Handelshøyskolen BI

GRA 19703 Master Thesis

Thesis Master of Science 100% - W

Predefinert informasjon

Startdato:	09-01-2023 09:00 CET	**Termin:**	202310
Sluttdato:	03-07-2023 12:00 CEST	**Vurderingsform:**	Norsk 6-trinns skala (A-F)
Eksamensform:	T		
Flowkode:	202310||11184||IN00||W||T		
Intern sensor:	(Anonymisert)		

Deltaker

Navn:	Thanh Khiem Tran og Ahmed Tariq

Informasjon fra deltaker

Tittel *:	Towards Improved Bankruptcy Prediction: Utilizing Variational Autoencoder Latent Representations in a Norwegian Context
Navn på veileder *:	Rogelio Andrade Mancisidor

Inneholder besvarelsen konfidensielt materiale?:	Nei	**Kan besvarelsen offentliggjøres?:**	Ja

Gruppe

Gruppenavn:	(Anonymisert)
Gruppenummer:	250
Andre medlemmer i gruppen:	

Master Thesis

Towards Improved Bankruptcy Prediction: Utilizing Variational Autoencoder Latent Representations in a Norwegian Context

Hand-in date:
16.01.2023

Campus:
BI Oslo

Examination code and name:
GRA 19702 Master Thesis

Supervisor:
Rogelio Andrade Mancisidor

Programme:
Master of Science in Business Analytics

Acknowledgements

The journey to the successful completion of this thesis, undertaken during the Spring semester of 2023 as an integral part of our Master of Science degree in Business Analytics at the BI Norwegian Business School, has been made possible through the relentless efforts, invaluable contributions, and unwavering support of several key individuals and entities.

At the outset, we would like to express our profound gratitude to our supervisor, Professor Rogelio A. Mancisidor. His expertise and keen insights have provided us with an academic compass, guiding us through the challenges we encountered. His unwavering support and constructive feedback were instrumental in elevating the quality of our work to the level it is today.

We also wish to extend our heartfelt appreciation to Espen Jütte, Director of Analytics and Data Science at CPM Analytics, whose insightful discussions and generous provision of access to Baltar, the company's server, have been integral to the computational backbone of our research.

Special thanks are due to the staff of the BI Library for their indispensable support in data acquisition. Their commitment to assist us in obtaining the necessary data significantly facilitated our research process.

Last but not least, our families have been our pillars of strength, providing unwavering support and constant encouragement throughout this academic journey. Their belief in our abilities and their ceaseless moral support have been an unending source of inspiration and motivation.

Abstract

This thesis conducts a rigorous examination of the potential for Variational Autoencoder (VAE)-derived latent embeddings to enhance the performance of classification algorithms when assessing business risk profiles from accounting data, specifically focusing on the Norwegian context. The study compares the performance of classifiers using VAE latent embeddings against those utilizing original or balanced training sets directly. Generally, it was found that, with the exception of Logistic Regression in certain experimental settings, the performance of classifiers using VAE latent embeddings was somewhat inferior. This outcome suggests that the dimensionality reduction process inherent to VAE may induce a degree of predictive power loss.

However, VAE latent embeddings were observed to bolster the performance of Logistic Regression by effectively capturing complex, high-dimensional relationships within a compressed, lower-dimensional space. This process reduced noise, identified non-linear relationships, and introduced a beneficial regularization effect, which may enhance the generalizability of the Logistic Regression model.

Furthermore, an increase in the dimensionality of the latent space up to a certain threshold improved the performance of classifiers, beyond which a decline was observed, indicating an optimal dimensionality z for these datasets. The application of under-sampling or over-sampling techniques to the training sets generally led to decreased classifier performance, particularly for Extreme Gradient Boosting and Multi-Layer Perceptron, with Logistic Regression as an exception in certain contexts.

Notably, for the Norwegian dataset, the Extreme Gradient Boosting classifier often demonstrated superior performance when utilizing raw training sets. These findings provide valuable insights into the capabilities and limitations of VAE in assessing business risk profiles and underscore the need for further research in this promising field.

Keywords: Variational Autoencoder, Bankruptcy Prediction, Machine Learning, Latent Embeddings, Norwegian Data Set.

Contents

Introduction

Motivation and Research Questions

The utilization of bankruptcy prediction models is multifarious and encompasses a vast array of clientele. The capability to evaluate the robustness and potential hazards of targeted entities is of paramount interest to banks, investors, and credit institutions (Baesens et al., 2003). Furthermore, public organizations are also invested in such models. Both the Central Bank of Norway and the Financial Supervisory Authority of Norway utilize bankruptcy models to gauge the financial well-being of corporations. The models need to satisfy the requirements for interpretability of the models. It is one of the reasons the models such as decision trees and logistic regression so widely used in bankruptcy prediction models.

As the ramifications of the 2008 financial crisis continue to be felt, the need for models to anticipate bankruptcy has become increasingly acute. This is consistent with the findings of (Agarwal & Taffler, 2008), who observed a surge in interest in the evaluation of credit market risk. The insolvency of a company is a matter of tremendous concern for its proprietors, government authorities, and other relevant parties within the national economy (McKee & Lensberg, 2002). The domino effect of bankruptcies that ensued in the aftermath of the global financial crisis can be compared to what happened during the COVID-19 pandemic (Reinhart, 2022).

The emergence of the Basel II further highlighted the importance of bankruptcy modelling. It requires the banks to maintain a capital reserve equal to at least 8% of their weighted assets. It also allowed banks to measure the company's risk of bankruptcy by using their internal models (de Andrés et al., 2012). This reserve ratio increased by 2.5% in Basel III, which is further expected to increase in Basel IV set to take effect from 2025 (Nordea, 2021). The better the models are at identifying bankruptcy, the faster the authorities can react to the threats ensued by these bankruptcies. It is therefore of the utmost importance for authorities to closely monitor the financial health of Norwegian companies and monitor the risks associated with company bankruptcies.

In the realm of predicting bankruptcy using financial statement data, Beaver's study (1966) was among the pioneers. This early research relied heavily on calculating financial ratios and comparing them against pre-determined cut-off thresholds. At the core of such studies, bankruptcy classification problems were

interpreted as binary classification problems (Barboza et al., 2017). Given the binary nature of bankruptcy outputs, certain models were deemed more suitable than others. Most of these bankruptcy prediction models were crafted utilizing statistical methods such as univariate statistical methods, multiple discriminant analysis (Altman, 1968), and logit (Ohlson, 1980) and probit (Zmijewski, 1984) analyses. Nevertheless, these methods grappled with certain limiting assumptions such as linearity, normality, and independence of predictor or input variables (Karels & Prakash, 1987; Balcaen & Ooghe, 2006; Yeh et al., 2014), reducing their effectiveness. Further, the use of a rigid function in these models posed challenges in capturing the complex and intricate relationships characteristic of financial systems, due to the inherent assumptions of these statistical methods. Over time, the development of machine learning and computational capabilities has enabled us to utilize more sophisticated methods for solving classification problems.

Variational autoencoder is a type of generative model that was introduced by Kingma and Welling (2013). VAEs leverage variational inference techniques to learn the parameters of the model and perform efficient optimization. Best known for its generative capability and the ability to learn a lower-dimensional latent space representation of the input data, numerous research has been done to examine its performance in learning meaning full representations and classification problem (Xu et al., 2017; Connor et al., 2021). Recently, Mancisidor et al. (2021) shows that VAE can capture the risk profile of customers. This motivates us to use this generative model in predicting bankruptcies. The innovative aspect of our thesis, when compared with extant literature, resides in the utilization of VAE model-derived latent embeddings as inputs for classification algorithms. The main objective of this exploration is to discern whether the VAE can effectively comprehend the risk profile of companies based on their respective accounting data. By leveraging three distinct datasets, deploying four separate classification methodologies, and assessing a variety of parameters, we meticulously examine if the VAE can effectively construct valuable representations from raw accounting data, thereby augmenting the classifier's performance. It is noteworthy to mention that, to the best of our understanding, there is a dearth of research that scrutinizes the efficacy of VAE latent embeddings when used as inputs for classification algorithms, particularly within the context of Norwegian companies.

This uncharted territory of research has led us to formulate the central question that this thesis seeks to answer:

Can Variational Autoencoder-derived latent embeddings enhance the performance of classification algorithms in assessing business risk profiles from accounting data, specifically in the Norwegian context?

Overview of The Sections

This thesis consists of seven sections. In the following section, we will delve into the literature surrounding bankruptcy prediction and highlight a recent study that explores the use of machine learning techniques for this purpose. It also provides an overview of the Variational Autoencoder (VAE) and its application in anomaly detection, especially bankruptcy prediction. Section three outlines the methodology employed in the analysis. Section four offers an in-depth examination of the data utilized in this analysis, including the derivation of the variables of interest and their transformation into the final data sets. Section five presents the results of the analysis. Section six delves into the limitations of our study and suggests potential avenues for further research. In the final section, we will summarize the main research questions, highlight the key findings of our analysis and suggest the implications of our results for future research.

Literature Review

Bankruptcy prediction refers to the practice of identifying companies that are likely to face financial difficulties and ultimately declare bankruptcy. This process typically involves the examination of various financial and non-financial indicators to evaluate a company's financial stability and determine its risk of defaulting on its debts (Altinn, 2021). Bankruptcy prediction models are usually based on statistical techniques and employ financial ratios, accounting data, and other relevant information to calculate the likelihood of a company facing financial distress. These models can be utilized by investors, creditors, and other stakeholders to make knowledgeable decisions about the companies they have dealings with, and by policymakers to monitor and regulate the overall health of the economy.

Table 1

Summaries of Papers in the Literature Review

Section	Author(s)	Year	Title
Earliest Works in Bankruptcy Prediction	Barboza et al.	2017	Machine Learning Models and Bankruptcy Prediction.
Earliest Adaptations	Fitzpatrick	1931	A Comparison of Ratios of Successful Industrial Enterprises with Those of Failed Companies
Earliest Adaptations	Horrigan	1968	A Short History of Financial Ratio Analysis
Earliest Adaptations	Smith & Winakor	1935	Changes in the Financial Structure of Unsuccessful Industrial Corporations
Earliest Adaptations	Chudson	1945	The Pattern of Corporate Financial Structure
Earliest Adaptations	Jackendoff	1962	A Study of Published Industry Financial and Operating Ratios
Earliest Adaptations	Treacy & Carey	2000	Credit Risk Rating Systems at Large US Banks
Beaver's Univariate Model	Beaver	1966	Financial Ratios as Predictors of Failure
Altman Z-Score	Altman	1968	Financial Ratios, Discriminant Analysis, and the Prediction of Corporate Bankruptcy
Ohlson O-Score	Ohlson	1980	Financial Ratios and the Probabilistic Prediction of Bankruptcy
Ohlson O-Score	Upneja & Dalbor	2001	An Examination of Capital Structure in the Restaurant Industry
Ohlson O-Score	Hillegeist et al.	2004	Assessing the Probability of Bankruptcy
Ohlson O-Score	Muzır & Çağlar	2009	The Accuracy of Financial Distress Prediction Models in Turkey: A Comparative Investigation with Simple Model Proposals
Ohlson O-Score	Diakomihalis	2012	The Accuracy of Altman's Models in Predicting Hotel Bankruptcy
Ohlson O-Score	Begley, Ming & Watts	1996	Bankruptcy classification errors in the 1980s: An empirical analysis of Altman's and Ohlson's models

Table 1

Summaries of Papers in the Literature Review

Section	Author(s)	Year	Title
Ensemble Learning	Opitz & Maclin	1999	Popular Ensemble Methods: An Empirical Study
Ensemble Learning	Polikar	2006	Ensemble Based Systems in Decision Making
Ensemble Learning	Dasarathy & Sheela	1979	A Composite Classifier System Design: Concepts and Methodology
Ensemble Learning	Hansen & Salamon	1990	Neural Network Ensembles
Ensemble Learning	Schapire	1990	A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting
Ensemble Learning	Freund & Schapire	1997	A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting
Ensemble Learning	Wolpert	1992	Stacked generalization
Ensemble Learning	Breiman	1996	Bagging Predictors
Ensemble Learning	Ho	1998	The Random Subspace Method for Constructing Decision Forests
Ensemble Learning	Breiman	2001	Random Forests
Ensemble Learning	Tanaka et al.	2016	Random Forests-Based Early Warning System for Bank Failures
Ensemble Learning	Joshi et al.	2018	A Bankruptcy Prediction Model Using Random Forest
Ensemble Learning	Prusak	2018	Review of Research into Enterprise Bankruptcy Prediction in Selected Central and Eastern European Countries
Ensemble Learning	Friedman	2001	Greedy Function Approximation: A Gradient Boosting Machine
Ensemble Learning	Wyrobek & Kluza	2018	Efficiency of Gradient Boosting Decision Trees Technique in Polish Companies' Bankruptcy Prediction
Ensemble Learning	Quynh & Phuong	2020	Improving the Bankruptcy Prediction by Combining Some Classification Models
Ensemble Learning	Chen & Guestrin	2016	XGBoost
Ensemble Learning	Breiman et al.	2017	Classification and Regression Trees
Ensemble Learning	Zięba et al.	2016	Ensemble Boosted Trees with Synthetic Features Generation in Application to Bankruptcy Prediction
Ensemble Learning	Carmona et al.	2019	Predicting Failure in the U.S. Banking Sector: An Extreme Gradient Boosting Approach
	Pawełek	2019	Extreme Gradient Boosting Method in the Prediction of Company Bankruptcy
Neural Networks	McCulloch & Walter	1943	A Logical Calculus of the Ideas Immanent in Nervous Activity
Neural Networks	Rosenblatt	1958	The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain
Neural networks	Minsky & Papert	1969	Perceptron's – An Introduction to Computational Geometry
Neural Networks	Hopfield	1982	Neural Networks and Physical Systems with Emergent Collective Computational Abilities

Table 1

Summaries of Papers in the Literature Review

Section	Author(s)	Year	Title
Neural Networks	Schmidhuber & Hochreiter	1997	Bankruptcy Classification Errors in the 1980s: An Empirical Analysis of Altman's and Ohlson's Models
Neural Networks	Fukushima	1980	Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position
Neural Networks	Lecun et al.	1998	Gradient-based Learning Applied to Document Recognition
Neural Networks	Odom & Sharda	1990	A Neural Network Model for Bankruptcy Prediction
Neural Networks	Bell et al.	1990	Neural Nets Versus Logistic Regression: A Comparison of Each Model's Ability to predict Commercial Bank Failures
Neural Networks	Wilson & Sharda	1994	Bankruptcy Prediction Using Neural Networks
Neural Networks	Atiya	2001	Bankruptcy Prediction for Credit Risk Using Neural Networks: A Survey and New Results
Neural Networks	Kim & Kang	2010	Ensemble with Neural Networks for Bankruptcy Prediction
Neural Networks	Mai et al.	2019	Deep Learning Models for Bankruptcy Prediction using Textual Disclosures
Neural Networks	Salchenberger et al.	1992	Neural Networks: A New Tool for Predicting Thrift Failures
Neural Networks	Barniv et al.	2002	Predicting Bankruptcy Resolution
Neural Networks	Kim & Ahn	2012	A Corporate Credit Rating Model Using multi-Class Support Vector Machines with an Ordinal Pairwise Partitioning Approach
Neural Networks	Chung & Tam	1993	A Comparative Analysis of Inductive-Learning Algorithms
Neural Networks	Coats & Fant	1993	Recognizing Financial Distress Patterns Using a Neural Network Tool
Neural Networks	Lee et al.	2005	A Comparison of Supervised and Unsupervised Neural Networks in Predicting Bankruptcy of Korean Firms
Neural Networks	Chen	2011	Predicting Corporate Financial Distress Based on Integration of Decision Tree Classification and Logistic Regression
Neural Networks	Öcal et al.	2015	Predicting Financial Failure Using Decision Tree Algorithms: An Empirical Test on the Manufacturing Industry at Borsa Istanbul
Neural Networks	du Jardin	2010	Predicting Bankruptcy Using Neural Networks and Other Classification Methods: The Influence of Variable Selection Techniques on Model Accuracy
Neural Networks	Zhao et al.	2015	Investigation and Improvement of Multi-layer Perceptron Neural Networks for Credit Scoring
Neural Networks	Tsai & Wu	2008	Using Neural Network Ensembles for Bankruptcy Prediction and Credit Scoring

Table 1

Summaries of Papers in the Literature Review

Section	Author(s)	Year	Title
Neural Networks	Hosaka	2019	Bankruptcy Prediction Using Imaged Financial Ratios and Convolutional Neural Networks
Autoencoder	LeCun & Fogelman-Soulié	1987	Connectionist Learning Models
Autoencoder	Hinton & Zemel	1993	Autoencoders, Minimum Description Length and Helmholtz Free Energy
	Hinton & Salakhutdinov	2006	Reducing the Dimensionality of Data with Neural Networks
Autoencoder	Xu et al.	2019	Adversarially Approximated Autoencoder for Image Generation and Manipulation
Autoencoder	Liou et al.	2014	Autoencoder for Words
Autoencoder	Zhou & Paffenroth	2017	Anomaly Detection with Robust Deep Autoencoders
Autoencoder	Zhuang et al.	2015	Supervised Representation Learning: Transfer Learning with Deep Autoencoders
Autoencoder	Vincent et al.	2008	Extracting and Composing Robust Features with Denoising Autoencoders
Autoencoder	Makhzani & Frey	2013	k-Sparse Autoencoders
Autoencoder	Makhzani et al.	2015	Adversarial Autoencoders
Autoencoder	Sakurada & Yairi	2014	Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction
Autoencoder	Pumsirirat & Yan	2018	Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine
Autoencoder	Soui et al.	2019	Bankruptcy Prediction Using Stacked Auto-Encoders
Variational Autoencoder	Kingma & Welling	2013	Auto-Encoding Variational Bayes
Variational Autoencoder	An & Cho	2015	Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability
Variational Autoencoder	Xu et al.	2018	Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications
Variational Autoencoder	Cozzatti et al.	2022	Variational Autoencoders for Anomaly Detection in Respiratory Sounds
Variational Autoencoder	Mancisidor et al.	2018	Segment-Based Credit Scoring Using Latent Clusters in the Variational
Variational Autoencoder	Mancisidor et al.	2021	Learning latent representations of bank customers with the Variational Autoencoder
Norway: Local Adaptations	Smogeli	1987	Dokumentasjonsnotat SEBRA
Norway: Local Adaptations	Bernhardsen	2001	Working Paper: A Model of Bankruptcy Prediction. Norges Bank
Norway: Local Adaptations	Bernhardsen & Larsen	2007	Modellering av kredittrisiko i foretakssektoren - Videreutvikling av SEBRA-modellen

Table 1

Summaries of Papers in the Literature Review

Section	Author(s)	Year	Title
Conclusion	Ravi Kumar & Ravi	2007	Bankruptcy Prediction in Banks and Firms via Statistical and Intelligent Techniques – A Review
Conclusion	Kirkos	2012	Assessing Methodologies for Intelligent Bankruptcy Prediction

Earliest Works in Bankruptcy Prediction

Bankruptcy prediction models, as stated by Barboza et al. (2017), are believed to have a significant impact on various stakeholders in the financial industry. Banks, investors, managers, rating agencies, and distressed businesses all stand to benefit from these models, as they can use them to make more informed decisions and potentially mitigate losses. The need for accurate bankruptcy prediction models is further justified by the high costs associated with inaccurate diagnoses of bankruptcy. An inaccurate diagnosis can lead to a loss of credibility and trust among investors and other stakeholders, whilst failing to identify a company that is actually at risk can lead to significant financial losses.

Earliest Adaptations

The underlying reasons for business failures have been the subject of extensive discourse and investigation for centuries with ratio analysis standing at the crux of the issue (Horrigan, 1968). Prior to the development of sophisticated models, financial institutions' evaluations of credit risk on corporate loans were largely based on subjective judgments that relied on a few key variables, such as leverage, collateral, and earnings. The traditional approach for risk assessment is using credit scoring models to evaluate the creditworthiness of a company and predict its likelihood of defaulting on its loans (Treacy & Carey, 2000).

During the interval spanning 1930 to 1965, scholarly literature on the subject was relatively scant. Early analyses of ratios were undertaken during this period; however, they failed to yield any substantial discoveries with regard to predicting bankruptcy (Fitzpatrick, 1931; Smith & Winakor, 1935; Chudson, 1945; Jackendoff, 1962). These univariate studies focused on financial ratios, comparing the financial performance of successful firms to that of failed firms. They discovered that the working capital to total assets ratio was a salient indicator of financial performance.

Beaver's Univariate Model

Next, Beaver's univariate model (1966) used financial ratios as a means of forecasting corporate failure. The research conducted by Beaver discovered that a combination of four financial ratios were most efficacious in predicting bankruptcy: the current ratio, acid-test ratio, working capital to total assets ratio, and net profit to net worth ratio. The study also found that these financial ratios were more effective in predicting bankruptcy for smaller firms rather than larger ones. The study was also one of the first to use univariate statistical analysis to examine the relationship between each financial ratio and the outcome (bankruptcy or non-bankruptcy) which is a simple but effective approach. Although Beaver's findings were a beacon of light in the field of bankruptcy prediction, illuminating the importance of financial ratios, there are many limitations with Beaver's approach. Firstly, the study only used a small number of financial ratios, which may not be sufficient to fully capture the financial condition of a company. Next, the study only used univariate statistical analysis, which only considers the relationship between a single variable and bankruptcy. Multivariate analysis, which considers the relationship between multiple financial ratios and the outcome, would likely provide more accurate predictions.

Altman Z-Score

Altman's (1968) study introduced the first multivariate model for bankruptcy prediction. He examined the financial ratios of publicly held manufacturing companies that have filed for bankruptcy and compared them to a sample of financially healthy companies. Through the use of discriminant analysis, Altman was able to develop a five-factor model to predict the bankruptcy of those firms. He proposed Z-score, also known as the Altman Z-score, which is a variation of the traditional statistic based on five financial ratios. The five ratios include working capital and total assets (X_1), retained earnings and total assets (X_2), earnings before interest and taxes and total assets (X_3), market value of equity and book value of total liabilities (X_4), and sales and total assets (X_5). The original Z-score formula is as follow:

$$Z = 1.2X_1 + 1.4X_2 + 3.3X_3 + 0.6X_4 + 1.0X_5$$

The aforementioned function generates a Z-score that serves as an indicator of the projected financial stability of an organization. It follows that an elevated Z-score implies a diminished likelihood of insolvency, while a reduced Z-score

signals an amplified probability of insolvency. Additionally, Altman suggests an upper limit for the Z-score set at 2.67 and a lower limit set at 1.81, with the goal of curtailing the incidence of misclassifications. A datum exhibiting a Z-score surpassing the higher limit is categorized as financially solvent, whereas a datum with a Z-score falling beneath the lower limit is designated as insolvent. A Z-score residing within these specified limits implies ambiguity in respect to its financial classification.

The model demonstrates strong predictive capabilities for predicting bankruptcy within one year, however, its accuracy diminishes as the forecast horizon increases. The model yields 79% accuracy for one-year horizon out-of-sample data. Overall, the paper has a significant impact on the field of bankruptcy prediction. Since its publication, a wide range of new models have been developed, such as logistic regression, neural networks, and decision tree models, and many of these have been applied in various fields and industries, such as banking, healthcare, and retail.

Ohlson O-Score

Ohlson (1980) published a paper on bankruptcy prediction which proposed a new model known as the Ohlson's O-Score model. Unlike traditional models which simply classified companies as bankrupt or non-bankrupt, this model estimates the probability of bankruptcy for a company using a combination of financial ratios and a logistic regression model. The formula of the model is as follows:

$$O-Score = -1.32 - 0.407SIZE + 6.03TLTA - 1.43WCTA + 0.0757CLCA - 1.72OENEG - 2.37NITA - 1.83FUTL + 0.285INTWO - 0.521CHIN$$

where:

- SIZE refers to the natural log of total assets divided by the GNP price-level index.
- TLTA refers to total liabilities divided by total assets.
- WCTA refers to working capital divided by total assets.
- CLCA refers to current liabilities divided by current assets.
- OENEG is a binary variable that equals 1 if total liabilities exceed total assets, and 0 otherwise.
- NITA refers to net income divided by total assets.
- FUTL refers to funds provided by operations divided by total liabilities.
- CHIN refers to the natural log of the absolute change in net income.

The model also takes into account the uncertainty of the predictions and the correlation between the predictors by using principal component analysis. The model yields 85% accuracy for one-year horizon out-of-sample data. The Ohlson's O-Score model has been considered an important contribution to the field of bankruptcy prediction as it offers a more realistic and robust approach by taking into account uncertainty and correlation.

