predictions. Additionally, the LR method handles linear and non-linear relationships between features and the log-odds of the outcome, providing flexibility in capturing complex patterns in the data. Altman's Z-score, a popular bankruptcy prediction model, incorporates LR to assess the financial health of companies based on multiple financial ratios (Altman, 1968).

2.5.5. Gradient Boosting (GB)

The initial GB approach, often referred to as the GB Machine, was introduced by Friedman (1999, 2002). This method acts as the cornerstone algorithm that lays the groundwork for subsequent advancements in boosting techniques such as XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova, Gusev, Vorobev, Dorogush, & Gulin, 2018). GB is a powerful machine learning ensemble technique that combines the predictions of multiple weak learners, usually decision trees, to create a robust and accurate model.

Given a training dataset (x_i, y_i) where x_i represents the input features and y_i the corresponding target labels, the objective is to construct an ensemble of weak learners $h(x)$. The final prediction is obtained by combining these weak learners in an additive manner as presented in Eq. (17):

$$F(x)=\sum_{m=1}^{M}\beta_m h_m(x) \tag{17}$$

where M denotes the number of weak learners, β_m represents the weight assigned to each learner, and $h_m(x)$ is an individual weak learner. The fundamental idea behind GB is to iteratively fit new models to the errors of the existing ensemble, thereby reducing the residual errors in predictions. In each iteration m, the model $F(x)$ is updated to get Eq. (18):

$$F_m(x)=F_{m-1}(x)+\lambda_m\cdot h_m(x) \tag{18}$$

where $F_m(x)$ is the composite model at iteration m, $F_{m-1}(x)$ is the model from the previous iteration, and λ_m is the learning rate that controls the contribution of each weak learner. At each iteration, the negative gradient of the loss function with respect to the current model $F_{m-1}(x)$ is calculated, denoted as $-\frac{\partial L(F_{m-1}(x))}{\partial F_{m-1}(x)}$. The weak learner $h_m(x)$ is then trained to fit the negative gradient, minimising the local approximation of the loss. This iterative process continues until a predefined number of weak learners are incorporated into the ensemble.

The effectiveness of GB in bankruptcy prediction has been demonstrated in various studies. For instance, in a comprehensive analysis by Carmona, Climent, and Momparler (2019), GB algorithms were employed to develop predictive models for bankruptcy, showcasing superior performance compared to other machine learning techniques. The study emphasised the importance of ensemble methods, such as GB, in achieving high predictive accuracy and robustness in the context of bankruptcy prediction.

2.6. Bayesian hyperparameter tuning

The basis of Bayesian hyperparameter tuning is Bayesian optimisation–a stochastic model-based scheme for maximising costly opaque functions. The goal of this work is to determine which combination of hyperparameters maximises an objective function (in this case, the performance metric for the machine learning models discussed in Section 2.5.

The first step of this method is to define the objective function $f(x)$ with hyperparameters denoted by x. For instance, in machine learning classification, the model's hyperparameters are represented by x, whilst the accuracy, F1 score, or any other assessment metric could be represented by $f(x)$. We adopted the famous Gaussian process ($\mathcal{GP}$) to model the objective function. This is because the function creates a probabilistic estimate with associated uncertainty by finding the distribution of likely values of the objective function at a particular location. It consists of a covariance function $\kappa(x, x')$ and a mean function $m(x)$ as depicted in Eq. (19).

$$f(x)\sim\mathcal{GP}(m(x),\kappa(x,x')) \tag{19}$$

The notation $\kappa(x, x')$ models the correlation between various points in the input space, whereas the $m(x)$ records the objective function's average behaviour.

The next step is to iteratively select the next point for evaluation in the parameter space based on an acquisition function. This strategy aims to maintain a balance between exploration, which involves sampling in areas of high uncertainty, and exploitation, which involves sampling in areas where optimal solutions are expected to exist. The goal is to systematically navigate the search space through an intelligent selection of evaluation points and effectively combine the search for new information with the use of existing knowledge to control the optimisation process. This careful balance allows the algorithm to efficiently explore unknown regions of the parameter space whilst exploiting regions where the objective function is likely to reach its optimal value.

After specifying the new hyperparameter configuration and evaluating the objective function, the $\mathcal{GP}$ model is updated using Bayesian inference. The updated $\mathcal{GP}$ posterior distribution is given by Eq. (20):

$$P(f(x)|\boldsymbol{D})=\frac{P(\boldsymbol{D}|f(x))\cdot P(f(x))}{P(\boldsymbol{D})} \tag{20}$$

Here, $\boldsymbol{D}$ represents the data collected so far, $P(\boldsymbol{D}|f(x))$ is the probability of the data given in the $\mathcal{GP}$ model, $P(f(x))$ is the prior distribution of $\mathcal{GP}$ and $P(\boldsymbol{D})$ is the marginal probability.

Table 3 provides a comprehensive summary of the optimal hyperparameters for the selected machine learning algorithms. For each model, details are provided on the specific hyperparameters tuned, the range or choices considered during the tuning process, and the best parameter values found. For example, the best parameters of the RF model include 64 estimators and a maximum depth of 6, while the SVM model achieved optimal performance with a C value of $1e^6$ and a gamma value of 0.00143 using an RBF kernel.

2.7. Domain adaptation learning

In our developed model, we addressed the challenge of imbalanced financial data by partitioning the information generated by the Genetic Algorithm (GA) into source and target domains. The source domain aimed to maintain a balanced representation of the majority class, while the target domain deliberately mirrored the imbalances present in the initial dataset. To establish a balanced subset for the source domain, we applied a clustering approach to the imbalanced training dataset, specifically focusing on the majority class. The cluster centroids, representing characteristic samples of the dominant class, were retained. This clustering process utilised the `imblearn` Python

Table 3
Best hyperparameters for different models.

Algorithm/Model name	Hyperparameter	Best parameter
Random forest	n_estimators: (10, 100) max_depth: (1, 20) min_samples_split: (2, 10) min_samples_leaf: (1, 10)	n_estimators: 64 max_depth: 6 min_samples_split: 2 min_samples_leaf: 1
Support vector machine	C: (1e−6, 1e+6, log-uniform) gamma: (1e−6, 1e+1, log-uniform) kernel: [linear, poly, rbf, sigmoid]	C: 1e+6 gamma: 0.00143 kernel: rbf
Logistic regression	C: (1e−6, 1e+6, log-uniform) penalty: [l1, l2] solver: [lbfgs, newton-cg, sag, saga] max_iter: (100, 1000, 10000) tol: (1e−6, 1e−3, log-uniform) fit_intercept: [True, False] class_weight: [None, balanced]	C: 1.7086 penalty: l2 solver: sag max_iter: 1000 tol: 4.4260e−5 fit_intercept: False class_weight: None
Gradient boosting	n_estimators: (10, 100) learning_rate: (1e−6, 1e+1, log-uniform) max_depth: (1, 20) min_samples_split: (2, 10) min_samples_leaf: (1, 10)	n_estimators: 42 learning_rate: 0.5144 max_depth: 7 min_samples_split: 4 min_samples_leaf: 10
k-nearest neighbours	n_neighbours: (1, 20) weights: [uniform, distance] p: [1, 2]	n_neighbours: 1 weights: uniform p: 2

package.[5] The resulting balanced subset served as the foundation for building a model with improved representativeness, capturing various patterns and features from the initially imbalanced data. The crucial step in developing a balanced source domain demonstrated a more thorough understanding of underlying patterns and made it easier for subsequent domain adaptation, which allowed the model to gain strong and widely applicable properties. It also reduced the possibility of bias towards particular cases. Conversely, the target domain was created from the unbalanced training dataset by deliberately manipulating the distribution to reflect imbalances, with a focus on the minority class. This deliberate imbalance aimed to simulate challenging circumstances for the model during adaptation. The method employed a random sampling procedure on minority class instances, ensuring that the target domain retained the intrinsic complexity and biases of the original data.

Following the creation of source and target domains, the challenge of class imbalance was addressed by introducing sample weights during model training. This proactive approach aimed to prevent the dominating majority class from overshadowing the learning process, promoting a more effective and balanced model. For the source domain, sample weights were calculated based on class frequencies, with less frequent classes receiving higher weights. In the target domain, sample weights were designed to address imbalances, assigning higher weights to the minority class. Both source and target domain sample weights underwent normalisation to maintain relative importance, ensuring their sum equalled 1. By integrating sample weights, the model prioritised minority class instances during training, enhancing its adaptability and performance in scenarios with prevalent class imbalances in the target domain.

To ensure that the selected features are evenly distributed in both the source and target domains, we implement a quantile transformation as a means of standardising these features. According to studies like Gallón, Loubes, and Maza (2013), Liu et al. (2019), Pan and Zhang (2018), Peterson and Cavanaugh (2019), this approach is preferred over alternative standardisation methods like mean normalisation or z-score scaling because of its robustness in handling outliers and non-Gaussian distributions. The provided pseudocode described in Algorithm 3 outlines the process of quantile transformation normalisation, a method used to transform data such that it follows a standard normal distribution. It starts by iterating over each feature in the dataset and sorts the values of each feature in ascending order. Then, it calculates the quantiles for each value based on its rank and the total number of samples. For each sample, it calculates its percentile based on its rank and the total number of samples and maps this percentile to a standard normal distribution using the inverse cumulative distribution function (CDF) of the normal distribution. These mapped values are stored in new arrays for each feature, and the normalised feature arrays are appended to form the normalised dataset, which is returned as the

Algorithm 3 Quantile Transformation Normalisation

Require: X: Input dataset with n samples and m features
Ensure: Normalised dataset X_{norm}
Initialise an empty array X_{norm} to store the normalised values
for $i \leftarrow 1$ to m **do** ▷ Iterate over each feature
  Sort the values of feature i in ascending order
  Calculate the quantiles for each value based on its rank and the total number of samples
  Initialise an empty array $X_{\text{norm\_feature}}$ to store the normalised values of feature i
  for $j \leftarrow 1$ to n **do** ▷ Iterate over each sample
    Calculate the rank of the jth sample in feature i
    Calculate the percentile of the jth sample based on its rank and the total number of samples
    Map the percentile to a standard normal distribution using the inverse cumulative distribution function (CDF) of the normal distribution
    Store the mapped value in $X_{\text{norm\_feature}}$
  end for
  Append $X_{\text{norm\_feature}}$ to X_{norm}
end for
return X_{norm}

[5] https://pypi.org/project/imblearn/.

output. By using quantile transformation, feature distributions between domains are more robustly and efficiently aligned.

Algorithm 4 Domain Adaptation Pipeline

Load Datasets:
  Load the source and target datasets, $X_{\text{source}}, y_{\text{source}}$ and $X_{\text{target}}, y_{\text{target}}$, respectively.
  Balance the source dataset and create an imbalanced target dataset.
Calculate Sample Weights:
  Calculate sample weights for source and target domains:
  $\text{sample\_weights\_source} \leftarrow \frac{0.5}{\sum_i (y_{\text{source}} == i)}$
  $\text{sample\_weights\_target\_train} \leftarrow \frac{0.7}{\sum_i (y_{\text{target\_train}} == i)}$
  $\text{sample\_weights\_target\_test} \leftarrow \frac{0.3}{\sum_i (y_{\text{target\_test}} == i)}$
Standardise Data:
  Standardise the features using Quantile transformation:
  $X_{\text{source\_standardised}} \leftarrow X_{\text{source}}$
  $X_{\text{target\_train\_standardised}} \leftarrow X_{\text{target\_train}}$
  $X_{\text{target\_test\_standardised}} \leftarrow X_{\text{target\_test}}$
Bayesian Optimisation:
  Iterate over classifiers and perform Bayesian optimisation.
  Find optimal hyperparameters.
Evaluate Optimised Classifiers:
  For each optimised classifier:
    Fit the classifier on the source domain.
    Transform source features.
    Train a transfer model on the target domain.
    Make predictions on the target domain testing set.

With regards to DAL, Bayesian optimisation is used to optimise the hyperparameters described in Section 2.6 by defining a parameter space tailored to the model classifiers in Section 2.5. In order to optimise based on cross-validated accuracy, each classifier's hyperparameter space is methodically explored and exploited using Bayesian optimisation. This method makes it possible to determine the ideal hyperparameters that improve each classifier's performance when it comes to DAL. Once the classifiers are optimised, a model transfer is executed by fitting these fine-tuned classifiers to the source domain data. After that, the pertinent characteristics are taken out of the source domain and used to train a new model on the unbalanced target domain. The main focus here is on using the data from the source domain to modify the model to fit the specifics of the target domain, which will in turn improve performance on the testing set for the target domain. Algorithm 4 depicts a comprehensive pipeline for DAL.

This proposed process involves several key steps aimed at leveraging knowledge from relevant source domains to adapt machine learning models to work effectively in the target domain.

3. Performance metrics

3.1. Confusion metrics

In assessing the performance of a classification model, such as in our DAL classifier for bankruptcy prediction, the evaluation is based on the test data, and the results are presented using a confusion matrix.

The confusion matrix, denoted as $\mathcal{M}(\varphi)$ for a model classifier φ, is defined as follows:

$$\mathcal{M}(\varphi) = \sum_{i,j}^{t} m_{ij}(\varphi), \tag{21}$$

where $\mathcal{M}(\varphi)$ is a $t \times t$ square matrix for a target domain test set with t target labels. Each entry $m_{ij}(\varphi)$ of $\mathcal{M}(\varphi)$ represents the number of values belonging to a target label i but have been assigned to a different target label j by φ. Specific computations derived from Eq. (21) include:

a. $\sum_{i,j=1}^{t} m_{i1}(\varphi)$, the sum of all values of target label $i \in t$.
b. $\sum_{i=1,j}^{t} m_{1j}(\varphi)$, the sum of all values of target label $j \in t$.
c. $\sum_{i,j=i}^{t} m_{ii}(\varphi)$, depicting the crosswise field that validly classifies all the target labels.

The introduction of Eq. (21) provides a fundamental framework for comprehending the performance metrics of our classification model. This formula is essential because it captures the essence of a confusion matrix, which is a vital instrument for evaluating the precision and effectiveness of the model. Each entry of this matrix signifies the instances where the model correctly or incorrectly assigned a label, forming the basis for various performance measures.