It has been observed that a plethora of research studies have sought to replicate earlier investigations utilizing discriminant analysis and logistic regression owing to their relative ease of implementation (Upneja & Dalbor, 2001; Hillegeist et al., 2004; Muzır & Çağlar, 2009; Diakomihalis, 2012). Despite this, Begley, Ming and Watts (1996) posited that the models established by Altman (1968) and Ohlson (1980) had become obsolete. As such, they advocated for the development of more advanced modelling techniques to assess default risk.

Machine Learning in Bankruptcy Prediction

Ensemble Learning

Ensemble learning is a machine learning technique that combines multiple models to create a stronger and more accurate predictive model (Opitz & Maclin, 1999; Polikar, 2006). By aggregating the predictions of diverse models, ensemble learning harnesses the collective wisdom to improve overall performance, increase stability, and reduce overfitting. It leverages different algorithms, data subsets, or training methods to enhance the model's capabilities, making it a valuable tool for various tasks such as classification, regression, and anomaly detection.

The seminal work that arguably introduced the concept of ensemble systems can be attributed to Dasarathy and Sheela (1979). They pioneered the innovative idea of using ensemble systems in a strategic divide-and-conquer manner by segmenting the feature space utilizing two or more classifiers. More than a decade later, the insightful work by Hansen and Salamon (1990) elucidated the characteristic of variance reduction inherent to ensemble systems. Furthermore, they convincingly demonstrated that the generalization performance of a neural network could be significantly enhanced through the use of an ensemble composed of similarly structured neural networks.

The three primary ensemble learning methods encompass bagging, stacking, and boosting. Schapire (1990) demonstrated that a robust classifier could be created from the probably approximately correct (PAC) concept through the

integration of weaker classifiers, a process he dubbed as "boosting". It is significant to note that boosting laid the groundwork for the AdaBoost family of algorithms (Freund & Schapire, 1997). These algorithms not only gained significant traction but also arguably became one of the most extensively used machine learning algorithms in recent history. Wolpert (1992) conceptualized stacking, a methodology otherwise denoted as stacked generalization. The technique demands the training of a model to synthesize the predictions emanating from an array of disparate learning algorithms. Initially, the ensemble of algorithms is trained using the given dataset. Following this, a combiner algorithm, also classified as the final estimator, is tutored to generate the conclusive prediction. This is accomplished either by employing all the predictions from the combiner algorithms as supplementary inputs or by harnessing cross-validated predictions from the base estimators, an approach which serves to forestall overfitting. The bagging algorithm was proposed by Breiman (1996). The methodology stipulates the creation of bootstrapped samples and the subsequent application of a regression or classification algorithm to each individual sample. In the sphere of classification tasks, the class that either secures the maximum votes or achieves the highest average class probability is selected as the resultant output. In this operation, the concept of aggregation assumes a pivotal role, specifically during the process of assimilating predictions from a multitude of learners.

In his seminal work, Ho (1998) proposed the innovative concept of the random subspace method. This technique is predicated on the establishment of diverse models, each carefully tailored to a randomly selected subspace of the input features. Expanding on this groundwork, Breiman (2001) put forth the idea of the Random Forest (RF). This method, both versatile and robust, is employed in numerous machine learning applications, particularly classification and regression tasks. Random forest constitutes a form of ensemble learning that deftly amalgamates principles of bagging and random feature subsets. The technique has found considerable application in a plethora of research, notably in bankruptcy prediction studies (Tanaka et al., 2016; Joshi et al., 2018; Prusak, 2018).

Subsequently, Friedman (2001) devised gradient boosting, a potent machine learning methodology applicable to both regression and classification problems. This approach generates a prediction model in the guise of an ensemble of weak prediction models, generally manifesting as decision trees. Gradient boosting stands as one of the most effective and potent machine learning methodologies,

constructing trees in a sequential order, where each succeeding tree attempts to rectify the errors of its predecessor. This method uses aggressive pre-pruning techniques to keep trees shallow, typically limiting their depth to between one and five layers. This approach not only economizes memory but also allows each tree to generate optimal projections for a subset of the data. Consequently, the augmentation of additional trees enhances the model's performance. Analogous to random forest, gradient boosting has also seen extensive utilization in bankruptcy research (Wyrobek & Kluza, 2018; Quynh & Phuong, 2020).

Following these advancements, Chen and Guestrin (2016) unveiled Extreme Gradient Boosting (XGB), a methodology that quickly ascended to prominence due to its exceptional performance and scalability. XGB builds on the foundational principles of gradient boosting by sequentially integrating weak learners into the ensemble, whilst simultaneously minimizing a loss function through gradient descent. Moreover, XGB utilizes binary classification and regression trees as its fundamental weak learners (Breiman et al., 2017). To avoid overfitting, XGB employs regularization techniques of L1 and L2 (Lasso Regression and Ridge Regression, respectively). The application of this model in bankruptcy prediction studies is documented widely in academic literature (Zięba et al., 2016; Carmona et al., 2019; Pawełek, 2019).

Neural Networks

Neural networks, also known as Artificial Neural Networks (ANNs), are a class of machine learning models that are based on the structure and function of the human brain. The concept of neural networks was first introduced by researchers such as McCulloch and Walter (1943), who sought to create mathematical models of the human brain's ability to process and analyze information. Since its inception, numerous research and models have been built upon it (Rosenblatt, 1958; Minsky & Papert, 1969). Hopfield (1982) introduced the first Recurrent Neural Networks (RNNs) in his journal paper. His idea was then expanded upon by Schmidhuber and Hochreiter (1997) with their renowned long short-term memory model. The model was designed to overcome the limitations of traditional RNNs in modeling long-term dependencies in sequential data. Traditional RNNs have difficulty in learning long-term dependencies because the gradients that are used to update the network's weights during training can vanish or explode as they are propagated through time. Fukushima (1980) introduce Neocognitron, an improved neural networks model,

which can be trained to recognize patterns based on learning. As a further development of this concept, Lecun et al. (1998) proposed a convolution network model, or LeNet-5 in their paper, which capable of recognizing handwritten digits and was trained on the MNIST data set. The model's architecture consists of two sets of convolutional and pooling layers, followed by two fully connected layers. The architecture is designed to extract features from the input image, and then use these features to classify the image into one of several possible classes.

The use of neural networks for bankruptcy predictions can trace back it roots to the studies (Odom & Sharda, 1990; Bell et al., 1990). As the field advanced, numerous studies have explored various approaches, techniques, and data sources, providing valuable insights into the potential of neural networks in enhancing bankruptcy prediction models (Wilson & Sharda, 1994; Atiya, 2001; Kim & Kang, 2010; Mai et al., 2019). However, during the nascent years there were several concerns regarding the NNs. Some of the concerns included overfitting of the model to the data, interpretability of the models, finding optimal architecture of the network, and the high computational cost (Salchenberger et al., 1992; Barniv et al., 2002; Kim & Ahn, 2012).

The performance of neural networks has been compared with traditional statistical methods is various studies (Chung & Tam, 1993; Coats & Fant, 1993; Lee et al., 2005). Several studies have shown better results for the AI models compared to the statistical methods mentioned above (Chen, 2011; Öcal et al., 2015). du Jardin (2010) uses neural network for predicting bankruptcy on a French bankruptcy dataset. The results of the study showed that the neural networks combined with a robust variable selection method gives better results. (Zhao et al., 2015) used multi-layer perceptron neural network to build a high accuracy automated credit system with an accuracy of 87 percent. Tsai and Wu (2008) studied the results of single and ensemble neural network classifier. The study showed that the ensemble neural network classifier was not able to produce better results than the best model of single neural network classifier for problems relating to binary classification. The success of neural network in the domains of bankruptcy prediction and credit scoring has encouraged novel approaches for the creating better models for the binary classification problems. In an interesting study (Hosaka, 2019) used the financial statement of Japanese listed companies and transformed the financial ratio data into grayscale image to make it better suited as an input for the convolutional neural network.

Autoencoder

An autoencoder is a type of neural network that was developed in the 1980s and has been widely utilized for unsupervised feature learning and dimensionality reduction (LeCun & Fogelman-Soulié, 1987; Hinton & Zemel, 1993; Hinton & Salakhutdinov, 2006). The fundamental concept behind autoencoder is to learn a compressed and low-dimensional representation of the data, known as the bottleneck or latent representation, by training a neural network to reconstruct the input data. The inception of autoencoder was motivated by the aspiration to learn useful representations of the data that can be applied to various tasks such as compression, denoising, and generation. The impact of autoencoder has been substantial in the field of machine learning and deep learning. Autoencoders have been extensively utilized for unsupervised feature learning and dimensionality reduction and have been employed in various applications such as image generation (Xu et al., 2019), text generation (Liou et al., 2014), anomaly detection (Zhou & Paffenroth, 2017), and representation learning (Zhuang et al., 2015). There are numerous variations of autoencoder to enhance the original version including denoising autoencoder, sparse autoencoder, adversarial autoencoder, and convolutional autoencoder (Vincent et al., 2008; Makhzani & Frey, 2013; Makhzani et al., 2015). In the field of anomaly detection, there has been a profuse amount of research on the topic, with promising results in anomaly image recognition and credit fraud (Sakurada & Yairi, 2014; Zhou & Paffenroth, 2017; Pumsirirat & Yan, 2018). However, there has been limited research on the application of deep learning algorithms in the prediction of bankruptcy (Soui et al., 2019). Despite its efficacy in the detection of anomalies, a primary shortcoming of the traditional autoencoder model is its inability to discern variations that deviate from its training data, owing to the discontinuity of its learned latent space.

Variational Autoencoder

One of the key shortcomings of traditional autoencoders pertains to the characteristics of their latent space, the lower-dimensional space where the encoded representations of the input data are situated. Specifically, traditional autoencoders generate a latent space that is both discrete and unstructured. This means that the placement and organization of data points within this space do not follow any discernible or cohesive pattern, which can subsequently affect the autoencoder's ability to effectively learn and generate new data. Moreover, the absence of a robust

probabilistic framework in traditional autoencoders hampers their capacity to handle uncertainty, variability, and complexity inherent in data. As such, whilst traditional autoencoders are undoubtedly effective in tasks related to feature extraction and dimensionality reduction, their applicability may be restricted when it comes to more complex unsupervised learning scenarios, especially those demanding a well-defined, probabilistic treatment of data.

Kingma & Welling (2013) proposed the Variational Autoencoder (VAE) as a variation of the original model. A key attribute of the VAE model is its continuous latent space, enabling seamless sampling and interpolation. This feature is extremely important in identifying patterns that deviate from the norm, which is a key aspect of anomaly detection. VAE can model uncertainty in the data by learning a probabilistic latent representation. Additionally, they can handle missing data by inferring the missing data from the learned probabilistic latent representation. Besides that, due to its generative capability, they can generate new data from the learned distribution. Similar to autoencoder, there are several studies using VAE for anomaly detection (An & Cho, 2015; Xu et al., 2018; Cozzatti et al., 2022), but to the best of our knowledge, there is no research for bankruptcy prediction. Recent research papers by Mancisidor et al. (2018; 2021) have shown the potential of using VAE for bankruptcy prediction. Using real data sets from three different regions, including Norway, Finnish, and Kaggle, the authors show that with the use of Weight of Evidence, it is possible to steer the configurations of the latent space, allowing the clusters of the data naturally reveal themselves.

Norway: Local Adaptations

In 1987, the Norwegian Central Bank established a comprehensive database of financial data for Norwegian firms known as SEBRA (Smogeli, 1987). Building upon this resource, they developed an accounting-based model for predicting corporate bankruptcy, known as the SEBRA model. This model takes a holistic approach to evaluating the credit and default risk of Norwegian banks and financial institutions, utilizing 12 key accounting variables from the SEBRA data set. Widely adopted in the Norwegian market as a means of assessing the stability and risk of companies and financial institutions, the SEBRA model was the first to employ logistic regression in the Norwegian market and has since become a benchmark for credit risk evaluation in the financial industry. In response to evolving financial

conditions, Norges Bank updated and refined the SEBRA model in 2007 to improve its accuracy and effectiveness (Bernhardsen 2001; Bernhardsen & Larsen, 2007).

Conclusion

This section has succinctly reviewed the literature pertaining to bankruptcy prediction. The field has undergone significant advancement since the advent of Beaver's univariate model. With the rapid advancement of machine learning, particularly deep learning, new models have emerged for detecting bankruptcy (Ravi Kumar & Ravi, 2007; Kirkos, 2012). Numerous research has demonstrated the ability of deep generative models to identify anomalies; however, the application of deep learning model for bankruptcy prediction is still limited. Drawing inspiration from the work of Mancisidor (2018, 2021), this thesis aims to contribute to the literature by applying Variational Autoencoder models to the prediction of bankruptcy in Norway.

Methodology

Variational Autoencoder

Variational Inference

Variational Inference is a technique used to approximate the posterior distribution of latent variables given observed data. In many cases, the true posterior distribution $p(z|x)$ is intractable to compute directly. Variational Inference offers a way to approximate this posterior by posing the problem as an optimization task. Variational Inference introduces a family of simpler distributions, called the variational family which are parameterized by variational parameters. These distributions are designed to be tractable, making the inference problem more manageable. The goal of variational inference is to find the best approximation to the true posterior distribution within the chosen variational family (Hinton & Van Camp, 1993). This is done by minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior distribution. The KL divergence measures the dissimilarity between two distributions and is minimized when they are similar. Note that KL divergence is always non-negative.

Mathematically, let $q(z)$ be the variational distribution and be the true posterior distribution given observed data x. The variational inference problem can be formulated as follows:

$$q^*(z) = \text{argmin}_{q(z)} \text{KL}(q(z) || p(z|x))$$

Unfortunately, directly optimizing this KL divergence is still intractable. However, we can reformulate it as the maximization of *the evidence lower bound* (ELBO). The ELBO is a lower bound on the log-likelihood of the data and is given by:

$$\text{ELBO}(x) = \mathbb{E}_{q(z)}[\log p(x,z) - \log q(z)]$$

Maximizing the ELBO is equivalent to minimizing the KL divergence, and it provides an effective way to perform variational inference. By optimizing the variational parameters of $q(z)$, we can find an approximation that closely matches the true posterior distribution $p(z|x)$. A derivation of ELBO is given in the upcoming section.

Figure 2

Graphical Presentation of VAE Architecture

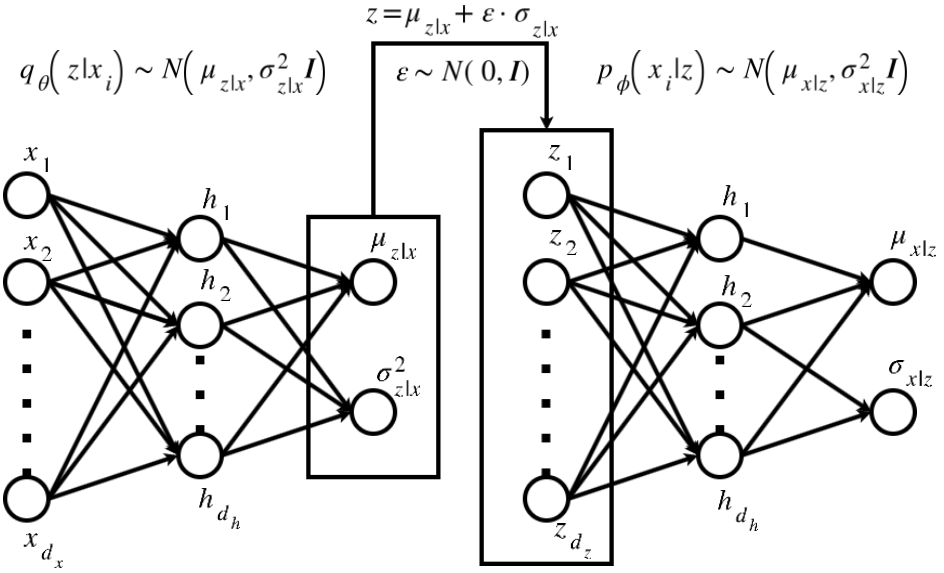


Figure 2 demonstrates the components of a VAE. The encoder of a VAE gives an approximate posterior distribution $q(z|x)$, which in our derivation is parametrised by the weights denoted as θ. Thus, the encoder can be denoted more precisely as $q_\theta(z|x)$. The decoder portion of the VAE yields a likelihood distribution $p(z|x)$, it takes a sample from the latent distribution and maps it back to the original input space. The decoder is also implemented using a neural network with weights collectively which in our derivation is denoted as ϕ. So, we denote the decoder as $p_\phi(x|z)$.

The external generation of ε, which follows a standard normal distribution denoted as $N(0, I)$, is a result of an approach commonly referred to as the "reparameterization trick". This is integral in Variational Autoencoder where the objective is to compute gradients through stochastic operations to optimize the parameters of the encoder using optimization techniques such as gradient descent and backpropagation. Notably, the conventional means do not permit this, as the sampling operation is inherently non-deterministic and consequently non-differentiable. The reparameterization trick offers an innovative solution to this predicament. Instead of directly sampling from the distribution $q_\theta(z|x)$, we generate ε from a standard normal distribution, and subsequently derive z using a deterministic function that incorporates ε as a component ($z = \mu_{z|x} + \varepsilon \cdot \sigma_{z|x}$). This approach facilitates the inclusion of the sampling operation in the computational

graph, which renders backpropagation through this operation feasible because it's fully differentiable with respect to μ and σ. Consequently, this promotes the successful application of gradient descent and backpropagation optimization methods in the training process of the VAE.

The goal of the decoder is to reconstruct the original input data from the sampled latent variables. It takes a sample from the latent space and generates a distribution over the input space. This distribution represents the likelihood of generating the original data point given the latent variables. During training, the VAE model aims to learn the parameters θ and ϕ in such a way that the generated data closely matches the original data. This is achieved by maximizing a lower bound on the log-likelihood of the data, which is the ELBO. The ELBO consists of two terms: the reconstruction loss, which measures how well the decoder reconstructs the input data, and the KL divergence between the approximate posterior and a prior distribution over the latent variables. The latter term encourages the learned latent distribution to be close to the prior distribution.

Here we assume that x is the company data/evidence like ROA, Profit margin, P/E ratio etc. and, z is the latent variable. Furthermore, $p(x)$ is the evidence probability, and the $p(z)$ would be the prior probability. The posterior probability would then be given as, and $p(x|z)$ would be the likelihood probability. Note that the KL divergence is always non-negative since:

$$D_{KL}(q(x)\,||p(x)) = -\int q(x)\log\left(\frac{p(x)}{q(x)}\right)dx \geq 0$$

The KL divergence between the approximate and the real posterior distributions is given by:

$$D_{KL}(q_\theta(z|x_i)\,||p(z|x_i)) = -\int q_\theta(z|x_i)\log\left(\frac{p(z|x_i)}{q_\theta(z|x_i)}\right)dz \geq 0 \quad (1)$$

The Bayes' Theorem, which is a systematic method for revising beliefs in light of new data or information, is given by:

$$p(z|x) = \frac{p(x|z)\,p(z)}{p(x)}$$

- $p(z|x)$: This is called the posterior probability. It represents the updated probability of the hypothesis z being true given the observed evidence x. Essentially, after we observe evidence x, we use Bayes' theorem to update our belief about the likelihood of hypothesis z.

- $p(x|z)$: This is called the likelihood. It is the probability of observing the evidence x given that the hypothesis z is true. This quantifies the compatibility of the evidence with the given hypothesis.
- $p(z)$: This is called the prior probability. It is the probability of the hypothesis z before observing the evidence. It represents our initial belief about the likelihood of the hypothesis before we have observed any evidence.
- $p(x)$: This is the marginal probability of the evidence x, also known as the evidence term. It represents the total probability of observing the evidence, across all possible hypotheses.

Applying Bayes' Theorem to the equation (1) we get:

$$D_{KL}(q_\theta(z|x_i)\,||\,p(z|x_i)) = -\int q_\theta(z|x_i)\log\left(\frac{p_\phi(x_i|z)\,p(z)}{q_\theta(z|x_i)\,p(x_i)}\right)dz \geq 0 \tag{2}$$

This can be broken down using laws of logarithms, giving:

$$\log p(x_i) \geq -D_{KL}\left(q_\theta(z|x_i)\,||\,p(z)\right) + E_{q_\theta(z|x_i)}\left[\log\left(p_\phi(x_i|z)\right)\right] \tag{3}$$

Distributing the integrand will gives us:

$$-\int q_\theta(z|x_i)\log\frac{p_\phi(x_i|z)\,p(z)}{q_\theta(z|x_i)}dz + \log p(x_i) \geq 0 \tag{4}$$

In the above, we note that $\log p(x_i)$ is a constant and can therefore be pulled out of the second integral above, and since $q_\theta(z|x_i)$ is a probability distribution it integrates to 1, further simplifying as:

$$-\int q_\theta(z|x_i)\log\frac{p_\phi(x_i|z)\,p(z)}{q_\theta(z|x_i)}dz + \log p(x_i) \geq 0 \tag{5}$$

Carrying integral over to the other side of the inequality and applying the rule of algorithm we get:

$$\log p(x_i) \geq \int q_\theta(z|x_i)\log\frac{p_\phi(x_i|z)\,p(z)}{q_\theta(z|x_i)}dz \tag{6}$$

$$\log p(x_i) \geq \int q_\theta(z|x_i)\left[\log p_\phi(x_i|z) + \log p(z) - \log q_\theta(z|x_i)\right]dz \tag{7}$$

Writing the right-hand side as an expectation we get:

$$\log p(x_i) \geq E_{\sim q_\theta(z|x_i)}\left[\log p_\phi(x_i|z) + \log p(z) - \log q_\theta(z|x_i)\right] \tag{8}$$

$$\log p(x_i) \geq E_{\sim q_\theta(z|x_i)}\left[\log p(x_i, z) - \log q_\theta(z|x_i)\right] \tag{9}$$

From equation (6) it follows that:

$$\log p(x_i) \geq -D_{KL}(q_\theta(z|x_i)\,||\,p(z)) + E_{\sim q_\theta(z|x_i)}\left[\log p_\phi(x_i|z)\right] \tag{10}$$

The right-hand side of the equation is *the evidence lower bound* also called variational lower bound. The term is intuitive as it bounds the likelihood of the data which is the term that we are trying to maximise. This term works as a proxy for maximising the log probability of the data which is the core idea behind variational inference. The Kullback-Leibler term $-D_{KL}(q_\theta(z|x_i)||p(z))$ in ELBO is a regulariser as it acts as a constraint on the form of approximate posterior.

We can obtain a closed form for the loss function if we choose a gaussian representation for the latent prior $p(z)$ and the approximate posterior, $q_\theta(z|x_i)$. In addition to yielding a closed form loss function, the gaussian model enforces a form of regularization in which the approximate posterior has variation or spread. We choose:

$$p(z) \to \frac{1}{\sqrt{2\pi\sigma_p^2}}\exp\left(\frac{-(x-\mu_p)^2}{2\sigma_p^2}\right)$$

and

$$q_\theta(z|x_i) \to \frac{1}{\sqrt{2\pi\sigma_p^2}}\exp\left(\frac{-(x-\mu_p)^2}{2\sigma_p^2}\right)$$

then KL in ELBO becomes:

$$-D_{KL}\left[q_\theta(z|x_i)||p(z)\right] = \int \frac{1}{\sqrt{2\pi\sigma_p^2}}\exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right)\log\left(\frac{\frac{1}{\sqrt{2\pi\sigma_p^2}}\exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_p^2}}\exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right)}\right)dz$$

With simplifying the above term further by evaluating in log, and further simplifying we get the expression show in expectation (E_q):

$$-D_{KL}\left[q_\theta(z|x_i)||p(z)\right] = \log\frac{\sigma_q}{\sigma_p} - \frac{1}{2\sigma_p^2}E_q\left[(x-\mu_p)^2\right] + \frac{1}{2\sigma_q^2}E_q\left[(x-\mu_q)^2\right]$$

Since

$$\sigma_q^2 = E_q\left[(x-\mu_q)^2\right]$$

we receive,

$$-D_{KL}\left[q_\theta(z|x_i)||p(z)\right] = \log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{1}{2\sigma_p^2}E_q\left(\underbrace{x-\mu_q}_{a} + \underbrace{\mu_q-\mu_p}_{b}\right)^2 + \frac{1}{2}$$

By applying $(a+b)^2 = a^2 + 2ab + b^2$ to the expression in expectation brackets we simplify further and get,

$$-D_{KL}\left[q_\theta(z|x_i)||p(z)\right] = \log\left(\frac{\sigma_q}{\sigma_p}\right) - \frac{\sigma_q^2 + (\mu_q-\mu_p)^2}{2\sigma_p^2} + \frac{1}{2}$$

Taking $\mu_p = 0$ and $\sigma_p = 1$, 22stima,

$$-D_{KL}\left[q_\theta(z|x_i)\,||p(z)\right]=\frac{1}{2}\left(1+\log(\sigma_q^2)-\sigma_q^2-\mu_q^2\right)$$

From ELBO given in equation (10). The objective to be maximised can be given as:

$$\frac{1}{2}\left(1+\log(\sigma_j^2)-\sigma_j^2-\mu_j^2\right)+E_{\sim q_\theta(z|x_i)}\left(\log p_\phi(x_i|z)\right)$$

where σ_j and μ_j are parameters of approximate distribution q, and j is the index of the latent vector z.