Given our binary classification focus, the confusion matrix becomes a 2×2 matrix, encompassing true positives (TPs), false positives (FPs), false negatives (FNs), and true negatives (TNs). These components denote the counts of correctly identified positive (no bankruptcy) and negative (bankruptcy) instances and their misclassifications, respectively. For clarity, in our binary classification scenario: "Positive" signifies the absence of bankruptcy (no bankruptcy), whilst "Negative" corresponds to the presence of bankruptcy. Table 4 summarises various classification measures derived from the confusion matrix, such as sensitivity, specificity, precision, recall, and accuracy, along with their respective computations (Hossin & Sulaiman, 2015).

3.2. Probability calibration

Probability calibration is a concept in machine learning that refers to the alignment of predicted probabilities with the true likelihood of the corresponding events. In classification tasks, machine learning models often output predicted probabilities that represent the model's confidence in its predictions. Ideally, these predicted probabilities should accurately reflect the actual probabilities of the events being predicted. One common method for probability calibration is Platt scaling (Böken, 2021; Niculescu-Mizil & Caruana, 2005), which involves fitting a logistic regression model to the predicted probabilities generated by the original model. This additional calibration step can help in refining the predicted probabilities to be more accurate. The mathematical formula for Platt scaling can be expressed as follows:

Let $\Lambda(x)$ be the output of your machine learning model (before calibration) for a given instance x. $\Lambda(x)$ is often a raw score or logit. The calibrated probability (p_c) is then obtained using the sigmoid (logistic) function defined in Eq. (22):

$$p_c(x) = \frac{1}{1 + e^{(\alpha \cdot \Lambda(x) + \beta)}} \tag{22}$$

where $p_c(x)$ is the calibrated probability, $\Lambda(x)$ is the output of the original model, α and β are the parameters of the logistic regression model, which are learned during the calibration process. During the calibration process, we typically use a set of labelled data to train the function in Eq. (22). The labels would be the true class labels (0 or 1), and the input to the logistic function would be $\Lambda(x)$. The logistic regression model is trained to minimise the log-likelihood of the true labels given the calibrated probabilities. Once the model is trained, the learned parameters (α and β) are used to calibrate new predicted probabilities.

The Brier score is a metric used to assess the calibration of probabilistic predictions. For binary classification, the Brier score is calculated as the mean squared difference between the predicted probabilities and the actual outcomes. The formula for the Brier score is as follows:

$$\text{Brier score} = \frac{1}{N} \sum_{i=1}^{N} (y_i - p_i)^2 \tag{23}$$

Table 4
Classification measures for DAL classifier performance evaluation.

Name	Description	Computation
Sensitivity	The ability of the classifier to correctly identify all positive scenarios.	$TP/(TP+FN)$
Specificity	The ability of the classifier to correctly reject all negative scenarios.	$TN/(TN+FP)$
Precision	The ratio of relevant scenarios correctly identified by the classifier.	$TP/(TP+FP)$
Recall	The ratio of all relevant scenarios correctly identified by the classifier	$TP/(TP+FN)$
Accuracy	The overall ability of the classifier to make correct decisions, considering both positive and negative scenarios.	$\frac{TP+TN}{TP+TN+FP+FN}$

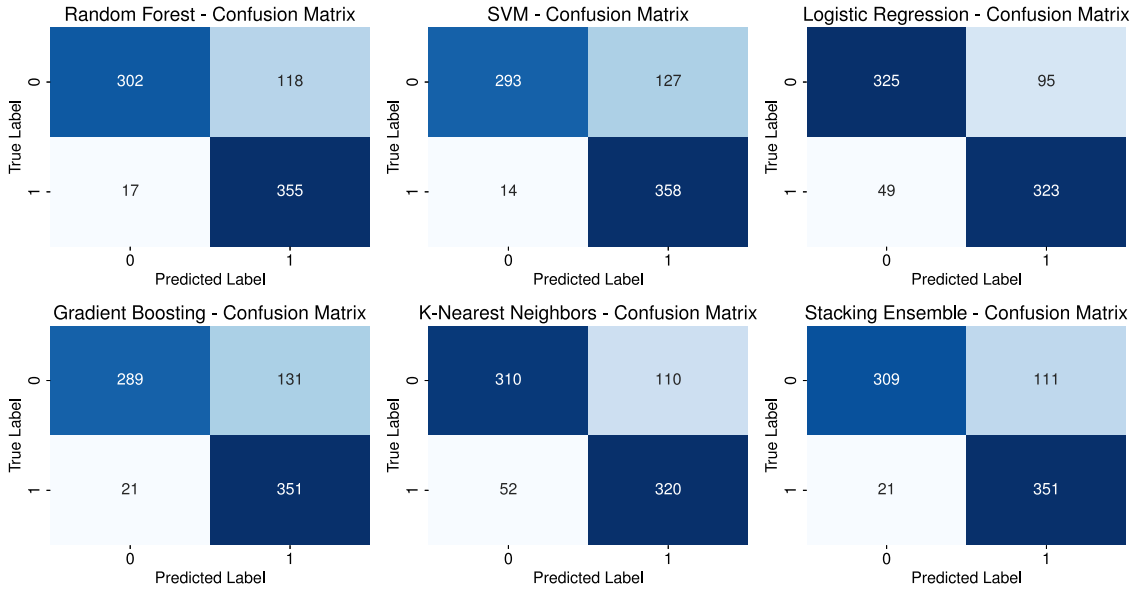


Fig. 4. Comparative confusion matrices of bankruptcy prediction models in DAL: The performance differences between the classifiers (RF, SVM, LR, GB, *k*-NN, and SE) in recognising cases that are bankrupt and those that are not are depicted in this figure. Notable patterns include SVM and LR exhibiting good results, and RF and SE demonstrating improved sensitivity with larger true positives and reduced false positives. On the other hand, *k*-NN and GB show comparatively more false positives.

where N is the number of instances in the dataset, y_i is the true binary outcome (0 or 1) for instance i, p_i is the predicted probability for the positive class for instance i. In the case of a multi-class classification problem, we can generalise Eq. (23) to get Eq. (24):

$$\text{Brier Score} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}(y_{ik} - p_{ik})^2 \tag{24}$$

where K is the number of classes, y_{ik} is an indicator variable equal to 1 if the true class for instance i is k, and 0 otherwise, p_{ik} is the predicted probability for class k for instance i. This metric Score ranges from 0 to 1, where a lower score indicates better calibration (better alignment of predicted probabilities with actual outcomes).

4. Results and discussion

The approach adopted in this work uses a 5-fold cross-validation strategy within the Bayesian optimisation process to ensure reliable and generalisable hyperparameter tuning and model evaluation. This method divides the dataset into five subsets, iteratively using four subsets for training and one for validation. This reduces the risk of overfitting and provides a comprehensive assessment of model performance across different data splits. Additionally, by employing balanced source and target datasets and applying sample weighting and resampling techniques such as ClusterCentroids, RandomOverSampler, and RandomUnderSampler, the validation process is further refined. These steps effectively address class imbalance issues, ensuring that evaluation metrics like accuracy, ROC AUC, and classification reports accurately reflect the models' capabilities in handling imbalanced data scenarios within the domain adaptation framework. The discussion that follows provides insights into the model's performance and its implications for stakeholders in the financial domain. For ease of comprehension, in our binary classification scenario: "0" represents the negative class or the absence of the event being measured (non-bankrupt), whilst "1" denotes the positive class or the presence of the event being measured (bankrupt).

In Fig. 4, multiple classifiers, including RF, SVM, LR, GB, *k*-NN, and a stacking ensemble (SE), exhibited varying confusion matrices. Take note that the SE algorithm amalgamates predictions from multiple base classifiers–in our case, SVM, LR, GB, and *k*-NN. These predictions are then processed by a meta-learner, with RF serving as the meta-learner in this case. The meta-learner is trained on the predictions gleaned by the base classifiers to make the final prediction. The consistent trend across these matrices reveals that the models generally perform well in identifying true negatives (non-bankrupt companies) but show differences in their ability to correctly classify bankrupt instances. Notably, RF and SE models produced higher true positives (355 and 351, respectively) and lower false positives (118 and 111, respectively), indicating a better ability to identify companies at financial risk. SVM and LR also demonstrated favourable results, while *k*-NN and GB showed relatively higher false positives, potentially leading to unnecessary concerns for non-bankrupt entities. Stakeholders in the financial sector should consider these nuances when selecting a model, with a focus on minimising false negatives to avoid overlooking actual bankruptcies. The absence of evident class-imbalance in the confusion matrix suggests that the classification models are performing well in recognising both bankrupt and non-bankrupt cases without a significant bias towards one class over the other. This balanced performance is crucial for accurate predictions and indicates the effectiveness of the models in handling the imbalanced target data.

Fig. 5 reveals that RF and the SE models achieved well-rounded performance across all metrics (precision, recall, and F1-score) for

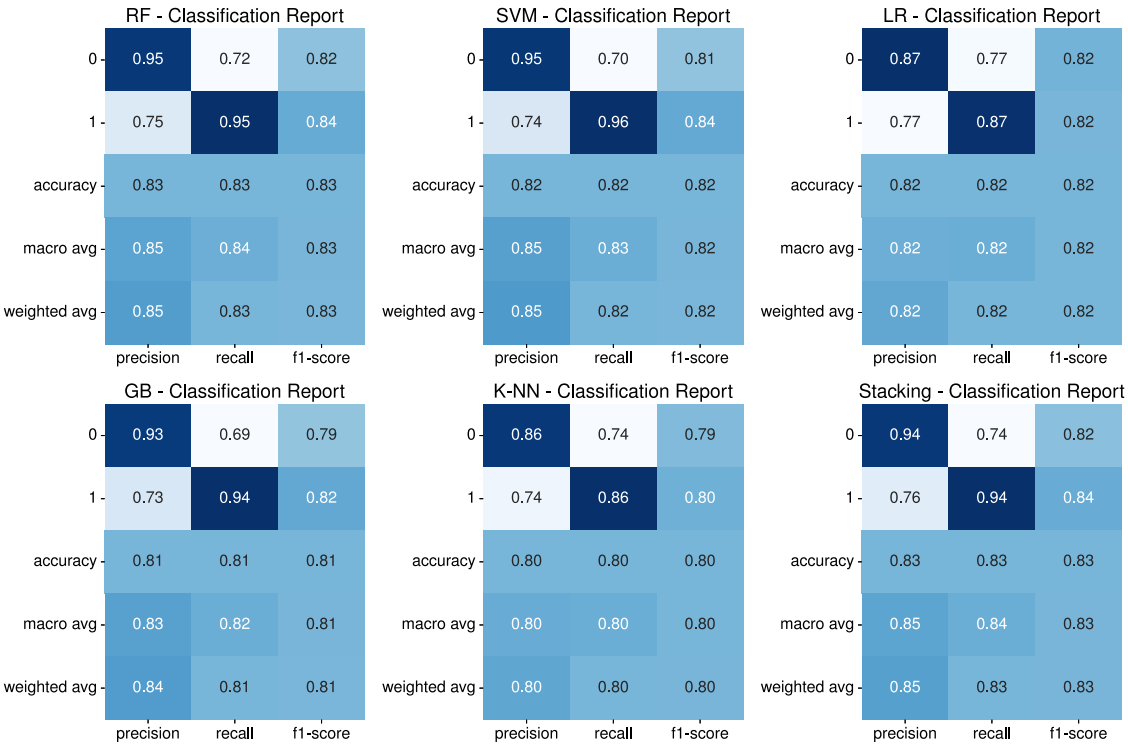


Fig. 5. Comparative analysis of classification results for bankruptcy prediction models. This figure presents performance insights for RF, SVM, LR, GB, *k*-NN, and a SE based on precision, recall, and F1-score metrics for both non-bankrupt (0) and bankrupt (1) classes.

both bankrupt and non-bankrupt classifications. This suggests their effectiveness in accurately identifying companies in financial distress and those that are financially stable. While SVM and LR also performed well, they exhibited a trade-off between precision and recall. SVM prioritised identifying most bankrupt companies (high recall) even if it meant misclassifying some healthy ones (lower precision). Conversely, LR focused on avoiding false positives (high precision for non-bankrupt companies) at the expense of potentially missing some bankrupt firms (lower recall). The GB followed a similar trend to LR, demonstrating high precision for non-bankrupt companies but lower recall for bankrupt ones. *k*-NN achieved a reasonable balance but fell slightly behind RF and SE in both precision and recall. For financial institutions, these results suggest that RF and SE might be preferable choices due to their balanced performance in identifying both bankrupt and non-bankrupt companies. However, the optimal model selection should be tailored to specific priorities and operational needs. Financial stakeholders should carefully consider the potential consequences of misclassifications. A high tolerance for false positives (mistakenly classifying a healthy company as bankrupt) might favour models with high precision like LR. Conversely, situations where missing a truly bankrupt company (false negative) is more concerning might call for models with high recall like SVM. Ultimately, the trade-off between precision and recall should be weighed based on risk tolerance and the potential impact of misclassifications.

The ROC curve results showcased in Fig. 6 for various models, indicate their discrimination ability in distinguishing between bankrupt and non-bankrupt instances. The mean Area Under the Curve (AUC) values displayed in the ROC curve provide a summary of each model's overall performance across multiple evaluations. Higher AUC values generally suggest better model performance in terms of correctly ranking instances by their predicted probabilities. RF and SVM exhibit the highest mean AUC, indicating robust discriminative power. LR, GB, and the SE also perform well, with AUC values in the high 80 s to low 90 s. *k*-NN, while still respectable, shows a slightly lower mean AUC. These AUC results suggest that RF and SVM are particularly strong performers in terms of distinguishing between bankrupt and non-bankrupt cases based on their predicted probabilities. The choice between these models may depend on other considerations, such as interpretability and computational efficiency. LR, GB, and the SE also demonstrate reliable discriminatory capabilities, making them suitable alternatives. *k*-NN, while showing decent performance, has a slightly lower mean AUC and may require additional scrutiny.