The objective function can be given as:

$$\sum_{j=1}^{J}\frac{1}{2}\left(1+\log(\sigma_j^2)-\sigma_j^2-\mu_j^2\right)+\frac{1}{L}\sum_{l}E_{\sim q_\theta(z|x_i)}\left(\log p(x_i|z^{(i,l)})\right)$$

where J is the dimension of the latent vector z, and L is the number of samples drawn through re-parametrization trick. This objective function is maximised during the training. To get the loss function we take the negative of this term:

$$\mathscr{L}=\sum_{j=1}^{J}\frac{1}{2}\left(1+\log(\sigma_i^2)-\sigma_i^2-\mu_i^2\right)+\frac{1}{L}\sum_{l}E_{\sim q_\theta(z|x_i)}\left(\log p(x_i|z^{(i,l)})\right)$$

The VAE is thus trained to get the optimal network parameters (θ^*,ϕ^*) that minimise $\mathscr{L}$:

$$(\theta^*,\phi^*)=argmin_{(\theta,\phi)}\mathscr{L}(\theta,\phi)$$

Architecture

In this thesis, we use tanh activations in all hidden layers, linear and sigmoid activations in the μ output layer for the encoder and decoder respectively, and linear activations in all log σ layers. Mathematically, tanh activation can be represented as:

$$f(x)=\frac{\left(e^x-e^{-x}\right)}{\left(e^x+e^{-x}\right)}$$

whilst the sigmoid function is defined as:

$$f(x)=\frac{1}{1+e^{-x}}$$

and linear activation function simply means:

$$f(x)=x$$

The MLP models are trained with the 'Adam' optimizer (Kingma, 2014) using constant 0.01 learning rate. Adam, short for Adaptive Moment Estimation, is an algorithm for gradient-based optimization that adapts the learning rate for each of the weights in the model, which often leads to more efficient learning.

We test three different *z* dimensions. The specific *z* dimensions vary depend on the dataset in use. The hidden layer size also varies depend on the dataset in use. The VAE we tested for the Norwegian data set, the Taiwanese data set and Polish data set are shown in Table 3.

Table 3

Summary of Tested Architectures

Architecture ID	z dimension	Neurons
N1	20	45
N2	25	45
N3	30	45
T1	30	70
T2	45	70
T3	60	70
P1	20	50
P2	30	50
P3	40	50

Classification Algorithms

Logistic Regression

The application of Logistic Regression (LR) for default prediction with a novel set of financial ratios as inputs was first introduced by (Ohlson, 1980). Logistic regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. The dependent variable in Logistic Regression is a binary variable, meaning it can take on one of two values, such as "success" or "failure." The independent variables, also known as predictors, can be any type of variable, such as continuous, categorical, or a combination of both. Whilst the models like Multivariate Discriminate Analysis when used for default prediction as in (Altman, 1968) generate a score that is then used for classifying, whereas Logistic Regression is the probability of the default. The Logistic Regression model is based on the sigmoid function:

$$f(x) = \frac{1}{(1 + e^{-x})}$$

that maps any input value to a value between 0 and 1.

Let us denote our input feature vector as $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ where n is the number of features. Each feature is associated with a weight or coefficient, denoted as $\mathbf{w} = (w_1, w_2, \ldots, w_n)$. The bias term is denoted as b. The Logistic Regression

model predicts the probability of the outcome variable y belonging to class 1, denoted as $P(y=1|\mathbf{x})$. This probability is modelled using the sigmoid function, which maps any real-valued input to the range $(0, 1)$:

$$P(y=1|x) = \frac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}}$$

The Logistic Regression model assumes that the outcome variable follows a Bernoulli distribution. Therefore, the probability of the outcome belonging to class 0 can be computed as:

$$P(y=0|\mathbf{x}) = 1 - P(y=1|\mathbf{x})$$

During the training phase, the logistic regression model aims to find the optimal values for $\mathbf{w}$ and b that maximize the likelihood of the observed data. This is typically done by minimizing the negative log-likelihood (or equivalently maximizing the log-likelihood):

$$\text{Loss} = -\sum_{i=1}^{m}\left[y^{(i)}\log(P(y=1|\mathbf{x}^{(i)})) + (1-y^{(i)})\log(1-P(y=1|\mathbf{x}^{(i)}))\right]$$

Here m represents the number of training examples, $\mathbf{x}^{(i)}$ is the feature vector of the i^{th} example, and $y^{(i)}$ is the corresponding true class label i.e., 0 or 1. The optimization process, such as gradient descent, is then used to update the values of $\mathbf{w}$ and b iteratively, minimizing the loss function. This allows the logistic regression model to learn the optimal decision boundary that separates the two classes in the feature space.

The Logistic Regression model can be used for binary and multiclass classification problems (Escalona-Morán et al., 2015). In binary classification, the model is used to predict the probability of one of two outcomes. In multiclass classification, the model is used to predict the probability of one of more than two outcomes. One of the benefits of Logistic Regression is that it is relatively simple to implement and interpret. The model can be fit using maximum likelihood estimation, and the coefficients of the independent variables can be used to estimate the effect Of each variable on the outcome. Additionally, Logistic Regression can be regularized to prevent overfitting, improving the model's performance.

Random Forest Classifier

In a Random Forest, many decision tree classifiers are built using two elements of randomness. Firstly, every tree is trained on a bootstrap replicate of the initial dataset and a random subset of independent variables, which bolsters the model's diversity and resilience. The other aspect of randomness is attribute

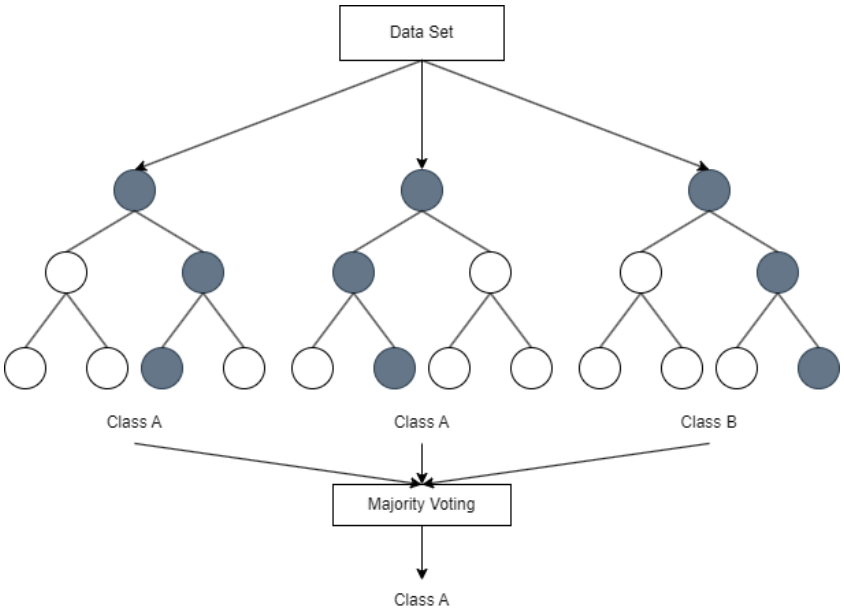
sampling. At each node split, a subset of the input variables is randomly selected to find the optimal split. The value proposed by Breiman (2001) to be given to this parameter is:

$$\log_2 M + 1$$

where M is the number of inputs. The final classification is determined by majority voting among the decision trees. By introducing randomness in both the training samples and feature space, Random Forest aims to enhance generalization performance by reducing variance whilst maintaining or slightly increasing bias. Each internal node of the tree represents a test on an input feature, and each leaf node represents a prediction. The decision tree grows by repeatedly splitting the data based on the feature that maximizes impurity reduction.

Figure 4

Diagram of Random Decision Forest



Random Forest has multiple benefits compared to individual decision trees. Firstly, it is less susceptible to overfitting due to the model's variance being minimized by averaging numerous trees. Furthermore, the model enhances precision by decreasing the bias. Random Forest can process both categorical and numerical variables, and it is capable of dealing with missing information and outliers. Lastly, the model possesses inherent feature selection, enabling it to manage high-dimensional data by selecting the most pertinent features for predictions. One of the advantages of Random Forest is that it functions reasonably well with default parameters (Fernández-Delgado et al., 2014). However, we can tune the quantity of random attributes selected for each split, and parameters that

control the depth of the decision tree. Normally, decision trees in a Random Forest grow until each leaf is pure, which could result in excessively large trees. To prevent this, the tree's growth can be restricted by establishing a maximum depth or necessitating a minimum number of instances per node before or after a split. In pursuit of the optimal model, we adopted a grid search strategy. This involved testing myriad combinations of parameters using a five-fold cross-validation technique, and subsequently deploying the most effective parameters for final result generation.

The 'number of 27stimateors' parameter dictates the quantity of trees in the forest. We trialled three options: 100, 300, and 500. Each value was assessed to evaluate the model's performance with differing quantities of decision trees. The intention was to strike a balance between possessing an ample number of trees for adequately discerning data patterns, and circumventing unduly extensive forests that may result in augmented computational burden and heightened risk of overfitting. The 'maximum depth' parameter constrains the maximum depth of each tree. We experimented with three depth tiers: 4, 6, and 8. Imposing limits on the depth of our trees ensures that the model does not overfit the training data, thereby enhancing generalization to unseen data. The 'criterion' parameter refers to the function utilized to ascertain the quality of a split. We considered two types of criteria: 'Gini' and 'Entropy'. The Gini impurity quantifies the probability of incorrect classification of a randomly selected element if it was randomly labelled in accordance with the label distribution in the subset. Conversely, Entropy is a metric of information gain that directs towards the most homogeneous branches. The final parameter, 'maximum features', determines the number of features to consider when searching for the most effective split. We elected to test three methods: 'auto', 'sqrt', and 'log2'. 'Auto' simply considers all features, 'sqrt' uses the square root of the total number of features, and 'log2' employs the base-2 logarithm of the number of features. The concept behind using fewer features is to introduce a degree of randomness into individual tree creation and enhance model robustness by reducing the variance.

Through the use of grid search, we methodically tested all combinations of these parameter values – a total of 54 combinations – in order to pinpoint the set that yields the best performance as per a specified scoring metric. This comprehensive process facilitated the maximization of the Random Forest model's performance.

Table 5

Grid Search for Optimal Parameters in Random Forest Classification Model

Number of Estimators	100, 300, 500
Maximum Depth	4, 6, 8
Criterion	Gini, Entropy
Maximum Feature	Auto, Sqrt, Log2

Extreme Gradient Boosting

Extreme Gradient Boosting is a model that falls under the category of decision tree-based machine learning methodologies. Decision trees are intuitive and straightforward supervised machine learning techniques that can tackle both regression and classification problems. In the context of classification, a single decision tree segregates an observation based on predefined conditions or "if-else" rules. These trees typically have a root node, internal nodes, branches, and leaf nodes. The internal nodes represent a test or condition on a specific attribute, the branches reflect the outcome of that test, and leaf nodes illustrate the class label.

Compared to many other machine learning strategies used for predictive analysis, decision tree-based approaches often boast superior predictive power. The decision tree model structure brings several advantages, especially pertinent to our analysis. For instance, these models can inherently handle missing values, as they can learn branch directions for missing values during training. Moreover, due to their inherent structure, decision tree-based models are generally more robust to multicollinearity than models like generalized linear models that assume feature independence. This robustness makes them an intriguing choice when we incorporate sentiment variables that could be somewhat interrelated. However, there are some downsides.

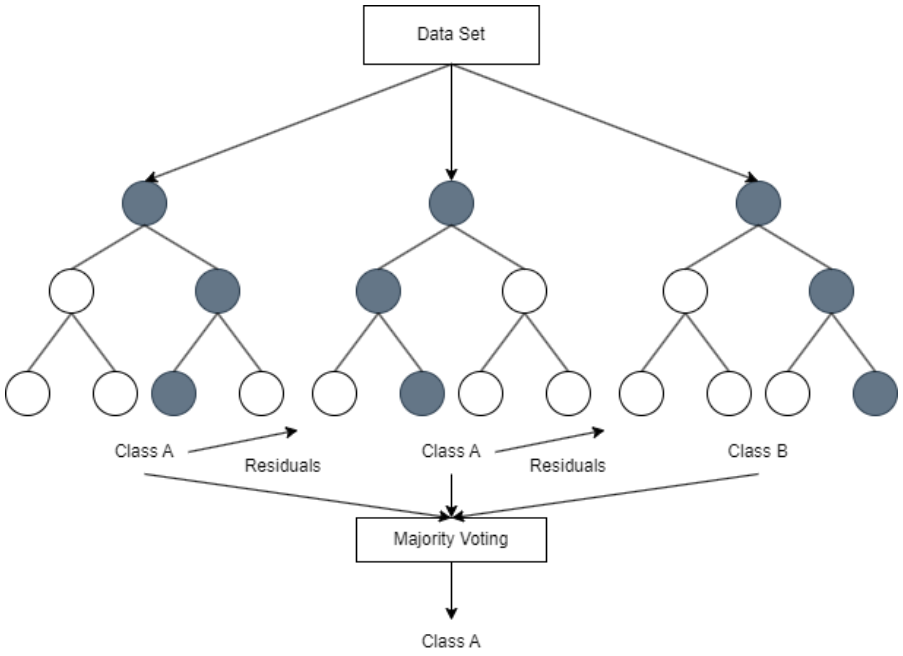
XGB is built on the principle of boosting, a technique applied when constructing multiple trees. Boosting aims to enhance the model using insights from previously built trees. By combining numerous individual trees, we can derive a single, consensus prediction, significantly enhancing accuracy but sacrificing some interpretability. Decision tree-based models, such as XGB, can be difficult to interpret, particularly in terms of understanding the individual effects of explanatory variables.

Specifically, when building subsequent trees, the method uses residuals from the previous tree. Each tree is built sequentially, leading to diminishing residuals as more trees are added. The model is trained on the unexplained variance

in the dataset, allowing the model to improve in areas where it previously underperformed. In essence, by combining multiple weak learners (individual trees), we can create a unified, robust learner. The enhancements incorporated in XGB include a regularization term to prevent overfitting and a second-degree approximation to boost performance relative to standard gradient boosting machines. Akin to gradient boosting machines, weights w_i are assigned to each observation i. These weights are then used in L2 norm regularization, a technique akin to the least squares method, which aims to discourage model complexity by penalizing models in proportion to the square root of the sum of the weights w.The second-degree approximation simplifies the existing objective function proposed by Friedman et al. (2000). This simplification not only reduces computation time but also improves prediction accuracy (Chen & Guestrin, 2016).

Figure 6

Diagram of Extreme Gradient Boosting



Analogous to the Random Forest Classifier, we employed a grid search technique for optimizing the XGB model. The hyperparameters under investigation included the number of estimators, the maximum depth, and the learning rate. The number of estimators parameter in XGB classifier designates the count of gradient boosted trees. The options for this study included 100, 300, and 500. The quantity of trees is a pivotal parameter, as insufficient trees may lead to underfitting, whilst an excessive number could precipitate overfitting and extended training duration. Our goal, therefore, was to ascertain an optimal equilibrium. The maximum depth

parameter signifies the maximum depth of each tree. This parameter was varied across three tiers which are 4, 6, and 8. The tree's maximum depth influences the model's complexity. A deeper tree can encapsulate more intricate patterns, potentially resulting in overfitting, whilst a shallow tree may fail to capture vital patterns, leading to underfitting. Finally, the learning rate parameter determines the step size shrinkage applied at each update, functioning as a form of regularization. This was varied across three levels which are 0.1, 0.05, and 0.01. A smaller learning rate necessitates additional boosting rounds, which can empower the model to discern more complex patterns. However, it also augments the risk of overfitting and computational burden.

In total, we tested 27 different combinations of parameters the find the best candidate to generate the final results. Our approach to parameter optimization ensured a rigorous and systematic exploration of the hyperparameter space, resulting in a finely tuned and robust XGB classification model.

Table 7

Grid Search for Optimal Parameters in XGB Classification Model

Number of Estimators	100, 300, 500
Maximum Depth	4, 6, 8
Learning Rate	0.1, 0.05, 0.01

Multi-layers Perceptron

The fundamental constituent of a Multi-layers Perceptron (MLP) is a neuron, or a node. Each neuron processes a set of inputs and generates an output. Neurons are organized into layers, which include an input layer, one or more hidden layers, and an output layer. The input layer receives the raw input from the dataset, with each node in this layer corresponding to one feature in the dataset. This is followed by the hidden layers, wherein each node performs computations on the data passed from the previous layer and forwards the result to the subsequent layer. These computations are dictated by parameters or weights that the network acquires during training. Finally, the output layer generates the result of the network's computations. The quantity of nodes in this layer is equivalent to the number of possible outputs.

Each neuron performs a simple computation. First, each input is multiplied by a weight. The results are then summed, and a bias term is added to the sum. This value is then passed through an activation function, which produces the neuron's output. If we denote the inputs to a neuron as $x_1, x_2, \ldots, x_n$, the wights as

$w_1, w_2, \ldots, w_n$, the bias as b, and the activation function as f, then the output of the neuron, y, can be computed as:

$$y = f(w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b)$$

Training a neural network involves iteratively adjusting the weights and biases to minimize the difference between the network's predictions and the actual data. This is typically done using gradient descent, where the weights and biases are adjusted in the direction that most decrease the error. In the case of a classification task, such as bankruptcy prediction, the output layer would have two neurons representing the two possible classes (bankrupt or not bankrupt). The activation function for the output layer is typically the softmax function,

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_i}} \text{ for } i = 1, \ldots, K \text{ and } z = (z_1, \ldots, z_k) \in \mathbb{R}^k$$

which produces a probability distribution over the possible classes. The softmax function essentially applies the standard exponential function to each element, denoted as z_i, of the input vector, represented as z. It then normalizes these values by dividing each one by the total sum of these exponentials. This normalization process guarantees that the sum of all components of the output vector, represented as $\sigma(z)$, is equal to 1. This is essentially a way of transforming the outputs into probabilities of the input belonging to each class, which is why it is particularly useful in multi-class classification problems. Given the inputs, the network is then trained to maximize the probability of the correct class.

In bankruptcy prediction, the goal is to classify companies into two categories: those that will go bankrupt and those that will not, based on various financial indicators. This is a binary classification problem, which is a common use case for neural networks. The input to the network would be the financial indicators for a company. The network would then learn patterns in these indicators that predict bankruptcy. For instance, one might learn that a high debt-to-equity ratio and declining profits are indicative of bankruptcy. The network's output would be the company's probability of going bankrupt. The network's weights and biases would be adjusted during training to minimize the difference between these predicted probabilities and the actual outcomes.

Since we are using VAE latent embeddings as input for the MLP there are a few factors to consider:

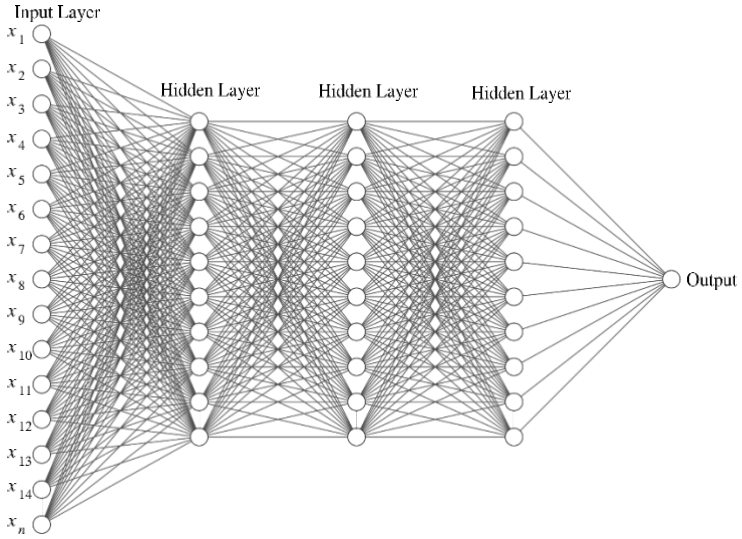
- Quality of the latent embeddings: If the VAE is well-trained, it should learn a useful latent space where similar data points are close together, and

different data points are further apart. In this scenario, the MLP could potentially perform better on the latent embeddings than on the original data, especially if the original data is very high-dimensional. High-dimensional data can suffer from the "curse of dimensionality," where the data becomes sparse and it's hard for a model to learn meaningful patterns. The latent embeddings could alleviate this issue by providing a more compact, dense representation of the data.

- Loss of information: Compression of data into a lower-dimensional space usually involves some loss of information. This is particularly true if the original data is very high-dimensional and complex. The lost information might be irrelevant noise, but it could also be useful information for your prediction task. If that's the case, then the MLP might perform worse on the latent embeddings than it would on the original data.
- Overfitting: MLPs with multiple hidden layers have the capacity to model complex, non-linear relationships, but they can also be overfit to the training data, especially when the amount of data is small relative to the complexity of the model. Using lower-dimensional embeddings can help to mitigate this issue, by reducing the complexity of the input space and hence the capacity of the model.

Figure 8

Graphical Presentation of Multi-Layer Perceptron for Binary Classification



Within the scope of this thesis, a Multi-layer Perceptron Classifier was utilized, composed of three hidden layers (Figure 8). The Rectified Linear Units

(ReLU) function was selected as the activation function for these hidden layers due to its efficiency and effectiveness in training deep neural networks. Introduced by Hahnloser et al. (2000), ReLU is given by the formula:

$$ReLU(x) = max(0, x)$$

For the output layer, the sigmoid activation function was employed to ensure output values fall within the range of 0 to 1, facilitating their interpretation as probabilities. L2 regularization, with a strength of 0.001, was applied as a preventive measure against overfitting by adding a penalty term to the loss function proportional to the magnitude of the coefficients. The Adam optimizer was chosen for the task of minimizing the loss function due to its efficient memory usage and suitability for problems with large amounts of data or parameters. The training process was conducted over a total of 500 epochs to enable adequate learning while preventing excessive computational costs.

Sampling techniques

Class imbalance constitutes a prevalent issue in machine learning, characterized by unequal representation of classes (Provost & Fawcett, 2001; Elrahman & Abraham, 2014; Pirizadeh et al., 2021). Predominantly, machine learning algorithms are architected to minimize the overall error rate. Consequently, in the presence of class imbalance, these algorithms may exhibit a bias towards the majority class, often compromising the performance on the minority class. This phenomenon manifests prominently in our chosen application, where a model could ostensibly achieve an accuracy rate of 98% by categorically predicting every instance as class A (non-bankrupt) yet fail to discern any instances of class B (bankrupt), which constitutes the primary objective of our model.

Sampling methods, considered as preprocessing techniques in machine learning, are employed to manage the challenges posed by class imbalance. These techniques operate by altering the training dataset to balance the distribution of minority and majority classes. The two principal types of sampling techniques are under-sampling and over-sampling.

Under-sampling functions by reducing the number of samples from the majority class to balance the dataset, whereas over-sampling augments the number of samples in the minority class. In situations of severe class imbalance, over-sampling is often the favoured choice. Nonetheless, both methodologies are associated with potential drawbacks (Elrahman & Abraham, 2014). Under-

sampling might inadvertently eliminate important patterns, potentially resulting in a loss of valuable information. On the other hand, over-sampling could give rise to overfitting and impose an additional computational burden. Therefore, it is crucial to carefully choose and apply these methods, taking into consideration the specific characteristics and requirements of the given problem.

Synthetic Minority Over-sampling Technique

Chawla et al. (2002) addressed the issue of class imbalance by proposing the Synthetic Minority Over-sampling Technique (SMOTE), a method that generates synthetic examples rather than merely duplicating instances. By doing this, SMOTE enables the identification of more distinctive regions in the feature space of the minority class. This not only optimizes classifier performance but also shifts the learning bias towards the minority class.