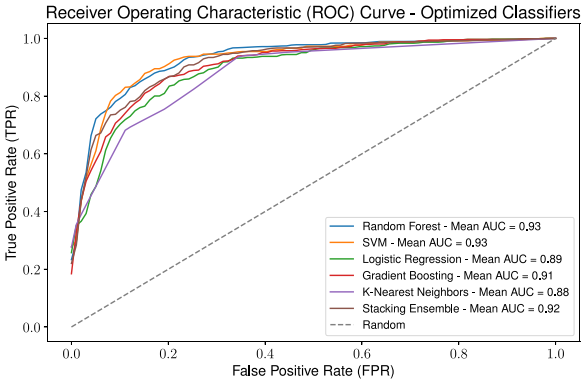


Fig. 6. ROC curve analysis of DAL models for bankruptcy prediction: The ROC curve results showcase the discrimination abilities of RF, SVM, LR, GB, *k*-NN, and a SE. The AUC values quantify overall model performance, with RF and SVM demonstrating the highest AUC (93%).

In the graphical representation (Fig. 7), the decision boundary is a crucial aspect of understanding how machine learning models make predictions by separating different classes in the feature space. The decision boundary of RF (first graph from the left) is nonlinear and can adapt well to intricate patterns. In this graph, we can observe that RF creates a piecewise constant decision boundary as it combines multiple decision trees to form a consensus prediction. The SVM plot (second graph from the left) aims to find a hyperplane that maximally separates classes in the feature space. In the context of the high AUC results,

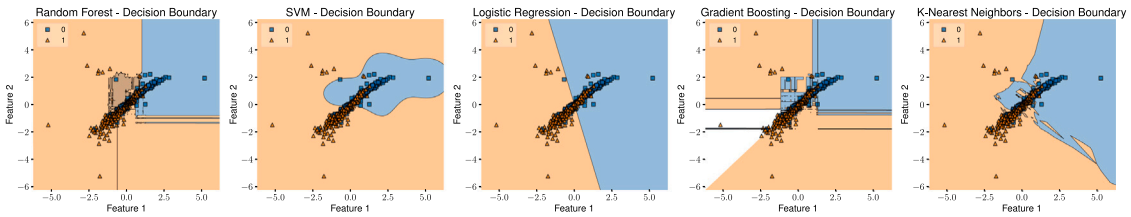


Fig. 7. Decision boundaries of DAL models for bankruptcy prediction: This figure illustrates the decision boundaries of RF (first from the left), SVM (second from the left), LR (third from the left), GB (fourth from the left), and *k*-NN (last from the left) based on their performance characteristics. RF and SVM are expected to have flexible, potentially nonlinear boundaries, while LR exhibits linear structures. GB combines simple decision boundaries, contributing to a complex overall boundary. *k*-NN's locally adaptive boundary is influenced by the data point distribution. Feature 1 refers to the foremost significant financial ratio, 'Return on Assets (ROA) Before Interest and % After Tax', while Feature 2 represents the subsequent crucial financial ratio for predicting bankruptcy, 'Net Income to Stockholder's Equity'.

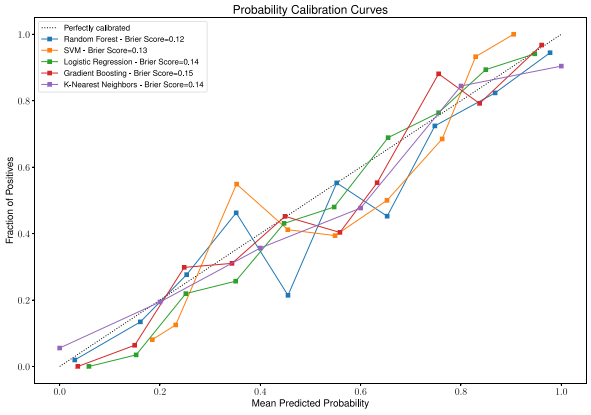


Fig. 8. Illustration of calibration curves and Brier scores for binary classifiers. The Brier scores (RF: 0.12, SVM: 0.13, LR: 0.14, GB: 0.15, *k*-NN: 0.14) indicate well-calibrated probabilistic predictions. The calibration curves showcase the alignment of predicted probabilities with true positive class probabilities.

SVM establishes a discriminative hyperplane that effectively separates bankrupt and non-bankrupt instances. Here, the decision boundary is nonlinear, because the best kernel function we chose was 'rbf'. LR models the log-odds of the probability of belonging to a particular class. The decision boundary for LR is a linear function of the input features. Since the relationship between features and the log-odds is assumed to be linear, the decision boundary is a hyperplane. GB builds an ensemble of weak learners, typically decision trees, sequentially, with each tree aiming to correct the errors of the previous ones. The decision boundary is a combination of simpler decision boundaries, leading to a complex and adaptive overall boundary. The *k*-NN classifies instances based on the majority class among its *k* nearest neighbours. The decision boundary is flexible and nonlinear, adapting to the local density of instances in the feature space.

A well-calibrated model should produce predicted probabilities that are indicative of the true probability of belonging to the positive class. One way to assess calibration is by examining a calibration curve and calculating the Brier score, as illustrated in Fig. 8. The Brier scores for each model are as follows: RF with a Brier score of 0.12, SVM with 0.13, LR with 0.14, GB with 0.15, and *k*-NN with 0.14. These scores are relatively low, suggesting that the models provide well-calibrated probabilistic predictions. To gain further insights into the behaviour of each classifier, we analyse the distribution of samples across predicted probability bins. For example, dividing the predicted probabilities into bins (e.g., $0-0.1, 0.1-0.2, \ldots, 0.9-1.0$) and counting the number of samples in each bin would reveal how concentrated or dispersed the predicted probabilities are. In general, a well-calibrated model would exhibit a calibration curve that closely follows the diagonal line ($y = x$) and this is exactly what the calibration curves exhibit.

The histograms shown in Fig. 9 visually illustrate the central tendencies and spatial distribution of predicted probabilities for each model (RF, SVM, LR, GB, and *k*-NN) across all instances. In general, a well-calibrated model's histogram of mean predicted probabilities would ideally exhibit a balanced distribution. Specifically, a peak or concentration around 0.5 indicates that the model is uncertain about the class assignment for many instances, while values closer to 0 or 1 signify higher confidence in the predicted class. Models that are poorly calibrated may exhibit skewed histograms, indicating a mismatch between predicted probabilities and the true likelihood of positive outcomes. This aligns with the concept of calibration discussed by Kull, Filho, and Flach (2017). For RF (first from the left) and SVM (second from the left), both models are known for their discriminative power; therefore, we expect to observe a histogram with a clear peak or concentration towards the extremes (0 or 1). This suggests that these models are confident in their predictions and have effectively separated instances into distinct classes. In the case of LR, the model exhibits a smoother and more spread-out histogram, reflecting its inherent probabilistic nature. The mean predicted probability histogram for GB demonstrates a more refined distribution compared to models like RF. Hence, we expect to observe a histogram with peaks or concentrations that reflect the model's ability to iteratively improve its predictions. Peaks at 0 or 1 in the histogram are indicative of instances where boosting rounds consistently strengthen a specific class assignment, signifying a higher level of confidence in the predictions. For *k*-NN, which is based on local patterns in the data, produces a histogram with a more varied distribution. Instances where the nearest neighbours are consistently of the same class would result in peaks at 0 or 1, indicating higher confidence, while instances with mixed neighbours might lead to a peak around 0.5.