The procedure for synthesizing new samples can be explicated as follows. Initially, a random sample, denoted as x_0, is chosen from the minority class. Subsequently, the K-closest neighbours to x_0, all belonging to the same minority category, are determined. The Euclidean distance is used for this process due to its simplicity and effectiveness. The Euclidean distance between two points, say x_0 and x_i, in an n-dimensional space is calculated as:

$$d\left(x_0, x_i\right) = \sqrt{\sum_{j=1}^{n}\left(x_{0_j} - x_{i_j}\right)^2}$$

After calculating the distances, we sort them in ascending order. The first K instances in this sorted list are the K-nearest neighbours of x_0. From these K-closest instances, one is chosen at random, with this chosen instance being denoted as x_k, where k represents the rank of the selected neighbour. Following this, a linear interpolation between x_0 and x_k is executed, leading to the generation of a new synthetic sample, designated as *z*, utilizing the formula:

$$z = x_0 + w\left(x_k - x_0\right)$$

Here, *w* symbolizes a uniformly distributed random variable within the range of $[0, 1]$. This sequence of operations is iteratively carried out until the total count of synthetic samples and the instances in the minority class equals the count of instances in the majority class.

In the context of a bankruptcy prediction model, Synthetic Minority Over-sampling Technique (SMOTE) is applied by first identifying bankrupt companies

as instances of the minority class. Following this, SMOTE is utilized to generate synthetic examples of this class. The output is a dataset with a balanced class distribution, which is anticipated to yield a model more proficient at predicting bankruptcies.

A critical aspect of the application of SMOTE is its restricted use only on the training subset of the data, as opposed to the entire dataset. The order of this operation is of utmost importance. Were we to oversample prior to partitioning the data, we could inadvertently introduce identical instances in both the training and testing subsets. This could lead to overestimation of model performance due to a phenomenon known as information leakage. Hence, the prescribed procedure entails initially separating the data into training and testing sets. Post this segregation, SMOTE is applied solely on the training set. Adherence to this methodology assures an accurate validation process when evaluating the model, as it negates the chance of replicating identical instances across both subsets (Oreški, 2014).

Random Under-Sampling

Random Under-Sampling (RUS) serves as a technique that employs under-sampling. It aims to establish a balanced class distribution by decreasing the quantity of instances in the majority class. Notably, RUS has proven instrumental in tackling issues associated with class imbalance in diverse domains. In certain instances, it is utilized conjointly with other methodologies to enhance model performance (Hasanin & Khoshgoftaar, 2018; Hasanin et al., 2019)

Although the implementation of RUS is more straightforward and expeditious compared to SMOTE, it has an inherent disadvantage. Particularly, it poses the risk of disregarding valuable data, as it eliminates instances from the majority class indiscriminately (Dittman et al., 2014). The randomization aspect in under-sampling is critical to the modus operandi of the method, and its subsequent impact on the model. In the procedure of RUS, instances from the majority class are randomly selected and eliminated until the number of instances in the majority class equals those in the minority class.

Within the theoretical framework of a model predicting corporate bankruptcy, it is expected that non-insolvent enterprises will constitute the preponderant class. Consequently, instances from this predominant class would be subjected to stochastic elimination. Drawing a parallel with the utilization of

SMOTE, it becomes essential to restrict the application of RUS exclusively to the training dataset. This is to thwart any possible leakage of data, ensuring the integrity of the predictive model.

Performance Metrics

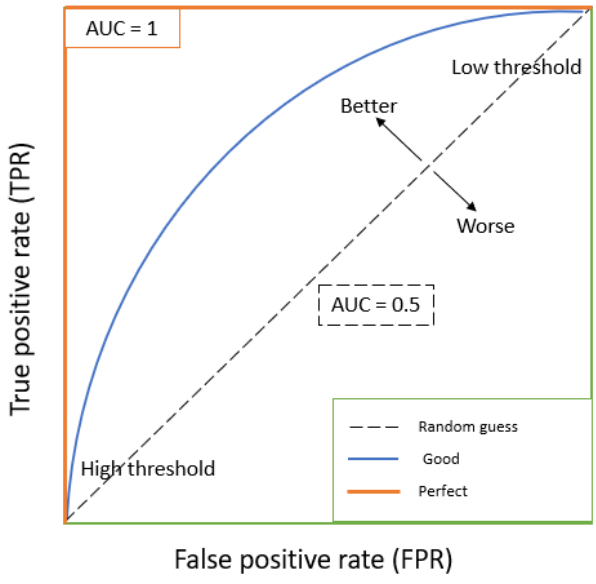
It is vital to choose a bankruptcy prediction model for an accurate out of sample prediction power. In this study, we assess the out-of-sample prediction performance of various models by splitting the data into training and testing datasets. we randomly partition the dataset, assigning 80% as the training set and the remaining 20% as the testing set. This approach has been commonly used in prior research (Doumpos et al., 2017; du Jardin, 2016). We evaluate each model's prediction ability using metrics such as out-of-sample area under the receiver operating characteristic curve (AUC), H-measure and KS score.

Area Under Receiver Operating Characteristic Curve

Receiver Operating Characteristic curve (ROC) along with the accuracy is the most commonly used measure to assess the performance of prediction model (Marqués et al., 2012). ROC curve is a graphical representation that showcases the performance of a classification model at various classification thresholds. It plots the true positive rate (TPR) against the false positive rate (FPR) at different thresholds, thereby displaying the trade-off between sensitivity (TPR) and specificity (1 - FPR).

Figure 9

Visualisation of ROC Curve



As demonstrate in Figure 9, the ROC curve represents all possible classification thresholds, demonstrating how these thresholds influence both the TPR and FPR.

One of the main advantages of an ROC curve is that it allows the users to choose the trade-off between sensitivity and specificity based on the model's specific objectives. However, increasing the sensitivity of the model will often decrease its specificity, and vice versa. Another advantage of ROC curves is their usefulness when dealing with imbalanced datasets and uneven classification error costs. ROC curves are not sensitive to changes in class distribution, which means that they stay constant even when the ratio of negative to positive observations in the data changes.

AUC, or the Area Under the ROC Curve, is a widely used metric to assess the overall discriminatory power of a model (Bradley, 1997). The metric arises from the ROC curve, measuring the model's ability to correctly classify positive and negative observations. The AUC value ranges between 0 and 1, with 1 indicating a perfect model and 0.5 representing a model whose performance is no better than random chance.

A model with an AUC of 1 will have an optimal point in the upper-left corner of the ROC space, reflecting a 100% TPR and 100% TNR. On the other hand, an AUC of 0.5 corresponds to a model that predicts positive and negative classes at the same rate as random guessing. A model with an AUC of 0.75 suggests that it has a reasonable ability to differentiate between classes.

We can calculate the AUC curve on the basis of integrating the areas of small trapezoidal bins from the ROC curve. That is to say:

$$AUC(T) = \sum_{\kappa} \overline{P}_D(T)\, \Delta P_{FA}(T)$$

where:

$$\Delta P_{FA}(T) = P_{FA}(\kappa+1)(T) - P_{FA}(\kappa)(T)$$

and,

$$\overline{P}_D(T) = \frac{P_D(\kappa+1)(T) + P_D(\kappa)(T)}{2}$$

- $AUC(T)$ denotes Area Under the Curve (AUC) at a specific threshold (T). It quantifies the overall performance of the binary classifier.
- κ denotes each individual trapezoid under the ROC curve.
- $P_D(T)$ denotes the true positive rate at threshold T.
- $P_{FA}(T)$ denotes the False positive rate at threshold T.

- $\Delta P_{FA}(T)$ denotes the width of each trapezoid, calculated as the difference in the false positive rate between the current and next threshold.
- $\overline{P}_{D}(T)$ denotes the average height of each trapezoid, calculated as the average of the true positive rates at the current and next threshold.

AUC is a widely adopted metric in bankruptcy prediction and financial markets. This preference comes from the fact that AUC is robust to imbalanced data, which is common in scenarios like bankruptcy prediction where the distribution of companies that go bankrupt versus those that don't is uneven. Therefore, in addition to balanced accuracy, AUC proves to be a useful performance measure for imbalanced classification problems**.**

When dealing with rare events like bankruptcy, relying solely on classification accuracy can be misleading. This is because accuracy assumes equal costs for both false positives and false negatives. However, the consequences of false negatives are usually more severe than false positives. Whilst it is possible to assign a higher cost to false negatives in certain cases, this cost structure remains specific to the context (du Jardin, 2016). Additionally, decision-makers often require more than a binary bankruptcy prediction.

The probability of bankruptcy can be valuable for constructing credit portfolios or determining loan interest rates (Hillegeist et al., 2004). Therefore, AUC provides a more flexible performance measure by utilizing the ROC curve. The ROC curve illustrates the trade-off between the false positive rate and the true positive rate across different decision criteria or cutoff probabilities. AUC represents the area under this curve and allows for evaluating a model's overall performance without assuming a specific cost structure. Typically, the AUC score ranges from 0.5 to 1, where 0.5 indicates random assignment of class labels and 1 suggests perfect classification.

H-Measure

The H-Measure, introduced by Hand (2009), is a widely utilized performance measure for classifiers. It aims to address the cost of misclassification without necessitating fixed values. This measure emerged as an improvement over traditional methods, specifically addressing the limitations of the AUC method by incorporating the costs associated with different types of misclassification errors. The H-measure is defined by the following formula:

$$H = 1 - \frac{L}{L_{\max}}$$

In this formula, the loss values for a distribution of scoring points derive from a monotonous distribution. The H-measure values range between 0 and 1, with higher values indicating the model's enhanced discriminatory power.

Ideally, with known costs, finding the optimal threshold would render the problem as simple as summarizing a standard confusion matrix. However, in real-world scenarios such as medical diagnostic systems, precise cost values often remain unknown. The severity of future misclassification depends on varying treatment options. To manage such circumstances, the H-Measure computes an expectation over a distribution of potential cost values. Researchers are advised to select a distribution, similar to a Bayesian prior, based on their understanding of the problem at hand. A standard default distribution is also recommended for typical measurement, with a specific form of the beta distribution proposed for this purpose.

The method has been generalized to accommodate scenarios with unknown class sizes. Furthermore, as Buja et al. (2005) demonstrated, selecting different cost distributions is akin to using distinct loss functions for estimating class membership probabilities in classifiers. This suggests that minimizing log-loss for a neural network and squared error loss for a random forest might be driven by unique motivations.

In essence, the H-Measure serves as an alternative metric quantifying the predictive capability or discriminatory power of classification models. It evaluates a model's ability to differentiate between different classes. Traditional metrics such as the AUC, despite their popularity, grapple with inherent inconsistency issues due to variations in the proportions of true positive and true negative rates. By computing the potential loss incurred through incorrect classifications, the H-Measure effectively quantifies the cost or penalty of misclassification. It offers a more nuanced performance measure than merely tallying correct or incorrect predictions.

One crucial feature of the H-measure is its dependence on the proportion of entities classified into each class. If a model skews towards one class, the H-Measure reflects this imbalance, thereby encouraging balanced classification. This aspect sets it apart from other metrics that often overlook class distribution. The H-Measure is especially beneficial when dealing with imbalanced datasets like

bankruptcy datasets, where one class of observations greatly outnumbers the other. In such instances, a model may achieve misleadingly high accuracy if it primarily classifies observations as the majority class. By focusing on the potential loss from misclassifications and accounting for the proportions of class predictions, the H-Measure provides a robust and insightful method to assess model performance. It is particularly effective when handling class imbalances or when misclassification costs are significant.

Kolmogorov-Smirnov

The Kolmogorov-Smirnov (KS) test is a powerful statistical method used to assess the similarity between two probability distributions or to test whether a sample follows a specific distribution (Liu, 2018). It is a nonparametric test, meaning that it does not rely on any assumptions about the underlying distribution of the data. The KS was originally introduced as an adherence hypothesis test for distribution fitting to data. For binary classification problems it is used as a dissimilarity metric for evaluating the classifier's discriminant power measuring the distance that is represented by the scores given for the two cumulative distribution functions (CDFs).

Let us consider the case where we have two samples, each obtained from a different population. We want to determine if these two populations are significantly different from each other. The KS test allows us to compare the CDFs of the two samples.

The CDF of a random variable X is defined as the probability that X takes on a value less than or equal to a given value x. Mathematically, it can be represented as:

$$F_X(x) = P(X \leq x)$$

Similarly, for a random variable Y, the CDF is denoted as $F_Y(y)$. The KS test assesses the maximum absolute difference between the two CDFs, which is known as the KS statistic D:

$$D = max(|F_X(x) - F_Y(y)|)$$

The KS test quantifies the probability of obtaining a KS statistic as extreme as D or more extreme, assuming that the two samples come from the same distribution. If the p-value is below a predetermined significance level (e.g., 0.05), we reject the null hypothesis, indicating that the two populations differ significantly.

The KS test also has applications as a goodness-of-fit test. In this scenario, we have only one random sample, and we want to assess whether it follows a specific distribution. We compare the empirical CDF (ECDF) of the sample to the expected CDF of the hypothesized distribution. The ECDF is constructed by ordering the observed values and assigning a probability of $\frac{i}{n}$ to each observation, where i is the rank of the observation and n is the sample size. The KS test statistic is calculated as:

$$D = max(|F_X(x) - F_{expected}(x)|)$$

Here $F_X(x)$ represents the ECDF of the sample, and $F_{expected}(x)$ is the expected CDF based on the hypothesized distribution. A low p-value suggests that the sample does not conform well to the assumed distribution.

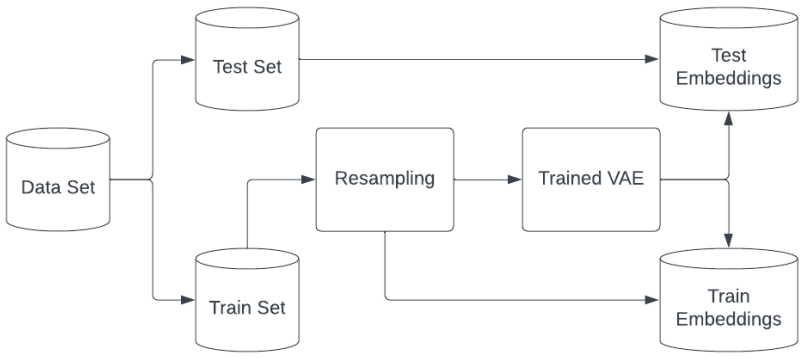
The KS test is used as a metric to evaluate the differences in distribution between bankrupt and non-bankrupt companies for individual financial variables, aiding in feature selection. On the other hand, the ROC curve and AUC-ROC assess the performance of the overall bankruptcy prediction model, considering the combination of multiple variables and the classification threshold. The ROC curve provides a visual representation of the model's discriminatory power, whilst the AUC-ROC summarizes its performance in a single metric.

Development Methodology

Each of the three datasets was bifurcated into two distinct subsets: a training set accounting for 80% of the data, and a testing set constituting the residual 20%. This conventional 80 to 20 partition was implemented to facilitate robust model learning whilst reserving a substantial segment of the data for assessing the models' generalization capabilities.

Figure 10

Data Preparation Flowchart: Sampling and Splitting (80/20 Ratio)



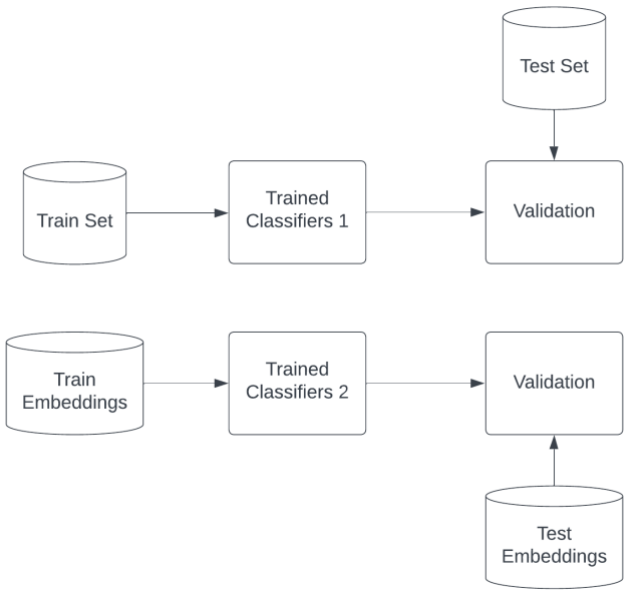
During the resampling process, three alternatives were examined. The first alternative entailed retaining the original training set. To assess the classifiers' robustness and ability to handle class imbalance, two different data balancing methods, Random Under Sampling (RUS) and Synthetic Minority Over-sampling Technique (SMOTE), were utilized (Figure 10). These techniques were employed on the training data before the training of the classifiers. The second alternative was to implement RUS on the training set, and the third involved the application of SMOTE.

Four distinct classifiers were selected for this study, specifically, the Random Forest Classifier (RFC), Extreme Gradient Boosting Classifier (XGBC), Multilayer Perceptron Classifier (MLPC), and Logistic Regression Classifier (LRC). Each of these classifiers was trained on the unprocessed training data for each dataset. To enhance the rigour of the study, the hyperparameters of the RFC and XGBC were optimized through the application of a grid search, coupled with five-fold cross-validation. Subsequent to their training, the classifiers were evaluated on the unprocessed test set (Figure 11).

Moreover, the training data was subjected to a Variational Autoencoder (VAE) to generate embedded representations. These encoded data representations were then employed to train the aforementioned quartet of classifiers (RFC, XGBC, MLPC, and LRC). Post-training, these classifiers were tested on the encoded test data to ascertain their performance.

Figure 11

Classifier Workflow: Training and Prediction



Additionally, to probe the influence of model complexity on performance, the experiment was also conducted employing three distinct hidden layer sizes for VAE. This step enables an understanding of how variations in model complexity impact the performance of the classifiers on the encoded data.

Experiment & Reproducibility

To manage the inherent variability that may arise during the data partitioning, sampling, training processes, and subsequent visualization, we deployed the use of a random state or seed. This is a common practice in machine learning to ensure that the randomness within stochastic processes is consistent and reproducible across different runs or even different machines. Setting a specific random state allows us to replicate our results later with precision, providing for both consistency in our work and facilitating peer review and independent verification of our results. This ensures the robustness of our findings and enhances the transparency and integrity of our research process.

To augment the dependability of the obtained outcomes, the experimentation was implemented on four distinct occasions to collate comprehensive statistical data on the performance metrics of the model. For each experimental iteration, a unique seed was employed to control the initialization and ensure the randomness of the trials. The overarching objective of this methodological approach was to extract a deepened understanding of the models' variability and consistency, which are essential parameters in assessing their reliability and predictability.

Data

In this thesis, we make use of three separate data sets, each encompassing bankruptcy information from distinct geographical territories.

- The 'Norwegian Bankruptcy Data set' is the first of these data sets. It is a comprehensive compilation of data extracted from two key sources: the Orbis Database and the Brønnøysund Register Centre (Brreg) in Norway.
- The second data set is the 'Taiwanese Bankruptcy Data Set.' This data set was procured from the UCI Machine Learning Repository, a renowned source for high-quality, pre-processed data sets.
- Lastly, we utilize the 'Polish Bankruptcy Data set,' which, similar to the Taiwanese Bankruptcy Data set, has been retrieved from the UCI Machine Learning Repository.

These three data sets collectively offer a broad and diverse range of insights into bankruptcy trends across various national economies, providing a robust foundation for our analysis in this thesis.

Data Source and Acquisition

Norwegian Data Set

The primary data set under discussion pertains to annual financial statistics and is henceforth referred to as the 'Annual Accounting Data set.' It encompasses a range of accounting data derived from the period between 2016 and 2021, inclusive, for all enterprises formally registered within the territorial jurisdiction of Norway. This invaluable information has been sourced from the renowned Orbis Database. Considering the global reach of the Orbis Database, which features data from corporations across the globe, specific parameters were established to focus our search. Our extraction criteria were designed to filter out companies based solely in Norway, including both active and inactive corporations spanning the years from 2016 to 2021. Additionally, to further refine our data, we strategically excluded enterprises within the financial institution and insurance sectors. A crucial aspect to consider is the intrinsic constraint within the Orbis database concerning the quantity of company data downloadable in a single instance. As we augment the volume of attributes selected for download, the permissible allowance decreases proportionately. Consequently, our strategy involved downloading only the most vital information, as outlined in Table 12, and subsequently recomputing relevant

ratios based on this downloaded data. Moreover, due to the restriction on the number of companies that can be downloaded at any given time, we engineered an automatic scraper tool. This tool is designed to streamline the data extraction process and amalgamate the data post extraction. Our downloaded data set is a comprehensive compilation comprising 1,274,594 active companies and 1,451,567 inactive companies across 26 distinct features. The attributes largely include standard accounting figures and key financial ratios, providing a holistic view of the corporate landscape in Norway during the specified period.

Table 12

Accounting variables in downloaded data set

Variable	Description
v1	Operating revenue (Turnover)
v2	P/L before tax
v3	P/L for period (Net income)
v4	Cashflow
v5	Total assets
v6	Shareholders' funds
v7	ROCE using P/L before tax
v8	Solvency ratio (Asset based)
v9	Number of employees
v10	Fixed assets
v11	Tangible fixed assets
v12	Current assets
v13	Cash and cash equivalents
v14	Current liabilities
v15	Net inventory
v16	EBITDA margin
v17	EBIT margin
v18	Interest cover
v19	Stock turnover
v20	Collection period (days)
v21	Credit period
v22	R&D expenses/Operating revenue
v23	Liquidity ratio
v24	Gearing
v25	Average cost of employee
v26	Working capital per employee

The secondary data set in our study pertains to bankruptcy data procured from the Brønnøysund Register Centre (Brreg) in Norway. This particular data set encapsulates a comprehensive record of 28097 Norwegian companies that declared bankruptcy during the period from 2018 to 2022. It provides detailed information about each bankrupt company, including the official name of the company, its

unique organization number, the corresponding industry codes, and the precise date on which bankruptcy proceedings were initiated (Appendix 1).

The time frame of 2016 to 2021 for Orbis data set was initially chosen for two principal reasons. First, this period encompasses the most recent data available at the time of our research, ensuring relevance and timeliness. Second, and equally important, Norwegian law stipulates that information concerning bankrupt companies is only publicly accessible for a period of five years. Consequently, the choice of this specific timeframe enhances the breadth and depth of our study, as it allows for the inclusion of bankruptcy data from Brreg.

Taiwanese Data Set

The Taiwanese Bankruptcy Prediction data set is a comprehensive collection of data aimed at predicting company bankruptcies (UCI Machine Learning Repository, n.d.-a). The data set was compiled from the Taiwan Economic Journal for the years 1999 to 2009, with company bankruptcy defined based on the business regulations of the Taiwan Stock Exchange. The data set is characterized as multivariate, with a total of 6,819 instances and 96 attributes, all of which are integers. It is primarily aimed at tasks involving classification and contains no missing values. The attributes in the data set represent a variety of financial metrics and ratios, including cost of interest-bearing debt, cash reinvestment ratio, current ratio, interest expenses to total revenue, total liability to equity ratio, and many others. Each attribute is represented by a code from X1 to X95 (Appendix 2). The first attribute is the class label, serving as the indicator of whether the company went bankrupt or not.

The data set was sourced from Deron Liang and Chih-Fong Tsai of the National Central University, Taiwan. It was used in a study titled "Financial Ratios and Corporate Governance Indicators in Bankruptcy Prediction: A Comprehensive Study", published in the European Journal of Operational Research (2016).

Polish Data Set

The Polish Companies Bankruptcy Data Set is a multivariate data set created for the purpose of predicting the bankruptcy of Polish companies (UCI Machine Learning Repository, n.d.-b). It was sourced from the Emerging Markets Information Service, a database that provides information on emerging markets worldwide. The data set provides information from the years 2000 to 2012 for bankrupt companies, and from 2007 to 2013 for companies that were still operating

during the evaluation period. The data set includes 43,405 instances, each characterized by 64 attributes, and the data does contain missing values (Appendix 3). These attributes include various financial ratios such as net profit to total assets, total liabilities to total assets, working capital to total assets, and many more. Each instance also has a class label that indicates the bankruptcy status of the company after a five-year forecasting period.

This data set was created and donated by Sebastian Tomczak from the Department of Operations Research at the Wrocław University of Science and Technology in Poland. The data has been used in research, such as in the paper "Ensemble Boosted Trees with Synthetic Features Generation in Application to Bankruptcy Prediction" published in Expert Systems with Applications (Zięba et al., 2016).

Pre-processing and feature engineering

Norwegian Data Set

Upon acquiring the data from Orbis, our initial task was to eliminate all observations with absent operating revenue values. The rationale for this decision is twofold: such a void often implies that the company either had not been established during the specific year or had declared bankruptcy. Consequently, when operating revenue is recorded as not available, other columnar values are typically absent as well, reflecting the company's inactivity or nonexistence. This procedure is consistent with the data representation standards of the Orbis Database. To initiate this process, the first step involves transforming the data from its wide format to a long format, whereby we extract the year information from the column names in the wide format. Let us consider an example of a company, denoted as Company A, which has six columns representing operating revenue data spanning from 2016 to 2021. This transformation will result in six separate observations for Company A, each containing two columns: year and operating revenue. By adopting this approach, we gain the ability to eliminate any years where the operating revenue data is not available, thereby ensuring the integrity and completeness of the data set.

Table 13

Additional accounting variables

Variable	Description	Formula
v27	Acid test ratio	$\frac{v12 - v15}{v14}$
v28	Net working capital	$v12 - v14$
v29	Net working capital turnover ratio	$\frac{v1}{v28}$
v30	Asset turnover ratio	$\frac{v1}{v5}$
v31	Fixed asset ratio	$\frac{v10}{v5}$
v32	Proprietary ratio	$\frac{v6}{v5}$
v33	Current ratio to fixed asset ratio	$\frac{v12}{v10}$
v34	Earning margin	$\frac{v3}{v1}$
v35	Cash flow margin	$\frac{v4}{v1}$
v36	Gross margin	$\frac{v2}{v1}$
v37	ROE using P/L before tax	$\frac{v2}{v6}$
v38	ROA using P/L before tax	$\frac{v2}{v5}$
v39	ROE using net income	$\frac{v3}{v6}$
v40	ROA using net income	$\frac{v3}{v5}$
v41	Capital employed	$\frac{v2}{v7}$
v42	ROCE using net income	$\frac{v3}{v41}$
v43	Current ratio	$\frac{v12}{v14}$
v44	Total liabilities	$\frac{v5}{v8}$
v45	Solvency ratio (liability based)	$\frac{v45}{v5}$
v46	Shareholder liquidity ratio	$\frac{v6}{v5}$
v47	Net assets turnover	$\frac{v1}{v5}$
v48	Fixed assets turnover	$\frac{v1}{v10}$
v49	Shareholder funds per employee	$\frac{v6}{v9}$
v50	Profit per employee	$\frac{v3}{v9}$
v51	Operating revenue per employee	$\frac{v1}{v9}$

v52	Total assets per employee	$\frac{v5}{v9}$
v53	Total employee costs	$v9 \cdot v25$
v54	Costs of employee to operating revenue ratio	$\frac{v53}{v1}$

Following the aforementioned procedure, we have addressed the issue pertaining to the absence of values in the remaining columns. Within both the accounting variables and the financial ratios, we encountered instances where data were missing, which posed a significant challenge to our analysis. To tackle this problem, we opted to assign a value of zero to all the missing data points, employing a direct and effective strategy to enhance the comprehensiveness of the data set. However, this approach introduces a predicament. The act of imputing zeros to the missing data points can potentially oversimplify the intricacies of the data set, leading to erroneous interpretations. Despite the associated risk, this method represents a necessary and expedient solution to effectively handle the issue of missing data, thereby ensuring the overall integrity and usability of the data set for subsequent analysis.

The subsequent phase of our research entails the computation of relevant accounting ratios. As previously noted, due to limitations in downloading a large number of companies simultaneously, additional computations were required on our part. Consequently, we undertook the task of identifying missing accounting variables and calculating supplementary accounting ratios based on these variables. The calculations for the new accounting variables are illustrated in Table 13. In cases where the denominator of a ratio is zero, we assign a value of zero to that particular feature. Through this meticulous process, our data set has been augmented with a comprehensive set of 28 diverse features. These encompassing features encompass a wide range of accounting ratios and variables, serving as the foundation for our modelling process and significantly enhancing the robustness and reliability of our subsequent analyses.

Next, we proceeded with the integration of our enhanced data set with bankruptcy data obtained from the Brønnøysund Register Centre (Brreg). First, let us define the ground truth class:

$$y = \begin{cases} 1 \; bankrupcy \\ 0 \; not \; bankruptcy \end{cases}$$

The integration was achieved by utilizing organization numbers as the primary identifier, ensuring a precise matching between the data sets. Through this

integration process, we established a new binary column within the data set, which serves as a predictive indicator of bankruptcy. In this column, a value of '1' is assigned to the most recent year's observation if the company is listed as bankrupt in the Brreg dataset and the year of bankruptcy differs from the final year of available accounting data. If these conditions are not met, the observation is assigned a value of '0'. Furthermore, we elected to exclude observations wherein the bankruptcy year corresponds with the final year of available accounting data. It is noteworthy that we adopted this methodology as opposed to employing the "succeeding year" criteria for the allocation of bankruptcy labels to organizations. This decision stemmed from the intricacies associated with bankruptcy procedures.

A salient challenge is the unpredictable timing of bankruptcy declarations, which can be influenced by an array of factors such as the duration of bankruptcy proceedings and administrative delays in the recordation of bankruptcies in statistical databases. As elucidated by the studies of Bernhardsen (2001), Wahlstrøm and Helland (2016), and Hjelseth and Raknerud (2016), there is often a temporal gap of one or two years, or sometimes even more, between the approval of the last set of financial accounts and the formal declaration of bankruptcy. Moreover, historical data from the aforementioned studies indicate that a minimum of 85% of insolvent firms officially declare bankruptcy within a two-year timespan. By incorporating this lag into our model, we endeavor to achieve a more accurate depiction of the bankruptcy status of firms, thereby accounting for any discrepancies or delays that may occur during the documentation of bankruptcy events.

Table 14

Bankruptcy Statistics by Year

Year	Total Observations	Total Bankruptcies	Bankruptcy Ratio
2016	221940	573	0.003
2017	275396	560	0.002
2018	298871	645	0.002
2019	324317	4,382	0.014
2020	334629	3,505	0.010
2021	346365	2500	0.007

At this juncture, our data set comprises 1,801,518 observations. However, the data set reveals a relatively low incidence of bankruptcies from 2016 to 2018. Upon cross-validation with the Brreg bankruptcy data and the Orbis database, it was observed that a substantial number of corporations within the Brreg data set

did not have corresponding accounting data in the Orbis database for the years 2016, 2017, and 2018. Consequently, this results in significant class imbalances for these particular years. Given this scenario, we decided to exclude these years from the final data set due to the inherent discrepancies and the potential skew they may introduce in the modelling process. Furthermore, we also chose to exclude the year 2021. This decision was predicated on the fact that our bankruptcy label data from Brreg only extends up to the year 2022. As per the previously discussed delays in bankruptcy declarations, the bankruptcy label for the year 2022 will not encompass an adequate representation of actual bankrupt corporations for the accounting data year of 2021. This approach is consistent with the methodology employed by Bernhardsen and Larsen (2007), further validating our data inclusion criteria. After these adjustments, the final data set comprises 658,946 observations. This refined data set not only ensures improved data quality, increasing the potential for more precise and meaningful analysis, but also provides a more manageable size for computational processing. Moreover, this selected period presents a unique opportunity for exploration, as it coincides with the timeline during which Norwegian businesses grappled with the policies implemented during COVID-19 pandemic (Ursin et al., 2020).

The final step in processing the dataset is to standardize the data to a mean of 0 and a standard deviation of 1 using the following formula:

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

where x represents an individual data point, μ is the mean of the data, σ is the standard deviation, and x_{scaled} represents the output.

Standardization is a critical preprocessing step when dealing with variables that have different units of measurement or scales. In our case, we decided to use both accounting variables and accounting ratios as input features. These two types of data operate on different scales, with accounting variables often taking on much larger values than ratios. Consequently, without standardization, variables with larger values could unduly influence the model, causing unstable learning. By standardizing, we are effectively rescaling our data to have a uniform scale, thereby ensuring all input features contribute equally to the final decision function. This step prevents any single feature from dominating the others due to its scale, thus allowing for a more balanced and accurate model performance.

Polish Data Set and Taiwanese Data Set

The data sets procured from the UCI Machine Learning Repository is already meticulously structured, thereby eliminating the need for additional feature engineering steps. Our intervention was limited to the removal of duplicate observations within the data. Following our established protocol, we addressed missing values by substituting them with a value of '0'. This consistent approach to handling missing data ensures uniformity across all data sets utilized in this thesis, facilitating a seamless comparative analysis. Lastly, similar to the Norwegian data set, we also perform scaling on the variables for these two data sets.

Data Quality Assessment

From Table 15, we observe a significant variation in the dataset size and bankruptcy ratio. This heterogeneity provides us with a unique opportunity to rigorously examine the impact of VAE latent embeddings on the performance of the selected classifiers. Such a variance within our dataset acts as a catalyst for a more comprehensive evaluation of our models, allowing us to discern how these classifiers react to differing conditions and data characteristics.

Table 15

Summary of Final Data Sets

Data Set	Prediction Horizon	Total Observations	Bankruptcy Ratio
Norwegian	12 months	658,946	0.012
Taiwanese	12 months	6,819	0.032
Polish	12 months	43,004	0.048

As depicted in Appendices 4, 5, and 6, there is evidence of multicollinearity in the Norwegian dataset, with somewhat more pronounced multicollinearity observed in the Taiwanese and Polish datasets. However, it is important to note that whilst multicollinearity can present challenges in some statistical analyses, it is not necessarily problematic in the context of the machine learning algorithms we employ in this study. The algorithms we utilize, logistic regression, random forest, XGB, and multi-layer perceptron, are generally robust to multicollinearity. Therefore, whilst we acknowledge the presence of multicollinearity in our datasets, we do not expect it to adversely affect the results of our machine learning models.

Results

In this section, outcomes derived from three distinct datasets are delineated. The structuring of these results is primarily organized by the dataset employed and the sampling techniques implemented. The term "VAE latent embeddings" is utilized to denote the latent representations procured through the use of the training dataset. The "raw training set" signifies the unaltered training dataset, whereas the "balanced training set" refers to the training dataset that has undergone balancing via the Synthetic Minority Over-sampling Technique (SMOTE) or the random under-sampling approach.

Norwegian Data Set

Without Rebalancing: Using Original Data

Table 16

Performance Measure – Norwegian Data Set Using VAE latent embeddings

Classifier	z	AUC	H-Measure	KS
LR	20	0.7619 (0.7537 – 0.7701)	0.2047 (0.1912 – 0.2182)	0.4053 (0.3870 – 0.4236)
LR	25	0.7639 (0.7594 – 0.7685)	0.2108 (0.2033 – 0.2183)	0.4153 (0.4039 – 0.4267)
LR	30	0.7692 (0.7590 – 0.7795)	0.2160 (0.2006 – 0.2314)	0.4205 (0.4020 – 0.4390)
RF	20	0.7620 (0.7566 – 0.7673)	0.2508 (0.2420 – 0.2595)	0.4160 (0.4084 – 0.4237)
RF	25	0.7615 (0.7579 – 0.7651)	0.2553 (0.2501 – 0.2605)	0.4143 (0.4051 – 0.4235)
RF	30	0.7664 (0.7583 – 0.7745)	0.2597 (0.2445 – 0.2749)	0.4227 (0.4069 – 0.4385)
XGB	20	0.7936 (0.7893 – 0.7979)	0.2548 (0.2508 – 0.2588)	0.4492 (0.4453 – 0.4531)
XGB	25	0.7992 (0.7958 – 0.8025)	0.2693 (0.2631 – 0.2756)	0.4588 (0.4495 – 0.4680)
XGB	30	0.8027 (0.7911 – 0.8143)	0.2762 (0.2549 – 0.2976)	0.4643 (0.4493 – 0.4792)
MLP	20	0.7976 (0.7928 – 0.8025)	0.2616 (0.2551 -0.2681)	0.4692 (0.4533 – 0.4717)
MLP	25	0.8011 (0.7978 – 0.8044)	0.2691 (0.2630 – 0.2752)	0.4719 (0.4656 – 0.4782)
MLP	30	0.8044 (0.7949 – 0.8140)	0.2787 (0.2630 – 0.2943)	0.4742 (0.4598 -0.4886)

Table 17

Performance Measure – Norwegian Data Set Using Raw Train Set

Classifier	AUC	H-Measure	KS
LR	0.7378 (0.7320 – 0.7435)	0.1710 (0.1632 – 0.1787)	0.3601 (0.3516 – 0.3687)
RF	0.8157 (0.8113 – 0.8201)	0.3530 (0.3438 – 0.3622)	0.5100 (0.5032 – 0.5168)
XGB	0.8671 (0.8635 – 0.8706)	0.4008 (0.3901 – 0.4116)	0.5683 (0.5577 – 0.5790)
MLP	0.8068 (0.7959 – 0.8176)	0.3057 (0.2892 – 0.3223)	0.4771 (0.4562 – 0.4979)

Table 16 delineates the performance metrics derived from the classifiers that employed latent embeddings from VAE trained on the original data. Concurrently, Table 17 outlines the results obtained from classifiers that utilized raw accounting data. For the dataset pertaining to Norway, there is a notable escalation in the AUC score for every classifier barring LR. A similar trend is observable for the H-measure and KS statistics. Furthermore, an increase in the performance of all classifiers is evident as the dimensionality of z escalates. It appears that the dimension reduction process employed by VAE leads to information loss, resulting in the learned latent representations exhibiting lesser predictive prowess compared to the raw data. The most efficacious model in this experimental setting proved to be the XGB model.

With Rebalancing: Random Under-Sampling

Table 18

Performance Measure – Norwegian Data Set Using VAE latent embeddings

Classifier	z	AUC	H-Measure	KS
LR	20	0.7539 (0.7467 – 0.7612)	0.1886 (0.1789 – 0.1982)	0.3915 (0.3825 – 0.4005)
LR	25	0.7564 (0.7495 – 0.7633)	0.1892 (0.1816 – 0.1969)	0.3982 (0.3937 – 0.4026)
LR	30	0.7557 (0.7399 – 0.7716)	0.1883 (0.1615 – 0.2152)	0.3992 (0.3694 – 0.4291)
RF	20	0.7899 (0.7817 – 0.7981)	0.2490 (0.2322 – 0.2658)	0.4451 (0.4282 – 0.4620)
RF	25	0.7901 (0.7798 – 0.8005)	0.2482 (0.2280 – 0.2684)	0.4469 (0.4255 – 0.4682)
RF	30	0.7913 (0.7792 – 0.8034)	0.2492 (0.2270 – 0.2715)	0.4485 (0.4270 – 0.4700)
XGB	20	0.7770 (0.7691 – 0.7850)	0.2218 (0.2068 – 0.2637)	0.4266 (0.4085 – 0.4448)

XGB	25	0.7789 (0.7643 – 0.7936)	0.2252 (0.2019 – 0.2486)	0.4308 (0.4025 – 0.4592)
XGB	30	0.7799 (0.7689 – 0.7909)	0.2231 (0.2035 – 0.2426)	0.4311 (0.4123 – 0.4500)
MLP	20	0.7837 (0.7759 – 0.7916)	0.2344 (0.2211 – 0.2476)	0.4459 (0.4295 – 0.4623)
MLP	25	0.7834 (0.7809 – 0.7859)	0.2330 (0.2258 – 0.2403)	0.4447 (0.4403 – 0.4490)
MLP	30	0.7873 (0.7749 – 0.7998)	0.2411 (0.2191 – 0.2630)	0.4513 (0.4290 – 0.4735)

Table 19

Performance Measure – Norwegian Data Set Using Balanced Train Set

Classifier	AUC	H-Measure	KS
LR	0.7576 (0.7519 – 0.7632)	0.1956 (0.1887 – 0.2025)	0.3970 (0.3898 – 0.4042)
RF	0.8580 (0.8565 – 0.8595)	0.3759 (0.3725 – 0.3793)	0.5555 (0.5505 – 0.5605)
XGB	0.8563 (0.8534 – 0.8592)	0.3707 (0.3616 – 0.3798)	0.5490 (0.5468 – 0.5511)
MLP	0.7848 (0.7704 – 0.7992)	0.2397 (0.2128 – 0.2666)	0.4569 (0.4326 – 0.4811)

Table 18 illustrates the performance indicators drawn from the classifiers utilizing latent embeddings from VAE trained on under-sampled data. In contrast, Table 19 demonstrates the outcomes from classifiers utilizing under-sampled data directly. A comparable pattern is discernible when using an under-sampled training set. The performance metrics indicate that the use of latent embeddings leads to diminished performance across all classifiers, excluding LR, where the performance utilizing latent embeddings surpasses that of using the under-sampled training set. Mirroring the usage of the original training set, there is an observed enhancement in the performance of the classifiers as the dimensionality of *z* increases. The most efficacious model in this experimental environment is the RF model. The application of random under-sampling to the training set results in a decrease in performance for both the XGB and MLP models in both experimental settings compared to using data without rebalancing.

With Rebalancing: SMOTE

Table 20

Performance Measure – Norwegian Data Set Using VAE latent embeddings

Classifier	z	AUC	H-measure	KS

LR	20	0.7710 (0.7641 – 0.7780)	0.2159 (0.2065 – 0.2252)	0.4274 (0.4156 – 0.4393)
LR	25	0.7677 (0.7606 – 0.7749)	0.2129 (0.2018 – 0.2240)	0.4188 (0.4062 – 0.4313)
LR	30	0.7732 (0.7627 – 0.7837)	0.2198 (0.2078 – 0.2318)	0.4294 (0.4116 – 0.4473)
RF	20	0.7862 (0.7797 – 0.7927)	0.2631 (0.2521 – 0.2742)	0.4378 (0.4248 – 0.4509)
RF	25	0.7863 (0.7817 – 0.7908)	0.2656 (0.2556 – 0.2755)	0.4423 (0.4346 – 0.4501)
RF	30	0.7885 (0.7847 – 0.7923)	0.2679 (0.2629 – 0.2729)	0.4433 (0.4323 – 0.4544)
XGB	20	0.7882 (0.7824 – 0.7940)	0.2525 (0.2401 – 0.2650)	0.4440 (0.4321 – 0.4559)
XGB	25	0.7871 (0.7801 – 0.7941)	0.2525 (0.2407 – 0.2644)	0.4467 (0.4319 – 0.4615)
XGB	30	0.7876 (0.7826 – 0.7926)	0.2563 (0.2499 – 0.2627)	0.4413 (0.4342 – 0.4484)
MLP	20	0.7806 (0.7692 – 0.7919)	0.2491 (0.2309 – 0.2674)	0.4332 (0.4141 – 0.4523)
MLP	25	0.7783 (0.7718 – 0.7848)	0.2487 (0.2399 – 0.2574)	0.4230 (0.4125 – 0.4335)
MLP	30	0.7783 (0.7735 – 0.7831)	0.2515 (0.2481 – 0.2548)	0.4214 (0.4118 – 0.4310)

Table 21

Performance Measure – Norwegian Data Set Using Balanced Train Set

Classifier	AUC	H-Measure	KS
LR	0.7542 (0.7485 – 0.7600)	0.1904 (0.1815 – 0.1992)	0.3889 (0.3765 – 0.4012)
RF	0.8381 (0.8360 – 0.8402)	0.3532 (0.3458 – 0.3605)	0.5314 (0.5252 – 0.5376)
XGB	0.8221 (0.8187 – 0.8255)	0.3054 (0.2977 – 0.3131)	0.4877 (0.4800 – 0.4955)
MLP	0.7801 (0.7717 – 0.7885)	0.2564 (0.2446 – 0.2681)	0.4296 (0.4157 – 0.4435)

Table 20 conveys the performance metrics derived from the classifiers using latent embeddings from VAE trained on over-sampled data. Concurrently, Table 21 details the outcomes from classifiers that have employed over-sampled data. The results mirror the outcomes observed when the training set is rebalanced using random under-sampling. Specifically, the performance of classifiers leveraging VAE latent embeddings is found to be inferior compared to those utilizing the rebalanced training set, with the exception of LR. Additionally, the performance of classifiers is noted to improve as the dimensionality of z increases, while the

performance of both XGB and MLP models declines when compared to the usage of data without rebalancing.

The latent space for these three experimental scenarios can be viewed in Appendices 7, 8, and 9.

Taiwanese Data Set

Without Rebalancing: Using Original Data

Table 22

Performance Measure – Taiwanese Data Set Using VAE latent embeddings

Classifier	z	AUC	H-Measure	KS
LR	30	0.9130 (0.8894 – 0.9366)	0.5847 (0.5080 – 0.6613)	0.7143 (0.6544 – 0.7741)
LR	45	0.9144 (0.8961 – 0.9327)	0.5947 (0.5274 – 0.6621)	0.7247 (0.6732 – 0.7761)
LR	60	0.9138 (0.8940 – 0.9335)	0.5942 (0.5230 – 0.6653)	0.7167 (0.6633 – 0.7700)
RF	30	0.9107 (0.8876 – 0.9338)	0.5883 (0.5082 – 0.6683)	0.7207 (0.6772 – 0.7642)
RF	45	0.9122 (0.8841 – 0.9404)	0.6112 (0.5237 – 0.6988)	0.7425 (0.6941 – 0.7909)
RF	60	0.9180 (0.900 – 0.9359)	0.5840 (0.5324 – 0.6357)	0.7332 (0.7089 – 0.7575)
XGB	30	0.8910 (0.8506 – 0.9313)	0.5478 (0.4483 – 0.6474)	0.6933 (0.6139 – 0.7727)
XGB	45	0.9185 (0.8972 – 0.9398)	0.5927 (0.5124 – 0.6731)	0.7288 (0.6805 – 0.7772)
XGB	60	0.9095 (0.8848 – 0.9342)	0.5627 (0.4979 – 0.6275)	0.6905 (0.6358 – 0.7453)
MLP	30	0.8525 (0.8038 – 0.9012)	0.4535 (0.3789 – 0.5281)	0.6043 (0.5226 – 0.6860)
MLP	45	0.8531 (0.8172 – 0.8891)	0.4428 (0.3600 – 0.5256)	0.6013 (0.5292 – 0.6735)
MLP	60	0.8370 (0.7932 – 0.8809)	0.4485 (0.3785 – 0.5184)	0.5728 (0.5034 – 0.6422)

Table 23

Performance Measure – Taiwanese Data Set Using Raw Train Set

Classifier	AUC	H-Measure	KS
LR	0.8974 (0.8791 – 0.9158)	0.5686 (0.5158 – 0.6214)	0.7078 (0.6628 – 0.7528)
RF	0.9324 (0.9129 – 0.9519)	0.6367 (0.5772 – 0.6962)	0.7471 (0.6937 – 0.8006)
XGB	0.9290 (0.9148 – 0.9431)	0.6559 (0.6158 – 0.6960)	0.7578 (0.7294 – 0.7836)

MLP	0.8507 (0.8139 – 0.8875)	0.4906 (0.4436 – 0.5376)	0.6338 (0.5954 – 0.6722)

Table 22 elucidates the performance metrics derived from the classifiers that used latent embeddings from VAE trained on original data, while Table 23 represents the outcomes from classifiers that leveraged raw accounting data for the Taiwanese dataset. As with the Norwegian dataset, it is observed that the learned latent representations *z* exhibit lower performance than when using the raw training set, with the exception of LR. Intriguingly, in this experiment, the performance of the classifiers improves as the dimensionality of *z* increases from 30 to 45, but it diminishes when the dimensionality of *z* further escalates from 45 to 60. The RF model emerged as the most effective model in this experimental context.