Tables 5 and 6 present a comprehensive comparison of traditional machine learning approaches and DAL techniques to predict bankruptcy for Taiwanese and Polish companies, respectively. The evaluation metrics used provide information about the performance of the models under different conditions. In Table 5, accuracy is generally higher in traditional learning in most classifiers. For example, the RF classifier achieves an accuracy of 0.95 in traditional learning, compared to 0.82 in DAL. Similar trends are observed for SVM, LR, and GB. Precision for Class 0 remains consistently high in both learning paradigms, with traditional learning showing a slight edge. For example, SVM has a precision of 0.96 in traditional learning versus 0.95 in DAL. However, DAL occasionally shows higher precision for Class 1, such as with LR (0 vs. 0.77). Recall for Class 1 is notably poor in traditional learning, as seen with SVM (0) compared to DAL (0.96). This indicates that traditional models struggle with identifying Class 1 instances. Domain adaptation techniques significantly improve recall for both Classes across all classifiers. The F1 score for Class 1 in DAL is considerably higher than in traditional learning. To illustrate, SVM's F1 score is 0.36 in traditional learning compared to 0.82 in DAL, highlighting the improved balance between precision and recall. The DAL method demonstrates competitive AUC-ROC values and improved calibration (Brier scores), suggesting better reliability and discrimination in predictions under shifting data distributions. The accuracy remains higher in traditional learning for Polish bankruptcy data as well, with SVM

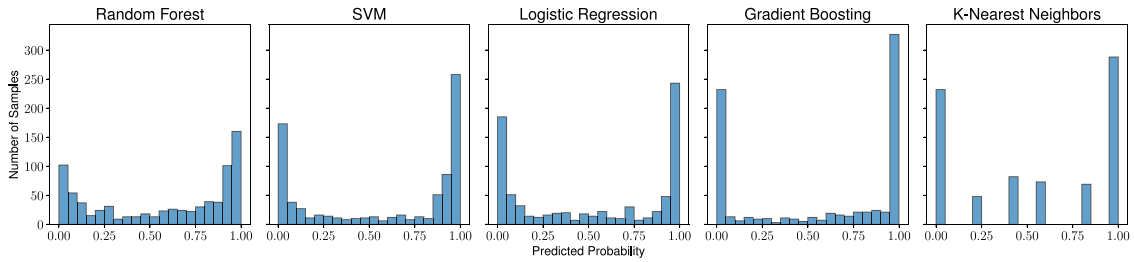


Fig. 9. Histograms depicting the central tendencies and spatial distribution of predicted probabilities for RF, SVM, LR, GB, and *k*-NN models across all instances. Well-calibrated models exhibit a balanced distribution around 0.5, indicating uncertainty in class assignment, while values closer to 0 or 1 signify higher confidence.

Table 5
Performance metrics comparison between traditional learning and domain adaptation Learning (Taiwanese bankruptcy prediction dataset).

Metric	Traditional learning						Domain adaptation learning					
	SVM	RF	LR	GBoost	K-NN	SE	SVM	RF	LR	GBoost	K-NN	SE
Accuracy	0.88	0.95	0.84	0.97	0.96	0.97	0.82	0.83	0.82	0.81	0.80	0.83
Precision (Class 0)	0.96	0.96	0.96	0.97	0.96	0.97	0.95	0.95	0.87	0.93	0.86	0.94
Precision (Class 1)	0.00	0.00	0.00	0.75	0.00	0.76	0.74	0.75	0.77	0.73	0.74	0.76
Recall (Class 0)	0.91	0.99	0.87	1.00	0.99	0.97	0.70	0.72	0.77	0.69	0.74	0.74
Recall (Class 1)	0.00	0.00	0.00	0.24	0.00	0.25	0.96	0.95	0.87	0.94	0.86	0.94
F1 Score (Class 0)	0.93	0.97	0.91	0.98	0.98	0.97	0.81	0.82	0.82	0.79	0.79	0.82
F1 Score (Class 1)	0.00	0.00	0.00	0.36	0.00	0.38	0.84	0.84	0.82	0.82	0.80	0.84
AUC-ROC	0.49	0.56	0.30	0.93	0.45	0.94	0.93	0.93	0.89	0.91	0.88	0.92
Calibration (Brier)	0.15	0.16	0.14	0.17	0.18	–	0.13	0.12	0.14	0.15	0.14	–

Table 6
Performance metrics comparison between traditional learning and domain adaptation Learning (Polish bankruptcy prediction dataset).

Metric	Traditional learning						Domain adaptation learning					
	SVM	RF	LR	GB	K-NN	SE	SVM	RF	LR	GBoost	K-NN	SE
Accuracy	0.98	0.98	0.98	0.98	0.99	98	0.72	0.73	0.79	0.73	0.73	0.79
Precision (Class 0)	0.98	0.98	0.98	0.99	0.98	98	0.91	0.94	0.84	0.81	0.66	0.88
Precision (Class 1)	1.00	0.90	0.88	0.90	0.92	80	0.65	0.66	0.74	0.68	0.88	0.73
Recall (Class 0)	1.00	1.00	1.00	1.00	1.00	0.99	0.49	0.50	0.70	0.60	0.93	0.66
Recall (Class 1)	0.15	0.21	0.18	0.31	0.14	0.20	0.95	0.96	0.87	0.86	0.52	0.91
F1 Score (Class 0)	0.99	0.99	0.99	0.99	0.99	0.99	0.64	0.66	0.77	0.69	0.77	0.76
F1 Score (Class 1)	0.27	0.35	0.30	0.46	0.25	0.23	0.77	0.78	0.80	0.76	0.66	0.81
AUC-ROC	0.87	0.86	0.88	0.85	0.84	0.87	0.82	0.86	0.86	0.82	0.92	0.86
Calibration (Brier)	0.25	0.16	0.15	0.17	0.18	–	0.12	0.18	0.16	0.12	0.17	–

achieving 0.98 compared to 0.72 in DAL. This trend persists across most classifiers. Precision for Class 0 remains high across both paradigms. However, recall for Class 1 shows a significant improvement in DAL. As an example, SVM's recall for Class 1 is 0.95 in domain adaptation versus 0.15 in traditional learning, indicating a better identification of the minority class. The F1 score for Class 1 is substantially higher in domain adaptation learning (e.g., RF with 0.78) compared to traditional learning (e.g., RF with 0.35), underscoring the enhanced performance in balancing precision and recall for the minority class. The comparison between Tables 5 and 6 reveals consistent trends across different datasets (Taiwanese and Polish bankruptcy data). DAL consistently improves recall and F1 scores for the minority class (that is, Class 1 in both Taiwanese and Polish data), addressing class imbalance and distributional shift issues better than traditional learning methods. However, traditional learning maintains higher accuracy, indicating better overall prediction accuracy in more stable data environments.

5. Conclusion

Our study has shown that the combination of DAL and GA techniques significantly improves the predictive power of bankruptcy models. There are several reasons why combining DAL with GA can achieve better performance.

First and foremost, one of the main reasons this hybrid approach is viable is that DAL deals with real-world data distributional shifts. By leveraging knowledge from a source domain and adapting the model to a target domain, DAL enhances the generalisation and adaptability of the predictive model to diverse financial environments. Such adaptability is highly important in the sense that it mitigates the impact brought about by the change of circumstances and assures that the model is robust in a dynamic business context.