With Rebalancing: Random Under-Sampling

Table 24

Performance Measure – Taiwanese Data Set Using VAE latent embeddings

Classifier	z	AUC	H-Measure	KS
LR	30	0.9128 (0.8903 – 0.9354)	0.5698 (0.5047 – 0.6348)	0.7105 (0.6611 – 0.7600)
LR	45	0.9176 (0.8990 – 0.9362)	0.5757 (0.5106 – 0.6407)	0.7064 (0.6610 – 0.7518)
LR	60	0.9177 (0.8976 – 0.9378)	0.5745 (0.5030 – 0.6460)	0.7186 (0.6640 – 0.7732)
RF	30	0.9092 (0.8865 – 0.9319)	0.5647 (0.5102 – 0.6192)	0.7018 (0.6506 – 0.7530)
RF	45	0.9142 (0.8922 – 0.9362)	0.5667 (0.4991 – 0.6342)	0.7020 (0.6463 – 0.7577)
RF	60	0.9161 (0.8971 – 0.9352)	0.5844 (0.5249 – 0.6439)	0.7172 (0.6568 – 0.7776)
XGB	30	0.8989 (0.8737 – 0.9241)	0.5412 (0.4668 – 0.6155)	0.6972 (0.6243 – 0.7701)
XGB	45	0.9074 (0.8941 – 0.9207)	0.5430 (0.5099 – 0.5761)	0.6940 (0.6526 – 0.7354)
XGB	60	0.9071 (0.8912 – 0.9229)	0.5628 (0.5194 – 0.6062)	0.7016 (0.6568 – 0.7464)
MLP	30	0.8852 (0.8561 – 0.9142)	0.5099 (0.4419 – 0.5779)	0.6893 (0.6263 – 0.7523)
MLP	45	0.8895 (0.8671 – 0.9119)	0.5216 (0.4564 – 0.5858)	0.6967 (0.6737 – 0.7197)
MLP	60	0.8841 (0.8617 – 0.9064)	0.5017 (0.4469 – 0.5565)	0.6718 (0.6345 – 0.7091)

Table 25

Performance Measure – Taiwanese Data Set Using Balanced Train Set

Classifier	AUC	H-measure	KS
LR	0.8937 (0.8645 – 0.9230)	0.5358 (0.4639 – 0.6077)	0.6741 (0.6044 – 0.7438)
RF	0.9343 (0.9236 – 0.9451)	0.6326 (0.6064 – 0.6588)	0.7558 (0.7359 – 0.7756)
XGB	0.9225 (0.9148 – 0.9301)	0.5937 (0.5646 – 0.6227)	0.7291 (0.7050 – 0.7532)
MLP	0.8990 (0.8773 – 0.9207)	0.5463 (0.4842 – 0.6084)	0.7055 (0.6491 – 0.7618)

The implementation of random under-sampling to the training set resulted in an enhancement in the performance of all classifiers, both when using latent embeddings from VAE and when using the balanced training set, with the exception of XGB, where the performance remained relatively static (Table 24 and Table 25). Contrary to previous experiments, the H-measure and KS statistics exhibited a slight increase for RF and XGB when the dimensionality of the latent representation z was augmented from 45 to 60. However, the AUC score continued to decline.

With Rebalancing: SMOTE

Table 26

Performance Measure – Taiwanese Data Set Using VAE latent embeddings

Classifier	z	AUC	H-Measure	KS
LR	30	0.8946 (0.8758 – 0.9134)	0.5612 (0.5093 – 0.6130)	0.6837 (0.6397 – 0.7276)
LR	45	0.8843 (0.8533 – 0.9153)	0.5549 (0.4700 – 0.6398)	0.6809 (0.6123 – 0.7495)
LR	60	0.8995 (0.8785 – 0.9206)	0.5542 (0.5037 – 0.6048)	0.6828 (0.6375 – 0.7281)
RF	30	0.8996 (0.8666 – 0.9307)	0.5501 (0.4848 – 0.6154)	0.6838 (0.6182 – 0.7494)
RF	45	0.8979 (0.8631 – 0.9327)	0.5670 (0.4762 – 0.6414)	0.6775 (0.6251 – 0.7298)
RF	60	0.9071 (0.8806 – 0.9336)	0.5670 (0.4960 – 0.6379)	0.6913 (0.6361 – 0.7466)
XGB	30	0.8975 (0.8797 – 0.9153)	0.5400 (0.4917 – 0.5883)	0.6619 (0.6026 – 0.7212)
XGB	45	0.9027 (0.8757 – 0.9298)	0.5497 (0.4692 – 0.6301)	0.6854 (0.6319 – 0.7389)
XGB	60	0.9034 (0.8714 – 0.9354)	0.5815 (0.4934 – 0.6697)	0.6949 (0.6116 – 0.7781)
MLP	30	0.8638 (0.8478 – 0.8797)	0.4940 (0.4605 – 0.5274)	0.6520 (0.6152 – 0.6889)

MLP	45	0.8503 (0.8248 – 0.8758)	0.4648 (0.4006 – 0.5290)	0.5980 (0.5357 – 0.6602)
MLP	60	0.8439 (0.8131 – 0.8747)	0.4631 (0.3990 – 0.5272)	0.5996 (0.5326 – 0.6667)

Table 27

Performance Measure – Taiwanese Data Set Using Balanced Train Set

Classifier	AUC	H-Measure	KS
LR	0.8824 (0.8512 – 0.9136)	0.5499 (0.4895 – 0.6103)	0.6721 (0.6214 – 0.7228)
RF	0.9317 (0.9172 – 0.9462)	0.6242 (0.5736 – 0.6749)	0.7421 (0.7165 – 0.7677)
XGB	0.9222 (0.9042 – 0.9402)	0.6063 (0.5580 – 0.6547)	0.7298 (0.7004 – 0.7593)
MLP	0.8328 (0.7921 – 0.8736)	0.4771 (0.4429 – 0.5112)	0.6170 (0.5697 – 0.6643)

In the concluding experiment conducted on the Taiwanese dataset, we observe a pattern analogous to previous experiments, wherein the performance of classifiers employing VAE latent embeddings is found to be inferior compared to those directly utilizing the balanced training set, with the exceptions of LR and MLP (Table 26 and Table 27). In this particular experiment, the performance of LR, RF, and XGB improved with an increase in the dimensionality of the latent representation *z*, while the performance of MLP declined as the dimensionality of *z* increased. The best model in this experiment is RF.

The visual representation of the learned latent space can be examined in Appendices 10, 11, and 12.

Polish Data Set

Without Rebalancing: Using Original Data

Table 28

Performance Measure – Polish Data Set Using VAE latent embeddings

Classifier	z	AUC	H-Measure	KS
LR	20	0.6768 (0.6705 – 0.6830)	0.1131 (0.1014 – 0.1247)	0.2650 (0.2548 – 0.2753)
LR	30	0.6960 (0.6798 – 0.7122)	0.1364 (0.1151 – 0.1576)	0.3062 (0.2776 – 0.3348)
LR	40	0.6865 (0.6773 – 0.6957)	0.1261 (0.1160 – 0.1361)	0.2930 (0.2830 – 0.3031)
RF	20	0.7130 (0.6974 – 0.7287)	0.1446 (0.1272 – 0.1875)	0.3525 (0.2941 – 0.3563)

RF	30	0.7277 (0.7097 – 0.7457)	0.1629 (0.1384 – 0.1875)	0.3475 (0.3102 – 0.3849)
RF	40	0.7234 (0.6988 – 0.7480)	0.1601 (0.1307 – 0.1895)	0.3429 (0.2949 – 0.3910)
XGB	20	0.7053 (0.6798 – 0.7308)	0.1381 (0.1076 – 0.1686)	0.3146 (0.2714 – 0.3578)
XGB	30	0.7281 (0.7138 – 0.7423)	0.1681 (0.1513 – 0.1849)	0.3553 (0.3361 – 0.3746)
XGB	40	0.7281 (0.7059 – 0.7504)	0.1675 (0.1410 – 0.1940)	0.3506 (0.3127 – 0.3884)
MLP	20	0.7105 (0.6947 – 0.7264)	0.1472 (0.1347 – 0.1597)	0.3321 (0.3100 – 0.3543)
MLP	30	0.7173 (0.6971 – 0.7374)	0.1621 (0.1431 – 0.1812)	0.3427 (0.3064 – 0.3789)
MLP	40	0.7121 (0.700 – 0.7243)	0.1546 (0.1405 – 0.1688)	0.3267 (0.2962 – 0.3572)

Table 29

Performance Measure – Polish Data Set Using Raw Train Set

Classifier	AUC	H-Measure	KS
LR	0.6472 (0.6564 – 0.6920)	0.1154 (0.0895 – 0.1412)	0.2584 (0.2316 – 0.2851)
RF	0.9092 (0.9000 – 0.9183)	0.5512 (0.5330 – 0.5695)	0.6638 (0.6434 – 0.6842)
XGB	0.9761 (0.9735 – 0.9787)	0.7836 (0.7755 – 0.7918)	0.8398 (0.8316 – 0.8480)
MLP	0.8287 (0.8182 – 0.8391)	0.3720 (0.3634 – 0.3806)	0.5350 (0.5240 – 0.5461)

Tables 28 and 29 reveal that the XGB classifier, when employing the raw training set, surpasses the performance of all other classifiers. As for the dimensionality of the latent representation *z*, it is observed that the performance of all classifiers escalates when the dimension of *z* increases from 20 to 30, but it experiences a slight decrease when *z* further escalates from 30 to 40. In terms of the performance of the VAE latent embeddings, it is noted that their application yields significantly lower performance than when the raw training set is used for training the classifiers. However, in the case of LR, there is an enhancement in performance.

With Rebalancing: Random Under-Sampling

Table 30

Performance Measure – Polish Data Set Using VAE latent embeddings

Classifier	z	AUC	H-Measure	KS

LR	20	0.6691 (0.6819 – 0.7162)	0.1320 (0.1110 – 0.1530)	0.3100 (0.2883 – 0.3316)
LR	30	0.6997 (0.6877 – 0.7117)	0.1307 (0.1202 – 0.1411)	0.3057 (0.2906 – 0.3208)
LR	40	0.7106 (0.6978 – 0.7235)	0.1453 (0.1322 – 0.1585)	0.331 (0.3199 – 0.3462)
RF	20	0.7293 (0.7106 – 0.7480)	0.1626 (0.1392 – 0.1859)	0.3513 (0.3206 – 0.3819)
RF	30	0.7323 (0.7176 – 0.7470)	0.1709 (0.1503 – 0.1916)	0.3520 (0.3145 – 0.3895)
RF	40	0.7369 (0.7246 – 0.7492)	0.1733 (0.1560 – 0.1907)	0.3699 (0.3389 – 0.4008)
XGB	20	0.7252 (0.7055 – 0.7448)	0.1529 (0.1321 – 0.1863)	0.3467 (0.3101 – 0.3832)
XGB	30	0.7321 (0.7199 – 0.7444)	0.1668 (0.1484 – 0.1853)	0.3527 (0.3290 – 0.3764)
XGB	40	0.7326 (0.7157 – 0.7494)	0.1661 (0.1441 – 0.1882)	0.3576 (0.3241 – 0.3911)
MLP	20	0.7318 (0.7149 – 0.7486)	0.1709 (0.1460 – 0.1959)	0.3633 (0.3305 – 0.3961)
MLP	30	0.7205 (0.7058 – 0.7353)	0.1589 (0.1359 – 0.1819)	0.3531 (0.3286 – 0.3776)
MLP	40	0.7186 (0.7005 – 0.7367)	0.1525 (0.1297 – 0.1752)	0.3531 (0.3291 – 0.3770)

Table 31

Performance Measure – Polish Data Set Using Balanced Train Set

Classifier	AUC	H-Measure	KS
LR	0.7451 (0.7321 – 0.7581)	0.1927 (0.1776 – 0.2077)	0.3814 (0.3669 – 0.3960)
RF	0.8890 (0.8729 – 0.9051)	0.4693 (0.4258 – 0.5129)	0.6220 (0.5945 – 0.6495)
XGB	0.9592 (0.9558 – 0.9625)	0.6915 (0.6731 – 0.7099)	0.7794 (0.7661 – 0.7927)
MLP	0.8149 (0.8035 – 0.8262)	0.3111 (0.2955 – 0.3266)	0.5127 (0.4944 – 0.5310)

In this experimental setting, it is rather surprising to find that the performance of LR utilizing VAE latent embeddings is inferior compared to its performance when using a balanced training set (Table 30 and Table 31). The application of random under-sampling to the training set leads to a decline in the performance of all classifiers when compared to the previous experiment, with the sole exception of LR. While there is an observed improvement in prediction accuracy when VAE latent embeddings are employed to train the classifiers

relative to the previous experiment, the overall performance remains significantly lower than when the raw training set is used.

With Rebalancing: SMOTE

Table 32

Performance Measure – Polish Data Set Using VAE latent embeddings

Classifier	z	AUC	H-measure	KS
LR	20	0.7071 (0.6877 – 0.7264)	0.1459 (0.1292 – 0.1626)	0.3181 (0.2878 – 0.3484)
LR	30	0.7132 (0.6889 – 0.7376)	0.1555 (0.1329 – 0.1781)	0.3323 (0.3023 – 0.3623)
LR	40	0.7164 (0.6964 – 0.7364)	0.1604 (0.1381 – 0.1826)	0.3416 (0.3094 – 0.3738)
RF	20	0.7249 (0.7078 – 0.7411)	0.1640 (0.1475 – 0.1805)	0.3459 (0.3177 – 0.3741)
RF	30	0.7261 (0.7144 – 0.7378)	0.1661 (0.1523 – 0.1798)	0.3465 (0.3285 – 0.3645)
RF	40	0.7295 (0.7157 – 0.7433)	0.1694 (0.1504 – 0.1883)	0.3559 (0.3346 – 0.3772)
XGB	20	0.7045 (0.6924 – 0.7167)	0.1413 (0.1246 – 0.1581)	0.3180 (0.3053 – 0.3307)
XGB	30	0.7089 (0.7051 – 0.7126)	0.1433 (0.1384 – 0.1482)	0.3191 (0.3037 – 0.3345)
XGB	40	0.6965 (0.6813 – 0.7116)	0.1317 (0.1169 – 0.1464)	0.3044 (0.2980 – 0.3108)
MLP	20	0.6926 (0.6808 – 0.7042)	0.1362 (0.1255 – 0.1469)	0.3109 (0.2956 – 0.3262)
MLP	30	0.7039 (0.6939 – 0.7139)	0.1496 (0.1376 – 0.1616)	0.3256 (0.2967 – 0.3545)
MLP	40	0.6939 (0.6858 – 0.7013)	0.1427 (0.1319 – 0.1535)	0.3089 (0.3031 – 0.3146)

Table 33

Performance Measure – Polish Data Set Using Balanced Train Set

Classifier	AUC	H-Measure	KS
LR	0.7372 (0.7219 – 0.7526)	0.1923 (0.1742 – 0.2104)	0.3750 (0.3589 – 0.3911)
RF	0.8684 (0.8572 – 0.8795)	0.4105 (0.3794 – 0.4417)	0.5899 (0.5756 – 0.6042)
XGB	0.9681 (0.9637 – 0.9724)	0.7528 (0.7406 – 0.7650)	0.8143 (0.8057 – 0.8229)
MLP	0.8473 (0.8310 – 0.8636)	0.4345 (0.4113 – 0.4576)	0.5742 (0.5513 – 0.5971)

In the final experiment, the outcomes bear similarity to those of the random under-sampling experiment, as exhibited in Tables 32 and 33. The sole

distinguishing aspect in this instance is that the Synthetic Minority Over-sampling Technique (SMOTE) bolsters the performance of the Multilayer Perceptron (MLP) classifier when trained with the balanced training set. However, the performance of the MLP classifier trained with Variational Autoencoder (VAE) latent embeddings is noted to be inferior to that of the previous experiments. Additionally, an increase in performance across classifiers is observed when the dimensionality of the latent representation *z* rises from 20 to 30, with a slight decrease noted when the *z* dimension further increases from 30 to 40. The best classifier in this experiment is XGB for using balanced train set and RF for using VAE latent embeddings.

Visualizations of the VAE latent embeddings can be accessed in Appendices 13, 14, and 15.

Summarizing results

Throughout the course of this investigation, it was consistently observed that classifiers' performance, when utilizing VAE latent embeddings, was generally inferior to those leveraging the original or balanced training sets directly. This held true with the singular exception of LR in specific experimental contexts. This pattern implies that the dimensionality reduction process undertaken by the VAE might induce some degree of predictive power loss. One hypothesis to explain the improved performance of LR when using VAE latent embeddings is that VAE are capable of capturing intricate, high-dimensional relationships within a compressed, lower-dimensional space. Consequently, these models can effectively filter noise and identify non-linear relationships in the data, a capability not inherent to LR. Furthermore, the process of learning a lower-dimensional representation may introduce a regularization effect, which can aid in the prevention of overfitting. This effect could potentially enhance the generalizability of the LR model, yielding improved performance on unseen data.

The performance of classifiers was observed to escalate as the dimensionality of the latent space increased, albeit only up to a certain threshold. Beyond this point, a decrement in performance was noted, implying the existence of an optimal dimensionality *z* for these data sets. A hypothesis to explain the existence of an optimal dimensionality for *z*, the latent space of the VAE, lies in the delicate equilibrium between information preservation and reduction of noise or data redundancy. As the dimensionality of *z* escalates, the VAE is granted enhanced capacity to encode intricate, high-dimensional data. This increased complexity, to

a certain extent, can boost the model's performance by enabling it to perceive more subtle correlations within the data. However, beyond this optimal threshold, any further augmentation in dimensionality may lead to the introduction of unnecessary complexity that begins to undermine performance. This is because the VAE may start to overfit the data noise, capturing random fluctuations that do not generalize well to unseen data. Essentially, the model may learn to represent specificities intrinsic to the training data that do not bear relevance to the overall data-generating process. Therefore, there is an optimal dimensionality of z at which the model optimally balances the trade-off between capturing salient data patterns and avoiding overfitting to noise.

When under-sampling or over-sampling techniques were applied to the training set, an overall decline in classifier performance was noted, particularly for XGB and MLP. However, LR emerged as an exception in certain contexts, exhibiting improved performance when employing latent embeddings.

With respect to the Taiwanese dataset, the RF classifier consistently proved to be the most effective model across various experimental configurations. On the contrary, for the Norwegian and Polish datasets, the XGB classifier often demonstrated superior performance relative to other classifiers when utilizing the raw training set.

Discussion

Limitations

The focus of this thesis is on the application of Variational Autoencoder (VAE) latent embeddings to train classifiers for bankruptcy prediction. While this focus has offered novel insights, it has also necessitated certain constraints which represent the limitations of this study. First, the computational resources available for this research were insufficient to facilitate the exhaustive exploration of all potential parameters of the VAE. As a result, the study could not fully explore the impact of different VAE configurations on the classifiers' performance.

The data preprocessing procedure employed in this study was basic and consisted mainly of replacing missing values with zeroes and removing duplicate entries. More sophisticated preprocessing methodologies, such as the removal of low variance features or the imputation of missing values with mean values, were not utilized. The impact of these alternative preprocessing techniques on classifier performance remains an open question.

In this study, the techniques of Synthetic Minority Over-sampling Technique or Random Under-Sampling were used to address the issue of class imbalances within the data. However, these methods may not always be the most effective approach for all datasets or classification tasks. Other techniques, such as SMOTETomek, SMOTEENN, or the generation of synthetic data using VAEs, may offer superior results but were not considered in this study.

Lastly, our findings suggest the existence of an optimal dimensionality for the latent space within these datasets. However, the changes in classifier performance associated with increasing dimensionality were minimal, and the standard deviation of the metrics was relatively large in comparison to these changes. This observation points to a complex, non-linear relationship between dimensionality and classifier performance, which was not thoroughly explored in this thesis.

Further research

Building upon the findings of this study, a plethora of promising avenues for further research are apparent. The task of optimizing the use of VAE in business risk assessment is particularly pertinent when considering the use of accounting data. One crucial aspect that came to the fore is the significant impact of VAE

parameters, notably the dimensionality of the latent space, on classifier performance. This suggests that future research could benefit from allocating more extensive computational resources for a more comprehensive exploration of potential parameters. This would enable researchers to identify optimal settings for various types of accounting data.

Our preprocessing strategy was relatively simple, involving the replacement of missing values with zeros and the removal of duplications. However, the adoption of more sophisticated preprocessing techniques, such as advanced imputation of missing values or the elimination of low-variance features, could potentially enhance the performance of classifiers trained with VAE latent embeddings. This warrants further investigation.

The study also demonstrated that the application of under-sampling or over-sampling techniques to the training set often led to decreased classifier performance. This finding invites the exploration of alternative methods for handling class imbalances, including advanced techniques like SMOTETomek, SMOTEENN, or the generation of synthetic data using VAEs.

The concept of an optimal dimensionality z for the latent space emerged from our findings, with classifier performance decreasing beyond a certain threshold. This indicates a need for further research to pinpoint this optimal dimensionality for different types of data and classifiers.

Moreover, the unique performance of logistic regression in certain contexts, particularly its improved performance with the use of latent embeddings, calls for an investigation into why this occurs and whether it extends to other types of classifiers.

Finally, the geographical focus of this study was on the Norwegian context. However, examining other geographic or industry contexts could yield valuable insights into the consistency of these findings across different business environments. By investigating these areas, future research can continue to refine our understanding of how VAEs can be harnessed to enhance classifier performance in business risk assessment.

Conclusion

Our primary research question revolved around the evaluation of VAE latent embeddings in training classifiers for bankruptcy prediction. The exploration was carried out using three distinct datasets and the performance of classifiers was compared when trained on VAE latent embeddings versus the original or balanced training sets.

Our key findings revealed that the performance of classifiers using VAE latent embeddings was generally not as robust as those trained directly on the original or balanced datasets. However, exceptions were observed, especially with Logistic Regression, which demonstrated improved performance in certain experimental settings. We observed distinct patterns of classifier performance across datasets, with the Random Forest classifier performing optimally on the Taiwanese dataset, while the Extreme Gradient Boosting classifier demonstrated superior performance on the Norwegian and Polish datasets when using the raw training set. An intriguing finding was the impact of the dimensionality of latent representations z on classifier performance, with an increase in the dimensionality of z enhancing performance up to a point, beyond which a decrease was observed. This suggests the existence of an optimal dimensionality for these datasets.

The implications of class imbalances were also examined. The use of under-sampling or over-sampling techniques generally led to a decrease in classifier performance, particularly in the case of Extreme Gradient Boosting and Multi-layer Perceptron. Again, Logistic Regression emerged as an exception, showing improved performance with VAE latent embeddings in certain settings.

While our research provides substantial insights, it also acknowledges inherent limitations, particularly in relation to VAE parameter exploration and data preprocessing techniques due to computational constraints.

These findings have significant implications for future research. Further studies could delve into a more comprehensive exploration of VAE parameters, more sophisticated data preprocessing techniques, and alternative methods to handle class imbalances. The observed performance of classifiers trained on VAE latent embeddings also opens up promising avenues for their application in bankruptcy prediction. In conclusion, the research has not only contributed valuable insights to the field of bankruptcy prediction using VAE latent embeddings but has also paved the way for future research, thereby highlighting the potential for further advancements in this area.

References

Agarwal, V., & Taffler, R. (2008). Comparing the performance of market-based and accounting-based bankruptcy prediction models. *Journal of Banking & Finance*, *32*(8), 1541–1551. https://doi.org/10.1016/j.jbankfin.2007.07.014

Altinn. (2021). *Bankruptcy of private limited companies*. https://www.altinn.no/en/start-and-run-business/deregistration-closure-bankruptcyliquidation/bankruptcy-liquidation/bankruptcy-of-private-limited-companies/

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *The Journal of Finance*, *23*(4), 589–609. https://doi.org/10.1111/j.1540-6261.1968.tb00843.x

An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, *2*, 1.

Atiya, A. F. (2001). Bankruptcy prediction for credit risk using neural networks: A survey and new results. *IEEE Transactions on Neural Networks*, *12*(4), 929–935. https://doi.org/10.1109/72.935101

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J. a. K., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, *54*(6), 627–635. https://doi.org/10.1057/palgrave.jors.2601545

Balcaen, S., & Ooghe, H. (2006). 35 years of studies on business failure: an overview of the classic statistical methodologies and their related problems. *British Accounting Review*, *38*(1), 63–93. https://doi.org/10.1016/j.bar.2005.09.001

Barboza, F., Kimura, H., & Altman, E. I. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications*, *83*, 405–417. https://doi.org/10.1016/j.eswa.2017.04.006

Barniv, R., Agarwal, A., & Leach, R. L. (2002). Predicting Bankruptcy Resolution. Journal of Business Finance & Accounting, 29(3 & 4), 497–520. https://doi.org/10.1111/1468-5957.00440

Beaver, W. H. (1966). Financial Ratios as Predictors of Failure. *Journal of Accounting Research*, *4*, 71. https://doi.org/10.2307/2490171

Begley, J., Ming, J., & Watts, S. (1996). Bankruptcy classification errors in the 1980s: An empirical analysis of Altman's and Ohlson's models. *Review of Accounting Studies*, *1*(4), 267–284. https://doi.org/10.1007/bf00570833

Bell, T. B., Ribar, G. S., & Verichio, J. (1990). Neural nets versus logistic regression: A comparison of each model's ability to predict commercial bank failures. *Proceedings of the University of Kansas Symposium on Auditing Problems*. https://egrove.olemiss.edu/cgi/viewcontent.cgi?article=1081&context=dl_proceedings

Bernhardsen, E. (2001). *Working Paper: A Model of Bankruptcy Prediction*. Norges Bank. Retrieved January 10, 2023, from https://www.norges-bank.no/globalassets/upload/import/publikasjoner/arbeidsnotater/pdf/arb-2001-10.pdf?v=03/09/2017122305

Bernhardsen, E., & Larsen, K. (2007). Modellering av kredittrisiko i foretakssektoren - Videreutvikling av SEBRA-modellen. *Penger Og Kreditt*. https://brage.bibsys.no/xmlui/handle/11250/2502276

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*(7), 1145–1159. https://doi.org/10.1016/s0031-3203(96)00142-2

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. https://doi.org/10.1007/bf00058655

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). Classification And Regression Trees. In *Routledge eBooks*. https://doi.org/10.1201/9781315139470

Buja, A., Stuetzle, W., & Shen, Y. (2005). Loss Functions for Binary Class Probability Estimation and Classification: Structure and Applications. *Computer Science*. http://stat.wharton.upenn.edu/~buja/PAPERS/paper-proper-scoring.pdf

Carmona, P., Climent, F., & Momparler, A. (2019). Predicting failure in the U.S. banking sector: An extreme gradient boosting approach. *International Review of Economics & Finance*, *61*, 304–323. https://doi.org/10.1016/j.iref.2018.03.008

Chawla, N. V., Bowyer, K. W., Hall, L. J., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Chen, M. Y. (2011). Predicting corporate financial distress based on integration of decision tree classification and logistic regression. *Expert Systems with Applications*, *38*(9), 11261–11272. https://doi.org/10.1016/j.eswa.2011.02.173

Chen, T., & Guestrin, C. (2016). *XGBoost*. https://doi.org/10.1145/2939672.2939785

Chudson, W. A. (1945). The Pattern of Corporate Financial Structure. *Journal of the Royal Statistical Society*, *108*(3/4), 472. https://doi.org/10.2307/2981301

Chung, H., & Tam, K. Y. (1993). A Comparative Analysis of Inductive-Learning Algorithms. *International Journal of Intelligent Systems in Accounting, Finance & Management*, *2*(1), 3–18. https://doi.org/10.1002/j.1099-1174.1993.tb00031.x

Coats, P. K., & Fant, L. F. (1993). Recognizing Financial Distress Patterns Using a Neural Network Tool. *Financial Management*, *22*(3), 142. https://doi.org/10.2307/3665934

Connor, M., Canal, G., & Rozell, C. J. (2021). Variational Autoencoder with Learned Latent Structure. In *International Conference on Artificial Intelligence and Statistics* (pp. 2359–2367). http://proceedings.mlr.press/v130/connor21a/connor21a.pdf

Cozzatti, M., Simonetta, F., & Ntalampiras, S. (2022). Variational Autoencoders for Anomaly Detection in Respiratory Sounds. *Lecture Notes in Computer Science*, 333–345. https://doi.org/10.1007/978-3-031-15937-4_28

Dasarathy, B. V., & Sheela, B. V. (1979). A composite classifier system design: Concepts and methodology. *Proceedings of the IEEE*, *67*(5), 708–713. https://doi.org/10.1109/proc.1979.11321

de Andrés, J., Landajo, M., & Lorca, P. (2012). Bankruptcy prediction models based on multinorm analysis: An alternative to accounting ratios. *Knowledge Based Systems*, *30*, 67–77. https://doi.org/10.1016/j.knosys.2011.11.005

Diakomihalis, M. N. (2012). The Accuracy of Altman's Models in Predicting Hotel Bankruptcy. *International Journal of Accounting and Financial Reporting*, *2*(2), 96. https://doi.org/10.5296/ijafr.v2i2.2367

Dittman, D. J., Khoshgoftaar, T. M., Wald, R., & Napolitano, A. (2014). Comparison of Data Sampling Approaches for Imbalanced Bioinformatics Data. In *The Florida AI Research Society*. https://dblp.uni-trier.de/db/conf/flairs/flairs2014.html#DittmanKWN14

Doumpos, M., Andriosopoulos, K., Galariotis, E. C., Makridou, G., & Zopounidis, C. (2017). Corporate failure prediction in the European energy sector: A multicriteria approach and the effect of country characteristics. *European Journal of Operational Research*, *262*(1), 347–360. https://doi.org/10.1016/j.ejor.2017.04.024

du Jardin, P. (2010). Predicting bankruptcy using neural networks and other classification methods: The influence of variable selection techniques on model accuracy. *Neurocomputing*, *73*(10–12), 2047–2060. https://doi.org/10.1016/j.neucom.2009.11.034

du Jardin, P. (2016). A two-stage classification technique for bankruptcy prediction. *European Journal of Operational Research*, *254*(1), 236–252. https://doi.org/10.1016/j.ejor.2016.03.008

Elrahman, S. M. A. (2014). *A Review of Class Imbalance Problem*. https://www.semanticscholar.org/paper/A-Review-of-Class-Imbalance-Problem-Elrahman-Abraham/bb2e442b2acb4530aa28d24e45578f84447d0425

Escalona-Morán, M., Soriano, M. C., Fischer, I., & Mirasso, C. R. (2015). Electrocardiogram Classification Using Reservoir Computing with Logistic

Regression. *IEEE Journal of Biomedical and Health Informatics*, *19*(3), 892–898. https://doi.org/10.1109/jbhi.2014.2332001

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems. *Journal of Machine Learning Research*. https://doi.org/10.5555/2627435.2697065

Fitzpatrick, F. (1932). A Comparison of Ratios of Successful Industrial Enterprises with Those of Failed Firm. *Certified Public Accountant*, *6*, 727–731.

Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139. https://doi.org/10.1006/jcss.1997.1504

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5). https://doi.org/10.1214/aos/1013203451

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202. https://doi.org/10.1007/bf00344251

Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M., Douglas, R., & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, *405*(6789), 947–951. https://doi.org/10.1038/35016072

Hand, D. J. (2009). Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning*, *77*(1), 103–123. https://doi.org/10.1007/s10994-009-5119-5

Hansen, L. D., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(10), 993–1001. https://doi.org/10.1109/34.58871

Hasanin, T., & Khoshgoftaar, T. M. (2018). *The Effects of Random Undersampling with Simulated Class Imbalance for Big Data*. https://doi.org/10.1109/iri.2018.00018

Hasanin, T., Khoshgoftaar, T. M., Leevy, J. L., & Seliya, N. (2019). *Investigating Random Undersampling and Feature Selection on Bioinformatics Big Data*. https://doi.org/10.1109/bigdataservice.2019.00063

Hillegeist, S. A., Keating, E. K., Cram, D. P., & Lundstedt, K. G. (2004). Assessing the Probability of Bankruptcy. *Review of Accounting Studies*, *9*(1), 5–34. https://doi.org/10.1023/b:rast.0000013627.90884.b7

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, *313*(5786), 504–507. https://doi.org/10.1126/science.1127647

Hinton, G. E., & Van Camp, D. (1993). Keeping Neural Networks Simple. In *Springer eBooks* (pp. 11–18). https://doi.org/10.1007/978-1-4471-2063-6_2

Hinton, G. E., & Zemel, R. S. (1993). Autoencoders, Minimum Description Length and Helmholtz Free Energy. *Neural Information Processing Systems*, *6*, 3–10. https://papers.nips.cc/paper/1993/file/9e3cfc48eccf81a0d57663e129aef3cb-Paper.pdf

Hjelseth, I. N., & Raknerud, A. A. (2016). *A model of credit risk in the corporate sector based on bankruptcy prediction*. Norges Bank. Retrieved January 10, 2023, from https://www.norges-

bank.no/contentassets/3da7332610b74bdeacfd208e1a1a76f2/staff_memo_20_2016.pdf?v=03/09/2017123537

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844. https://doi.org/10.1109/34.709601

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554–2558. https://doi.org/10.1073/pnas.79.8.2554

Horrigan, J. (1968). A Short History of Financial Ratio Analysis. *The Accounting Review*, *43*(2), 284–294. https://www.jstor.org/stable/243765

Hosaka, T. (2019). Bankruptcy prediction using imaged financial ratios and convolutional neural networks. *Expert Systems with Applications*, *117*, 287–299. https://doi.org/10.1016/j.eswa.2018.09.039

Jackendoff, N. (1962). *A Study of Published Industry Financial and Operating Ratios*. Temple University, Bureau of Economic and Business Research.

Joshi, S., Ramesh, R., & Tahsildar, S. (2018). *A Bankruptcy Prediction Model Using Random Forest*. https://doi.org/10.1109/iccons.2018.8663128

Karels, G. V., & Prakash, A. (1987). Multivariate Normality and Forecasting of Business Bankruptcy. *Journal of Business Finance & Accounting*, *14*(4), 573–593. https://doi.org/10.1111/j.1468-5957.1987.tb00113.x

Kim, K., & Ahn, H. (2012). A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach. *Computers & Operations Research*, *39*(8), 1800–1811. https://doi.org/10.1016/j.cor.2011.06.023

Kim, M., & Kang, D. (2010). Ensemble with neural networks for bankruptcy prediction. *Expert Systems with Applications*, *37*(4), 3373–3379. https://doi.org/10.1016/j.eswa.2009.10.012

Kingma, D. P. (2014, December 22). *Adam: A Method for Stochastic Optimization*. arXiv.org. https://arxiv.org/abs/1412.6980

Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv: Machine Learning*. http://export.arxiv.org/pdf/1312.6114

Kirkos, E. (2012). Assessing methodologies for intelligent bankruptcy prediction. *Artificial Intelligence Review*, *43*(1), 83–123. https://doi.org/10.1007/s10462-012-9367-6

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

LeCun, Y., & Fogelman-Soulié, F. (1987). Modèles connexionnistes de l'apprentissage. *Intellectica*, *2*(1), 114–143. https://doi.org/10.3406/intel.1987.1804

Lee, K., Booth, D. A., & Alam, P. (2005). A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms. *Expert Systems with Applications*, *29*(1), 1–16. https://doi.org/10.1016/j.eswa.2005.01.004

Liou, C. Y., Cheng, W. C., Liou, J. W., & Liou, D. R. (2014). Autoencoder for words. *Neurocomputing*, *139*, 84–96. https://doi.org/10.1016/j.neucom.2013.09.055

Liu, T. (2018, February 27). *A Kolmogorov-Smirnov type test for two inter-dependent random variables*. arXiv.org. https://arxiv.org/abs/1802.09899

Mai, F., Tian, S., Lee, C., & Ma, L. (2019). Deep learning models for bankruptcy prediction using textual disclosures. *European Journal of Operational Research*, *274*(2), 743–758. https://doi.org/10.1016/j.ejor.2018.10.024

Makhzani, A., & Frey, B. J. (2013). k-Sparse Autoencoders. *arXiv: Learning*. https://arxiv.org/pdf/1312.5663v2

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. J. (2015). Adversarial Autoencoders. *arXiv: Learning*. https://arxiv.org/pdf/1511.05644

Mancisidor, R. A., Kampffmeyer, M., Aas, K., & Jenssen, R. (2018). Segment-Based Credit Scoring Using Latent Clusters in the Variational Autoencoder. *arXiv: Computational Engineering, Finance, and Science*. https://arxiv.org/pdf/1806.02538.pdf

Mancisidor, R. A., Kampffmeyer, M., Aas, K., & Jenssen, R. (2021). Learning latent representations of bank customers with the Variational Autoencoder. *Expert Systems with Applications*, *164*, 114020. https://doi.org/10.1016/j.eswa.2020.114020

Marqués, A. I., García, V. D., & Sánchez, J. (2012). Two-level classifier ensembles for credit risk assessment. *Expert Systems with Applications*, *39*(12), 10916–10922. https://doi.org/10.1016/j.eswa.2012.03.033

McCulloch, W. S., & Pitts, W. (1943, December 1). *A logical calculus of the ideas immanent in nervous activity*. SpringerLink. https://link.springer.com/article/10.1007/BF02478259?error=cookies_not_supported&code=04ae0b85-41b9-4fcd-a9d2-298138b69d54

McKee, T. B., & Lensberg, T. (2002). Genetic programming and rough sets: A hybrid approach to bankruptcy classification. *European Journal of*

Operational Research, *138*(2), 436–451. https://doi.org/10.1016/s0377-2217(01)00130-8

Minsky, M., & Papert, S. (1969). *Perceptron's – An Introduction to Computational Geometry*. Massachusetts Institute of Technology. https://leon.bottou.org/publications/pdf/perceptrons-2017.pdf

Muzır, E., & Çağlar, N. (2009). The Accuracy of Financial Distress Prediction Models In Turkey: A Comparative Investigation With Simple Model Proposals. *Anadolu University Journal of Social Sciences*, *9*(2), 15–48. http://earsiv.anadolu.edu.tr/handle/11421/277

Nordea. (2021, July 15). *Basel IV is coming: What you need to know*. Retrieved January 16, 2023, from https://www.nordea.com/en/news/basel-iv-is-coming-what-you-need-to-know

Öcal, N., Ercan, M., & Kadioglu, E. (2015). Predicting Financial Failure Using Decision Tree Algorithms: An Empirical Test on the Manufacturing Industry at Borsa Istanbul. *International Journal of Economics and Finance*, *7*(7). https://doi.org/10.5539/ijef.v7n7p189

Odom, M. D., & Sharda, R. (1990). *A neural network model for bankruptcy prediction*. https://doi.org/10.1109/ijcnn.1990.137710

Ohlson, J. A. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research*, *18*(1), 109. https://doi.org/10.2307/2490395

Opitz, D. W., & Maclin, R. (1999). Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, *11*, 169–198. https://doi.org/10.1613/jair.614

Oreški, G. (2014). An experimental comparison of classification algorithm performances for highly imbalanced datasets. *ResearchGate*.

https://www.researchgate.net/publication/282673032_An_experimental_comparison_of_classification_algorithm_performances_for_highly_imbalanced_datasets

Pawełek, B. (2019). Extreme Gradient Boosting Method in the Prediction of Company Bankruptcy. *Statistics in Transition New Series*, *20*(2), 155–171. https://doi.org/10.21307/stattrans-2019-020

Pirizadeh, M., Alemohammad, N., Manthouri, M., & Pirizadeh, M. (2021). A new machine learning ensemble model for class imbalance problem of screening enhanced oil recovery methods. *Journal of Petroleum Science and Engineering*, *198*, 108214. https://doi.org/10.1016/j.petrol.2020.108214

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*(3), 21–45. https://doi.org/10.1109/mcas.2006.1688199

Provost, F., & Fawcett, T. (1998). Robust classification systems for imprecise environments. In *National Conference on Artificial Intelligence* (pp. 706–713). http://dblp.uni-trier.de/db/conf/aaai/aaai98.html#ProvostF98

Prusak, B. (2018). Review of Research into Enterprise Bankruptcy Prediction in Selected Central and Eastern European Countries. *International Journal of Financial Studies*, *6*(3), 60. https://doi.org/10.3390/ijfs6030060

Pumsirirat, A., & Yan, L. (2018). Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine. *International Journal of Advanced Computer Science and Applications*, *9*(1). https://doi.org/10.14569/ijacsa.2018.090103

Quynh, T. M., & Phuong, T. T. (2020). *Improving the bankruptcy prediction by combining some classification models*. https://doi.org/10.1109/kse50997.2020.9287707

Ravi Kumar, P., & Ravi, V. (2007). Bankruptcy prediction in banks and firms via statistical and intelligent techniques – A review. *European Journal of Operational Research*, *180*(1), 1–28. https://doi.org/10.1016/j.ejor.2006.08.043

Reinhart, C. (2022). From Health Crisis to Financial Distress. *IMF Economic Review*, *70*(1), 4–31. https://doi.org/10.1057/s41308-021-00152-6

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408. https://doi.org/10.1037/h0042519

Sakurada, M., & Yairi, T. (2014). Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. https://doi.org/10.1145/2689746.2689747

Salchenberger, L., Cinar, E. M., & Lash, N. (1992). Neural Networks: A New Tool for Predicting Thrift Failures. *Decision Sciences*, *23*(4), 899–916. https://doi.org/10.1111/j.1540-5915.1992.tb00425.x

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*(2), 197–227. https://doi.org/10.1007/bf00116037

Smith, R. F., & Winakor, A. H. (1935). Changes in the financial structure of unsuccessful industrial corporations. *University of Illinois eBooks*. http://ci.nii.ac.jp/ncid/BA1651116X

Smogeli, P. O. (1987). Dokumentasjonsnotat SEBRA. *Statistisk Sentralbyrå*, *87*(44).

Soui, M., Smiti, S., Mkaouer, M. W., & Ejbali, R. (2019). Bankruptcy Prediction Using Stacked Auto-Encoders. *Applied Artificial Intelligence*, *34*(1), 80–100. https://doi.org/10.1080/08839514.2019.1691849

Tanaka, K., Kinkyo, T., & Hamori, S. (2016). Random forests-based early warning system for bank failures. *Economics Letters*, *148*, 118–121. https://doi.org/10.1016/j.econlet.2016.09.024

Treacy, W. F., & Carey, M. S. (2000). Credit risk rating systems at large US banks. *Journal of Banking and Finance*, *24*(1–2), 167–201. https://doi.org/10.1016/s0378-4266(99)00056-4

Tsai, C., & Wu, J. (2008). Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, *34*(4), 2639–2649. https://doi.org/10.1016/j.eswa.2007.05.019

UCI Machine Learning Repository. (n.d.-a). *Taiwanese Bankruptcy Prediction Data Set*. UCI. Retrieved June 6, 2023, from https://archive.ics.uci.edu/ml/datasets/Taiwanese+Bankruptcy+Prediction

UCI Machine Learning Repository. (n.d.-b). *Polish Companies Bankruptcy Data Set*. UCI. Retrieved June 6, 2023, from https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data

Upneja, A., & Dalbor, M. C. (2001). An examination of capital structure in the restaurant industry. *International Journal of Contemporary Hospitality Management*, *13*(2), 54–59. https://doi.org/10.1108/09596110110381825

Ursin, G., Skjesol, I., & Tritter, J. (2020). The COVID-19 pandemic in Norway: The dominance of social implications in framing the policy response. *Health Policy and Technology*, *9*(4), 663–672. https://doi.org/10.1016/j.hlpt.2020.08.004

van der Maaten, L., & Hinton, G. E. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*(86), 2579–2605. http://isplab.tudelft.nl/sites/default/files/vandermaaten08a.pdf

van der Walt, S., & Smith, N. (2015). Matplotlib Colormaps: Viridis. Retrieved from https://bids.github.io/colormap/

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning - ICML '08*. https://doi.org/10.1145/1390156.1390294

Wahlstrøm, R. R., & Helland, F. (2016). *Konkursprediksjon for norske selskaper – en analyse ved maskinlæringsteknikker og tradisjonelle statistiske metoder* [Master's Thesis]. NTNU.

Wilson, R. K., & Sharda, R. (1994). Bankruptcy prediction using neural networks. *Decision Support Systems*, *11*(5), 545–557. https://doi.org/10.1016/0167-9236(94)90024-8

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259. https://doi.org/10.1016/s0893-6080(05)80023-1

Wyrobek, J., & Kluza, K. (2018). Efficiency of Gradient Boosting Decision Trees Technique in Polish Companies' Bankruptcy Prediction. In *Advances in intelligent systems and computing* (pp. 24–35). Springer Nature. https://doi.org/10.1007/978-3-319-99993-7_3

Xu, H., Feng, Y., Chen, J., Wang, Z., Qiao, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., & Pei, D. (2018). Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications. *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*. https://doi.org/10.1145/3178876.3185996

Xu, W., Keshmiri, S., & Wang, G. (2019). Adversarially Approximated Autoencoder for Image Generation and Manipulation. *IEEE Transactions*

on Multimedia, *21*(9), 2387–2396. https://doi.org/10.1109/tmm.2019.2898777

Xu, W., Sun, H., Deng, C., & Tan, Y. (2017). Variational Autoencoder for Semi-Supervised Text Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1). https://doi.org/10.1609/aaai.v31i1.10966

Yeh, C., Chi, D., & Lin, Y. (2014). Going-concern prediction using hybrid random forests and rough set approach. Information Sciences, 254, 98–110. https://doi.org/10.1016/j.ins.2013.07.011

Zhao, Z., Xu, S., Kang, B. H., Kabir, M. M. J., Liu, Y., & Wasinger, R. (2015). Investigation and improvement of multi-layer perceptron neural networks for credit scoring. *Expert Systems with Applications*, *42*(7), 3508–3516. https://doi.org/10.1016/j.eswa.2014.12.006

Zhou, C., & Paffenroth, R. C. (2017). Anomaly Detection with Robust Deep Autoencoders. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/3097983.3098052

Zhuang, F., Cheng, X., Luo, P., Pan, S. J., & He, Q. (2015). Supervised representation learning: transfer learning with deep autoencoders. *International Conference on Artificial Intelligence*, 4119–4125. https://ijcai.org/Proceedings/15/Papers/578.pdf

Zięba, M., Tomczak, S. K., & Tomczak, J. M. (2016). Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems with Applications*, *58*, 93–101. https://doi.org/10.1016/j.eswa.2016.04.001

Zmijewski, M. E. (1984). Methodological Issues Related to the Estimation of Financial Distress Prediction Models. *Journal of Accounting Research*, *22*, 59. https://doi.org/10.2307/2490859

Appendices

Appendix 1

Variables in Brreg Data Set

Organization Number	Industry Code 2
Name	Industry Code 2 Description
Business Address	Industry Code 3
Business Address Postal Code	Industry Code 3 Description
Business Address City	Bankruptcy Opened
Mailing Address	Grounds for Bankruptcy
Mailing Address Postal Code	Role Type
Mailing Address City	Chief Executive Officer
Organization Type	CEO Date of Birth
Industry Code	Board Leader/Chair
Industry Code Description	Board Leader/Chair Date of Birth

Appendix 2

Variables in Taiwanese Data Set

Variable	Description
Y	Class Label
X1	ROA(C) Before Interest and Depreciation Before Interest: Return on Total Assets(C)
X2	ROA(A) Before Interest and % After Tax: Return on Total Assets(A)
X3	ROA(B) Before Interest and Depreciation After Tax: Return on Total Assets(B)
X4	Operating Gross Margin: Gross Profit/Net Sales
X5	Realized Sales Gross Margin: Realized Gross Profit/Net Sales
X6	Operating Profit Rate: Operating Income/Net Sales
X7	Pre-Tax Net Interest Rate: Pre-Tax Income/Net Sales
X8	After-Tax Net Interest Rate: Net Income/Net Sales
X9	Non-Industry Income and Expenditure/Revenue: Net Non-Operating Income Ratio
X10	Continuous Interest Rate (After Tax): Net Income-Exclude Disposal Gain or Loss/Net Sales
X11	Operating Expense Rate: Operating Expenses/Net Sales
X12	Research And Development Expense Rate: (Research and Development Expenses)/Net Sales
X13	Cash Flow Rate: Cash Flow from Operating/Current Liabilities

Appendix 2

Variables in Taiwanese Data Set

Variable	Description
X14	Interest-Bearing Debt Interest Rate: Interest-Bearing Debt/Equity
X15	Tax Rate (A): Effective Tax Rate
X16	Net Value per Share (B): Book Value per Share(B)
X17	Net Value per Share (A): Book Value per Share(A)
X18	Net Value per Share (C): Book Value per Share(C)
X19	Persistent Eps in The Last Four Seasons: Eps-Net Income
X20	Cash Flow per Share
X21	Revenue per Share (Yuan ¥): Sales per Share
X22	Operating Profit per Share (Yuan ¥): Operating Income per Share
X23	Per Share Net Profit Before Tax (Yuan ¥): Pretax Income per Share
X24	Realized Sales Gross Profit Growth Rate
X25	Operating Profit Growth Rate: Operating Income Growth
X26	After-Tax Net Profit Growth Rate: Net Income Growth
X27	Regular Net Profit Growth Rate: Continuing Operating Income After Tax Growth
X28	Continuous Net Profit Growth Rate: Net Income-Excluding Disposal Gain or Loss Growth
X29	Total Asset Growth Rate: Total Asset Growth
X30	Net Value Growth Rate: Total Equity Growth
X31	Total Asset Return Growth Rate Ratio: Return on Total Asset Growth
X32	Cash Reinvestment %: Cash Reinvestment Ratio
X33	Current Ratio
X34	Quick Ratio: Acid Test
X35	Interest Expense Ratio: Interest Expenses/Total Revenue
X36	Total Debt/Total Net Worth: Total Liability/Equity Ratio
X37	Debt Ratio %: Liability/Total Assets
X38	Net Worth/Assets: Equity/Total Assets
X39	Long-Term Fund Suitability Ratio (A): (Long-Term Liability+Equity)/Fixed Assets
X40	Borrowing Dependency: Cost of Interest-Bearing Debt
X41	Contingent Liabilities/Net Worth: Contingent Liability/Equity
X42	Operating Profit/Paid-In Capital: Operating Income/Capital
X43	Net Profit Before Tax/Paid-In Capital: Pretax Income/Capital
X44	Inventory And Accounts Receivable/Net Value: (Inventory+Accounts Receivables)/Equity