Additionally, the incorporation of GA in the feature selection process optimises the model's ability to identify relevant financial factors that directly influence bankruptcy prediction. The evolutionary nature of GA enables the model to search for the most informative features, leading to a more accurate and efficient prediction process. The synergy between DAL and GA not only improves the model's predictive accuracy but also enhances its adaptability and flexibility in handling complex financial variables. By dynamically adjusting to shifting circumstances and maintaining strong performance, the hybrid model offers valuable insights into the intricate patterns of bankruptcy risk. We summarise the main findings of this research as follows:

i. Computing the median and MAD within each subsegment allows the BCP-HI algorithm to robustly estimate the central tendency and variability of the data, making it well-equipped to detect outliers effectively and accurately.
ii. The GA method provided valuable insights into the financial factors contributing to the risk of bankruptcy. In Fig. 3, we observe that Roa (A) before interest and % after tax, Net income to

stockholder's equity, Net worth/Assets, Borrowing dependency, Roa (B) Before interest and depreciation after tax, Debt ratio, Roa (C) before interest and depreciation before interest, Persistent Eps in the last four seasons, Current Liabilities/Equity, and Retained earnings to total assets are the top 10 features essential for stakeholders to make informed decisions and understand the specific indicators influencing the financial health of a company.

iii. The SE and RF exhibit balanced performance, making them potentially preferable for stakeholders seeking a refined approach.

iv. RF exhibits a nonlinear decision boundary capable of adapting to intricate patterns, while SVM establishes a discriminative hyperplane. Understanding these characteristics aids in interpreting how each model makes predictions in the feature space, providing insights for stakeholders on the model's interpretability and suitability for specific scenarios.

v. The calibration analysis, including calibration curves and Brier scores, indicated that the hybrid model produced well-calibrated probabilistic predictions. This implies that the predicted probabilities align closely with the true likelihood of positive outcomes, contributing to the model's reliability and effectiveness in real-world scenarios.

The study's innovation lies in combining GA, DAL, Bayesian Optimisation, and six classifiers to improve bankruptcy prediction despite distributional changes. While these methods are common individually, their synergistic use enhances the models' predictive power and adaptability. Here's a breakdown of the novelty:

i. *Combination of methods*: The study's originality lies in how it combines these methods to create an effective predictive model. GA optimises feature selection, DAL adapts to different financial environments, Bayesian optimisation tunes hyperparameters, and six classifiers provide diverse models to capture various data aspects.

ii. *Synergistic approach*: The innovative aspect is the synergy between these methods. This combination results in a robust and flexible predictive model that can handle changing conditions effectively.

iii. *Addressing specific research gap*: The study addresses the challenge of distributional changes in bankruptcy prediction, a gap not adequately covered by traditional methods. Integrating GA, DAL, and Bayesian Optimisation offers a tailored solution to enhance model accuracy and adaptability in dynamic financial environments.

iv. *Innovative methodological pipeline*: The study's originality is also in its methodological pipeline, which leverages the strengths of each method coherently, showcasing innovative thinking in model development.

Although our research contributes to addressing the research gap in bankruptcy prediction, there are still avenues for further exploration. Future studies could investigate the applicability of the hybrid model to different industries or explore the integration of additional techniques to enhance its performance. Furthermore, as discussed in Radovanovic and Haas (2023), we could improve the robustness of the hybrid model by integrating evaluation metrics that consider both the financial costs of bankruptcy and its social impacts, using job losses as a proxy.

In conclusion, while traditional learning methods show higher accuracy, DAL proves to be a more robust technique for dealing with real-world data distributional shifts, as evidenced by improved recall and F1 scores, particularly for the minority class. This indicates that domain adaptation is better suited for scenarios where the data distribution changes over time or differs between training and testing environments.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

For our analysis, we utilised two datasets from the University of California, Irvine machine learning repository: the Taiwanese Bankruptcy Prediction dataset and the Polish Companies Bankruptcy data.

Acknowledgements

The authors express their appreciation to the anonymous referees for their invaluable comments. Additionally, TA-N extends gratitude for the support received from the Ghana Space Science and Technology Institute (GSSTI) and for access to their High-Performance Computing (HPC) system. Special thanks to Elsie and Rubby of GSSTI for proofreading the paper.

References

Almaskati, N., Bird, R., Yeung, D., & Lu, Y. (2021). A horse race of models and estimation methods for predicting bankruptcy. *Advances in Accounting*, *52*, Article 100513.

Alsultanny, Y. A., & Aqel, M. M. (2003). Pattern recognition using multilayer neural-genetic algorithm. *Neurocomputing*, *51*, 237–247.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, *23*(4), 589–609.

Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 international conference on computing networking and informatics* (pp. 1–9). IEEE.

Barboza, F., Kimura, H., & Altman, E. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications*, *83*, 405–417.

Bhanu, B., Lee, S., & Ming, J. (1995). Adaptive image segmentation using a genetic algorithm. *IEEE Transactions on Systems, Man and Cybernetics*, *25*(12), 1543–1567.

Böken, B. (2021). On the appropriateness of Platt scaling in classifier calibration. *Information Systems*, *95*, Article 101641.

Carmona, P., Climent, F., & Momparler, A. (2019). Predicting failure in the US banking sector: An extreme gradient boosting approach. *International Review of Economics & Finance*, *61*, 304–323.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785–794). ACM, http://dx.doi.org/10.1145/2939672.2939785.

Chen, H.-L., Yang, B., Wang, G., Liu, J., Xu, X., Wang, S.-J., et al. (2011). A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method. *Knowledge-Based Systems*, *24*(8), 1348–1359.

Chi, L.-C., & Tang, T.-C. (2006). Bankruptcy prediction: Application of logit analysis in export credit risks. *Australian Journal of Management*, *31*(1), 17–27.

Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., et al. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing* (pp. 1–4). Springer.

Cunningham, P., & Delany, S. J. (2020). k-Nearest neighbour classifiers: (with Python examples). arXiv preprint arXiv:2004.04523.

Ding, X., Liu, J., Yang, F., & Cao, J. (2021). Random radial basis function kernel-based support vector machine. *Journal of the Franklin Institute*, *358*(18), 10121–10140.

Fahlevi, M. R., & Marlinah, A. (2018). The influence of liquidity, capital structure, profitability and cash flows on the company's financial distress. *Jurnal Bisnis dan Akuntansi*, *20*(1), 59–68.

Farahani, A., Voghoei, S., Rasheed, K., & Arabnia, H. R. (2021). A brief review of domain adaptation. In *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020* (pp. 877–894). Springer.

Friedman, J. H. (1999). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378, Nonlinear Methods and Data Mining.

Gallón, S., Loubes, J.-M., & Maza, E. (2013). Statistical properties of the quantile normalization method for density curve alignment. *Mathematical Biosciences*, *242*(2), 129–142.

Ghanea-Hercock, R. (2003). Principles of natural evolution. In *Applied evolutionary algorithms in java*. New York, NY: Springer, http://dx.doi.org/10.1007/978-0-387-21615-7_2.

Guan, H., & Liu, M. (2021). Domain adaptation for medical image analysis: a survey. *IEEE Transactions on Biomedical Engineering*, *69*(3), 1173–1185.

Heo, J., & Yang, J. Y. (2014). AdaBoost based bankruptcy forecasting of Korean construction companies. *Applied Soft Computing*, *24*, 494–499.

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 1.