Appendix 2

Variables in Taiwanese Data Set

Variable	Description
X45	Total Asset Turnover
X46	Accounts Receivable Turnover
X47	Average Collection Days: Days Receivable Outstanding
X48	Inventory Turnover Rate (Times)
X49	Fixed Assets Turnover Frequency
X50	Net Worth Turnover Rate (Times): Equity Turnover
X51	Revenue per Person: Sales per Employee
X52	Operating Profit per Person: Operation Income per Employee
X53	Allocation Rate per Person: Fixed Assets per Employee
X54	Working Capital to Total Assets
X55	Quick Assets/Total Assets
X56	Current Assets/Total Assets
X57	Cash/Total Assets
X58	Quick Assets/Current Liability
X59	Cash/Current Liability
X60	Current Liability to Assets
X61	Operating Funds to Liability
X62	Inventory/Working Capital
X63	Inventory/Current Liability
X64	Current Liabilities/Liability
X65	Working Capital/Equity
X66	Current Liabilities/Equity
X67	Long-Term Liability to Current Assets
X68	Retained Earnings to Total Assets
X69	Total Income/Total Expense
X70	Total Expense/Assets
X71	Current Asset Turnover Rate: Current Assets to Sales
X72	Quick Asset Turnover Rate: Quick Assets to Sales
X73	Working Capital Turnover Rate: Working Capital to Sales
X74	Cash Turnover Rate: Cash to Sales
X75	Cash Flow to Sales
X76	Fixed Assets to Assets
X77	Current Liability to Liability
X78	Current Liability to Equity
X79	Equity to Long-Term Liability

Appendix 2

Variables in Taiwanese Data Set

Variable	Description
X80	Cash Flow to Total Assets
X81	Cash Flow to Liability
X82	CFO to Assets
X83	Cash Flow to Equity
X84	Current Liability to Current Assets
X85	Liability-Assets Flag: 1 If Total Liability Exceeds Total Assets, 0 Otherwise
X86	Net Income to Total Assets
X87	Total Assets to GNP Price
X88	No-Credit Interval
X89	Gross Profit to Sales
X90	Net Income to Stockholder's Equity
X91	Liability Tt Equity
X92	Degree of Financial Leverage (DFL)
X93	Interest Coverage Ratio (Interest Expense to EBIT)
X94	Net Income Flag: 1 If Net Income Is Negative for The Last Two Years, 0 Otherwise
X95	Equity to Liability

Appendix 3

Variables in Polish Data Set

Variable	Description
X1	Net Profit / Total Assets
X2	Total Liabilities / Total Assets
X34	Working Capital / Total Assets
X4	Current Assets / Short-Term Liabilities
X5	[(Cash + Short-Term Securities + Receivables - Short-Term Liabilities) / (Operating Expenses - Depreciation)] * 365
X6	Retained Earnings / Total Assets
X7	EBIT / Total Assets
X8	Book Value of Equity / Total Liabilities
X9	Sales / Total Assets
X10	Equity / Total Assets
X11	(Gross Profit + Extraordinary Items + Financial Expenses) / Total Assets

Appendix 3

Variables in Polish Data Set

Variable	Description
X12	Gross Profit / Short-Term Liabilities
X13	(Gross Profit + Depreciation) / Sales
X14	(Gross Profit + Interest) / Total Assets
X15	(Total Liabilities * 365) / (Gross Profit + Depreciation)
X16	(Gross Profit + Depreciation) / Total Liabilities
X17	Total Assets / Total Liabilities
X18	Gross Profit / Total Assets
X19	Gross Profit / Sales
X20	(Inventory * 365) / Sales
X21	Sales (N) / Sales (N-1)
X22	Profit on Operating Activities / Total Assets
X23	Net Profit / Sales
X24	Gross Profit (In 3 Years) / Total Assets
X25	(Equity - Share Capital) / Total Assets
X26	(Net Profit + Depreciation) / Total Liabilities
X27	Profit on Operating Activities / Financial Expenses
X28	Working Capital / Fixed Assets
X29	Logarithm of Total Assets
X30	(Total Liabilities - Cash) / Sales
X31	(Gross Profit + Interest) / Sales
X32	(Current Liabilities * 365) / Cost of Products Sold
X33	Operating Expenses / Short-Term Liabilities
X34	Operating Expenses / Total Liabilities
X35	Profit on Sales / Total Assets
X36	Total Sales / Total Assets
X37	(Current Assets - Inventories) / Long-Term Liabilities
X38	Constant Capital / Total Assets
X39	Profit on Sales / Sales
X40	(Current Assets - Inventory - Receivables) / Short-Term Liabilities
X41	Total Liabilities / ((Profit on Operating Activities + Depreciation) * (12/365))
X42	Profit on Operating Activities / Sales
X43	Rotation Receivables + Inventory Turnover in Days
X44	(Receivables * 365) / Sales
X45	Net Profit / Inventory

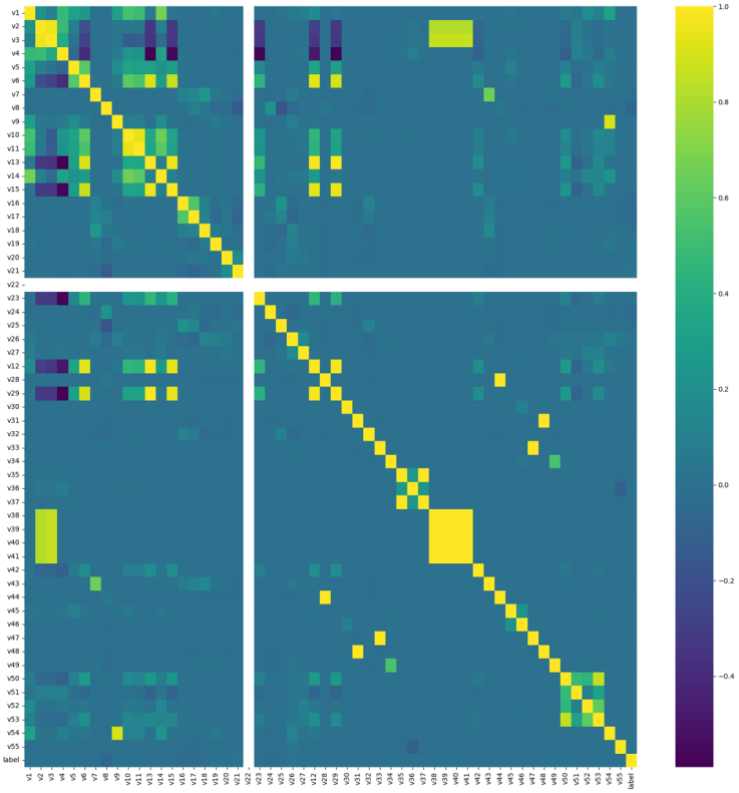
Appendix 3

Variables in Polish Data Set

Variable	Description
X46	(Current Assets - Inventory) / Short-Term Liabilities
X47	(Inventory * 365) / Cost of Products Sold
X48	EBITDA (Profit on Operating Activities - Depreciation) / Total Assets
X49	EBITDA (Profit on Operating Activities - Depreciation) / Sales
X50	Current Assets / Total Liabilities
X51	Short-Term Liabilities / Total Assets
X52	(Short-Term Liabilities * 365) / Cost of Products Sold)
X53	Equity / Fixed Assets
X54	Constant Capital / Fixed Assets
X55	Working Capital
X57	(Sales - Cost of Products Sold) / Sales
X58	Total Costs /Total Sales
X59	Long-Term Liabilities / Equity
X60	Sales / Inventory
X61	Sales / Receivables
X62	(Short-Term Liabilities *365) / Sales
X63	Sales / Short-Term Liabilities
X64	Sales / Fixed Assets

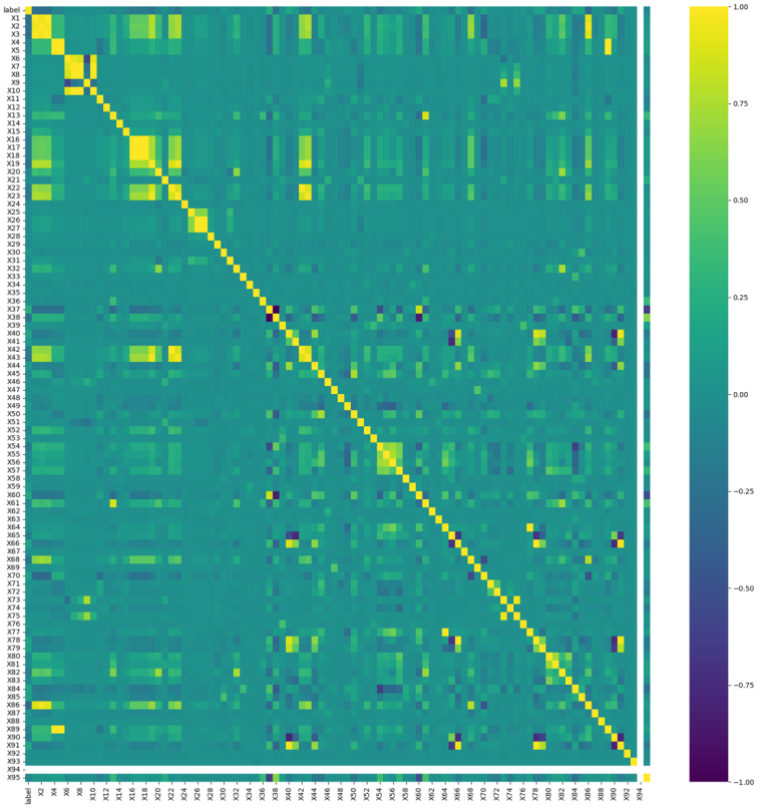
Appendix 4

Correlation Matrix Heatmap for Norwegian Data Set



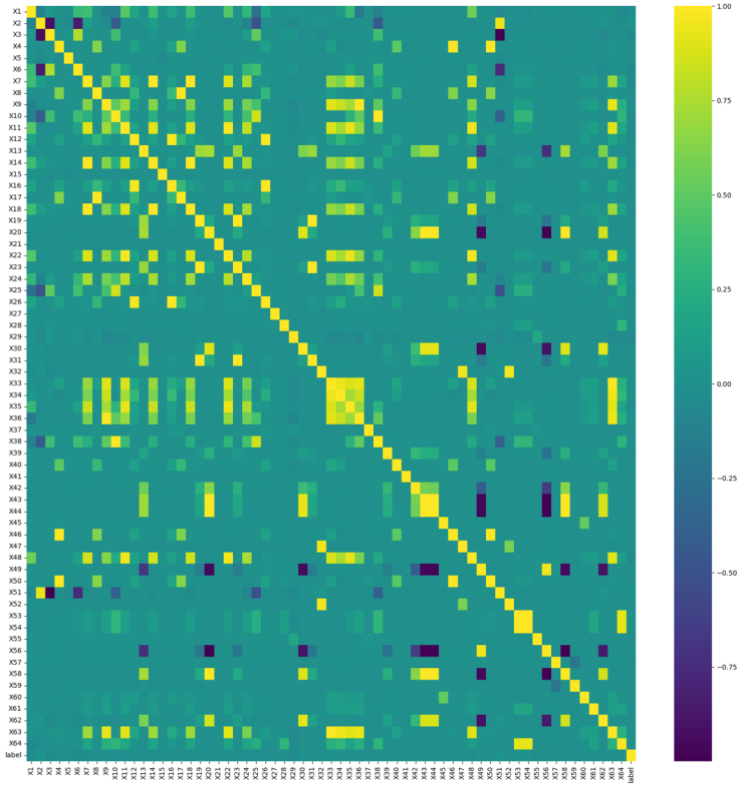
Appendix 5

Correlation Matrix Heatmap for Taiwanese Data Set



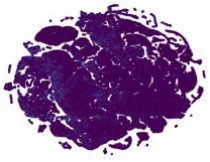
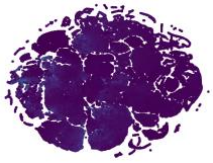
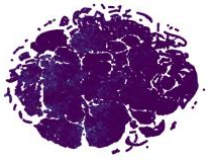
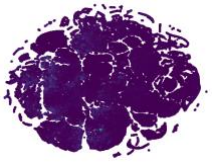
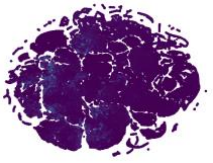
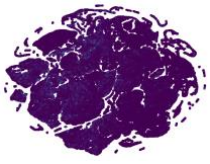
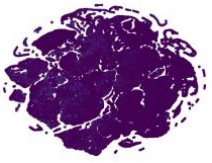
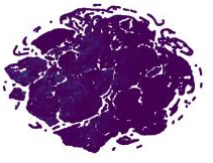
Appendix 6

Correlation Matrix Heatmap for Polish Data Set



Appendix 7

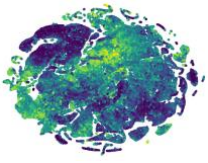
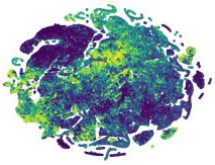
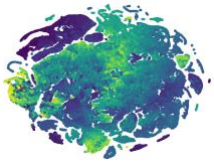
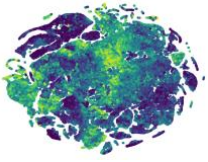
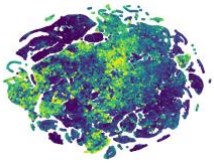
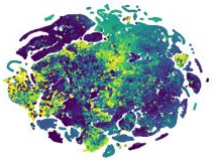
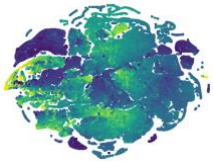
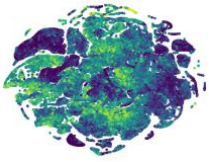
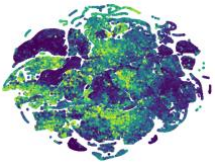
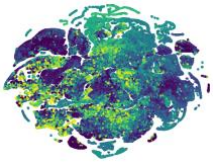
Latent Representation of Norwegian Data Set using Original Data

z	LR	RF	XGB	MLP
20				
25				
30				

Note. This visualization is derived from z_{mean} obtained through the trained VAE. Dimensionality reduction is achieved via t-SNE (van der Maaten & Hinton, 2008). The colors represent the risk profiles associated with each data point, as determined by the corresponding classifier. The 'viridis' color scale (van der Walt & Smith, 2015) is employed, with yellower shades indicating a higher risk of bankruptcy.

Appendix 8

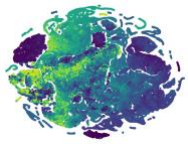
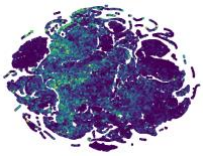
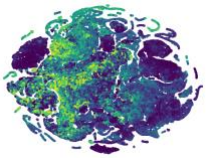
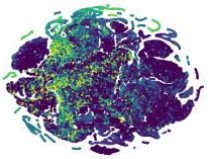
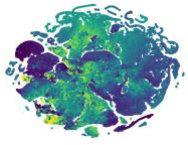
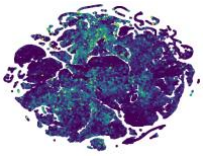
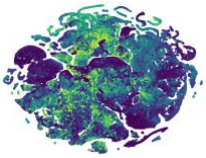
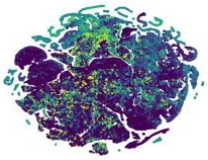
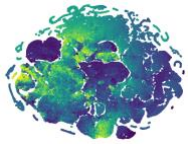
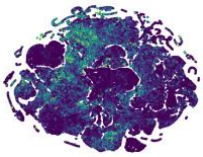
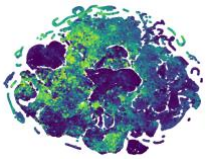
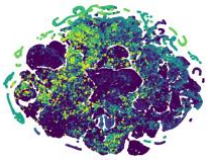
Latent Representation of Norwegian Data Set using Under-Sampling Train Set

z	LR	RF	XGB	MLP
20				
25				
30				

Note. This visualization is derived from z_{mean} obtained through the trained VAE. Dimensionality reduction is achieved via t-SNE (van der Maaten & Hinton, 2008). The colors represent the risk profiles associated with each data point, as determined by the corresponding classifier. The 'viridis' color scale (van der Walt & Smith, 2015) is employed, with yellower shades indicating a higher risk of bankruptcy.

Appendix 9

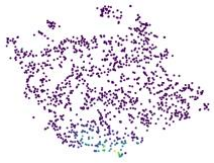
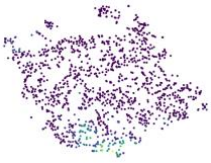
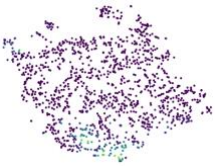
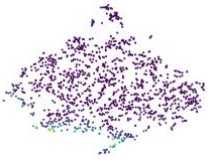
Latent Representation of Norwegian Data Set using SMOTE Train Set

z	LR	RF	XGB	MLP
20				
25				
30				

Note. This visualization is derived from z_{mean} obtained through the trained VAE. Dimensionality reduction is achieved via t-SNE (van der Maaten & Hinton, 2008). The colors represent the risk profiles associated with each data point, as determined by the corresponding classifier. The 'viridis' color scale (van der Walt & Smith, 2015) is employed, with yellower shades indicating a higher risk of bankruptcy.

Appendix 10

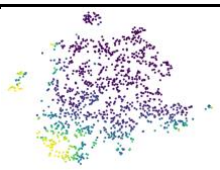
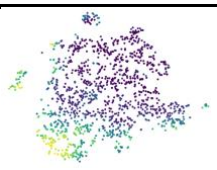
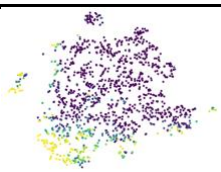
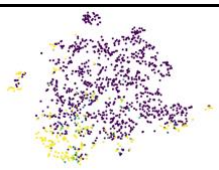
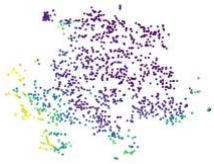
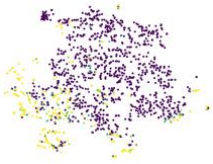
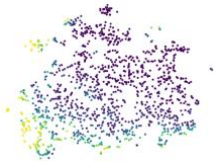
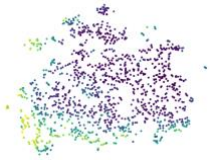
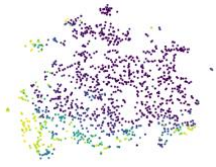
Latent Representation of Taiwanese Data Set using Original Data

z	LR	RF	XGB	MLP
30				
45				
60				

Note. This visualization is derived from z_{mean} obtained through the trained VAE. Dimensionality reduction is achieved via t-SNE (van der Maaten & Hinton, 2008). The colors represent the risk profiles associated with each data point, as determined by the corresponding classifier. The 'viridis' color scale (van der Walt & Smith, 2015) is employed, with yellower shades indicating a higher risk of bankruptcy.

Appendix 11

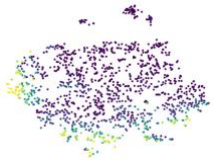
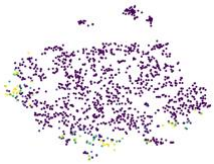
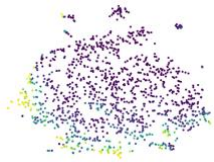
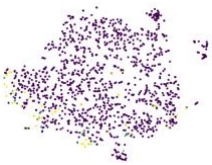
Latent Representation of Taiwanese Data Set using Under-Sampling Train Set

z	LR	RF	XGB	MLP
30				
45				
60				

Note. This visualization is derived from z_{mean} obtained through the trained VAE. Dimensionality reduction is achieved via t-SNE (van der Maaten & Hinton, 2008). The colors represent the risk profiles associated with each data point, as determined by the corresponding classifier. The 'viridis' color scale (van der Walt & Smith, 2015) is employed, with yellower shades indicating a higher risk of bankruptcy.

Appendix 12

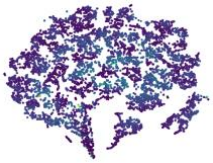
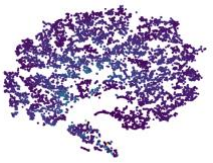
Latent Representation of Taiwanese Data Set using SMOTE Train Set

z	LR	RF	XGB	MLP
30				
45				
60				

Note. This visualization is derived from z_{mean} obtained through the trained VAE. Dimensionality reduction is achieved via t-SNE (van der Maaten & Hinton, 2008). The colors represent the risk profiles associated with each data point, as determined by the corresponding classifier. The 'viridis' color scale (van der Walt & Smith, 2015) is employed, with yellower shades indicating a higher risk of bankruptcy.

Appendix 13

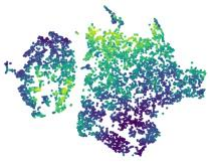
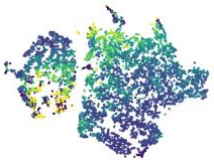
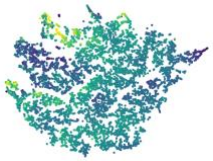
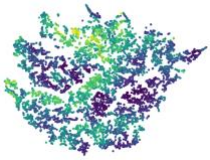
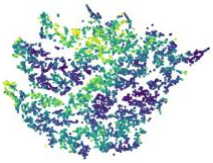
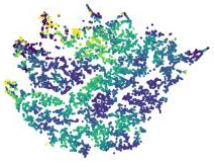
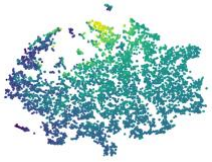
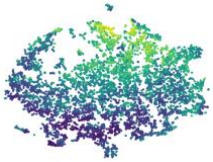
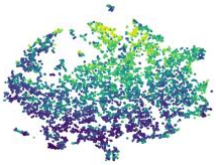
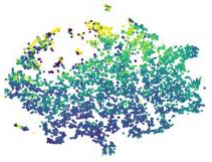
Latent Representation of Polish Data Set using Original Data

z	LR	RF	XGB	MLP
20				
30				
40				

Note. This visualization is derived from z_{mean} obtained through the trained VAE. Dimensionality reduction is achieved via t-SNE (van der Maaten & Hinton, 2008). The colors represent the risk profiles associated with each data point, as determined by the corresponding classifier. The 'viridis' color scale (van der Walt & Smith, 2015) is employed, with yellower shades indicating a higher risk of bankruptcy.

Appendix 14

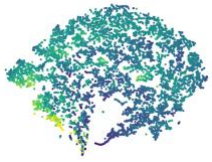
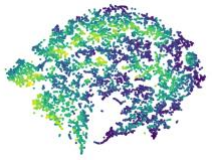
Latent Representation of Polish Data Set using Under-Sampling Train Set

z	LR	RF	XGB	MLP
20				
30				
40				

Note. This visualization is derived from z_{mean} obtained through the trained VAE. Dimensionality reduction is achieved via t-SNE (van der Maaten & Hinton, 2008). The colors represent the risk profiles associated with each data point, as determined by the corresponding classifier. The 'viridis' color scale (van der Walt & Smith, 2015) is employed, with yellower shades indicating a higher risk of bankruptcy.

Appendix 15

Latent Representation of Taiwanese Data Set using SMOTE Train Set

z	LR	RF	XGB	MLP
20				
30				
40				

Note. This visualization is derived from z_{mean} obtained through the trained VAE. Dimensionality reduction is achieved via t-SNE (van der Maaten & Hinton, 2008). The colors represent the risk profiles associated with each data point, as determined by the corresponding classifier. The 'viridis' color scale (van der Walt & Smith, 2015) is employed, with yellower shades indicating a higher risk of bankruptcy.