Inagaki, J., Haseyama, M., & Kitajima, H. (1999). A genetic algorithm for determining multiple routes and its applications. *Vol. 6*, In *1999 IEEE international symposium on circuits and systems* (pp. 137–140). IEEE.

Jiang, J., & Zhai, C. (2007). *Instance weighting for domain adaptation in NLP*. ACL.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*.

Kim, H.-D., Park, C.-H., Yang, H.-C., & Sim, K.-B. (2006). Genetic algorithm based feature selection method development for pattern recognition. In *2006 SICE-ICASE international joint conference* (pp. 1020–1025). IEEE.

Kliestik, T., Vrbka, J., & Rowland, Z. (2018). Bankruptcy prediction in Visegrad group countries using multiple discriminant analysis. *Equilibrium. Quarterly Journal of Economics and Economic Policy*, *13*(3), 569–593.

Kristóf, T., & Virág, M. (2020). A comprehensive review of corporate bankruptcy prediction in Hungary. *Journal of Risk and Financial Management*, *13*(2), 35.

Kull, M., Filho, T. M. S., & Flach, P. (2017). Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, *11*(2), 5052–5080.

Li, W. (2004). Using genetic algorithm for network intrusion detection. *Proceedings of the United States Department of Energy Cyber Security Group*, *1*, 1–8.

Li, H., Lee, Y.-C., Zhou, Y.-C., & Sun, J. (2011). The random subspace binary logit (RSBL) model for bankruptcy prediction. *Knowledge-Based Systems*, *24*(8), 1380–1388.

Li, H., Liang, Y., & Xu, Q. (2009). Support vector machines and its applications in chemistry. *Chemometrics and Intelligent Laboratory Systems*, *95*(2), 188–198.

Li, Y., & Wang, Y. (2017). Machine learning methods of bankruptcy prediction using accounting ratios. *Open Journal of Business and Management*, *6*(1), 1–20.

Liang, D., Lu, C.-C., Tsai, C.-F., & Shih, G.-A. (2016). Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. *European Journal of Operational Research*, *252*(2), 561–572.

Liang, D., Tsai, C.-F., Dai, A.-J., & Eberle, W. (2018). A novel classifier ensemble approach for financial distress prediction. *Knowledge and Information Systems*, *54*(2), 437–462.

Liu, X., Li, N., Liu, S., Wang, J., Zhang, N., Zheng, X., et al. (2019). Normalization methods for the analysis of unbalanced transcriptome data: a review. *Frontiers in Bioengineering and Biotechnology*, *7*, 358.

Luo, J., Yan, X., & Tian, Y. (2020). Unsupervised quadratic surface support vector machine with application to credit risk assessment. *European Journal of Operational Research*, *280*(3), 1008–1017.

Maia Polo, F., Izbicki, R., Lacerda, E. G., Ibieta-Jimenez, J. P., & Vicente, R. (2023). A unified framework for dataset shift diagnostics. *Information Sciences*, *649*, Article 119612. http://dx.doi.org/10.1016/j.ins.2023.119612, URL: https://www.sciencedirect.com/science/article/pii/S0020025523011970.

Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern Recognition*, *33*(9), 1455–1465.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133.

Min, S.-H., Lee, J., & Han, I. (2006). Hybrid genetic algorithms and support vector machines for bankruptcy prediction. *Expert Systems with Applications*, *31*(3), 652–660.

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning* (pp. 625–632).

Nolfi, S., Floreano, D., Miglino, O., Mondada, F., et al. (1994). How to evolve autonomous robots: Different approaches in evolutionary robotics. In *Artificial life IV* (pp. 190–197). Boston, MA.

Nyitrai, T., & Virág, M. (2019). The effects of handling outliers on the performance of bankruptcy prediction models. *Socio-Economic Planning Sciences*, *67*, 34–42.

Odom, M. D., & Sharda, R. (1990). A neural network model for bankruptcy prediction. In *1990 IJCNN international joint conference on neural networks* (pp. 163–168). IEEE.

Olson, D. L., Delen, D., & Meng, Y. (2012). Comparative analysis of data mining methods for bankruptcy prediction. *Decision Support Systems*, *52*(2), 464–473.

Osuna, E., Freund, R., & Girosit, F. (1997). Training support vector machines: an application to face detection. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition* (pp. 130–136). IEEE.

Pal, S. K., & Wang, P. P. (1996). *Genetic algorithms for pattern recognition*. USA: CRC Press.

Pan, M., & Zhang, J. (2018). Quantile normalization for combining gene-expression datasets. *Biotechnology & Biotechnological Equipment*, *32*(3), 751–758.

Pehlivan, H. (2024). A novel outlier detection method based on Bayesian change point analysis and Hampel identifier for GNSS coordinate time series. *EURASIP Journal on Advances in Signal Processing*, *2024*(1), 44.

Peng, X., & Xu, D. (2013). Robust minimum class variance twin support vector machine classifier. *Neural Computing and Applications*, *22*, 999–1011.

Peterson, R. A., & Cavanaugh, J. E. (2019). Ordered quantile normalization: a semiparametric transformation built for the cross-validation era. *Journal of Applied Statistics*.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 6639–6649). Red Hook, NY, USA: Curran Associates Inc..

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2022). *Dataset shift in machine learning*. Mit Press.

Radovanovic, J., & Haas, C. (2023). The evaluation of bankruptcy prediction models based on socio-economic costs. *Expert Systems with Applications*, *227*, Article 120275.

Saitoh, F. (1999). Image contrast enhancement using genetic algorithm. *Vol. 4*, In *IEEE SMC'99 conference proceedings. 1999 IEEE international conference on systems, man, and cybernetics (cat. no. 99CH37028)* (pp. 899–904). IEEE.

Sharma, P., Banerjee, S., Tiwari, D., & Patni, J. C. (2021). Machine learning model for credit card fraud detection-a comparative analysis. *The International Arab Journal of Information Technology*, *18*(6), 789–796.

da Silva Mattos, E., & Shasha, D. (2024). Bankruptcy prediction with low-quality financial information. *Expert Systems with Applications*, *237*, Article 121418.

Syed Nor, S. H., Ismail, S., & Yap, B. W. (2019). Personal bankruptcy prediction using decision tree model. *Journal of Economics, Finance and Administrative Science*, *24*(47), 157–170.

Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, *29*(4), 309–317.

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, *2*(Nov), 45–66.

Tripathi, D., Edla, D. R., Cheruku, R., & Kuppili, V. (2019). A novel hybrid credit scoring model based on ensemble feature selection and multilayer ensemble classification. *Computational Intelligence*, *35*(2), 371–394.

Vapnik, V. N. (1982). *Estimation of dependences based on empirical data*. Springer-Verlag.

Wang, H., Zheng, B., Yoon, S. W., & Ko, H. S. (2018). A support vector machine-based ensemble algorithm for breast cancer diagnosis. *European Journal of Operational Research*, *267*(2), 687–699.

Zendehboudi, A., Baseer, M. A., & Saidur, R. (2018). Application of support vector machine models for forecasting solar and wind energy resources: A review. *Journal of Cleaner Production*, *199*, 272–285.

Zhang, G., Hu, M. Y., Patuwo, B. E., & Indro, D. C. (1999). Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. *European Journal of Operational Research*, *116*(1), 16–32.