UNIVERSITY OF SOUTHAMPTON

Faculty of Social Sciences
Southampton Business School

Deep Learning for Credit Risk Management Under Market Complexity and Illiquidity

by

Kamesh Korangi
MBA, B.Tech
ORCiD: 0000-0001-6528-5092

A thesis for the degree of
Doctor of Philosophy

June 2025

University of Southampton

Abstract

Faculty of Social Sciences
Southampton Business School

Doctor of Philosophy

Deep Learning for Credit Risk Management Under Market Complexity and Illiquidity

by Kamesh Korangi

This thesis investigates three problems relating to the credit or financial risk management of Small and Medium-sized Enterprises (SMEs) and listed mid-cap firms. Both types of firms face specific challenges affecting their access to credit. Mid-caps firms have to deal with various market complexities and are often crowded out by large-cap firms, whilst SMEs have only indirect market exposure and collectively are a considerable risk to lenders. By utilising alternative data and recent advances in deep learning, the three papers forming this thesis develop and empirically test a series of novel prediction methods that can contribute to decreasing the cost of capital for these firms and enhance risk management practices for lenders.

First, in Chapter 1, an introduction is provided outlining the contextual setting of the thesis, the research aims and the intended contributions of the three papers.

The first paper (Chapter 2) is a study on default prediction for mid-cap firms, which introduces the challenges they face and modelling issues for predicting the default term structure. Different deep learning models are introduced and a novel multimodal architecture is proposed to make effective use of fundamental, market and pricing data, along with a framework to interpret the model predictions. The results show that deep learning models are powerful predictors and confirm some results from the literature.

The second paper (Chapter 3) studies large-scale time-varying portfolio optimisation for the same class of mid-cap firms. It shows how to filter complex networks at a large scale by combining existing techniques in a novel way. These networks are then used as inputs to a deep learning architecture that employs graph neural network models and a series of further layers for portfolio selection. The results confirm the effectiveness of using network information when devising portfolios to maximise return per risk, showing robust performance of the graph neural networks over long periods. Unlike earlier studies, this study shows that investing in peripheral firms in the networks might create additional risks. To our knowledge, this is the first study

that includes firms that defaulted over the data period and explicitly considers changes in the universe of investable firms over time.

The third paper (Chapter 4) studies credit lines of small and medium-scale enterprises to predict their default probability using behavioural and network data. Building on the previous two papers, we use a multimodal model with graph neural networks and deep learning to advance behavioural credit default prediction models. We use explicit networks from transactions, ownership and supply chain relationships over a large set of such firms and together with behavioural data that we derive from the revolving credit lines usage, we find the behavioural data highly predictive of default whilst the need for more complex models arises when using the network data.

Finally, Chapter 5 concludes with methodological contributions and the scope of application of these studies, individually and collectively. It also puts forward ideas for future studies that could extend the application of deep learning models to other credit risk modelling problems.

Contents

List of Figures

List of Tables

Declaration of Authorship

Print Name: Kameswara Rao Korangi

Title of thesis: Deep Learning for credit risk management under market complexity and illiquidity

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:
   Kamesh Korangi, Christophe Mues, and Cristián Bravo. A transformer-based model for default prediction in mid-cap corporate markets. *European Journal of Operational Research*, 308(1):306–320, October 2022. ISSN 0377-2217

Signed:.. Date:..................

Acknowledgements

The journey of a PhD is often akin to a hurdle race—every hurdle is a chance to explore the frontier of research, but also likely to fail in that endeavour. It is also a benchmark for other potential life races. I feel immensely fortunate to have been guided and coached through this experience by my supervisors, Prof. Christophe Mues and Prof. Cristian Bravo. Their invaluable mentorship and unwavering support have been a cornerstone of my academic and personal growth over the past few years. Whether offering a helping hand or compassionate understanding, their guidance has been instrumental in bringing me to this milestone—a journey shared with them that I will always cherish.

I am profoundly grateful to my sponsors, SCDTP and UKRI, for supporting my pursuit of a PhD on a topic I was deeply passionate about. As I approached the conclusion of my doctoral work, I owe special thanks to my managers at AFII, Dr. Ulf Erlandsson and Justine Leigh-Bell, for their exceptional understanding of the demands of a PhD. Their flexibility and encouragement ensured that I could balance my academic endeavors with professional growth seamlessly.

Such a fulfilling journey would have been impossible without the strength of the relationships around me. I extend my heartfelt thanks to my family and friends for their encouragement—perhaps more accurately, their bewilderment at my path—yet their trust in me and willingness to let me find my own way made all the difference.

Finally, my deepest gratitude goes to my soulmate, Priya Korangi. Without her unwavering support, I would not have had the courage to begin this journey, let alone the fortitude to complete it. Our shared journey began with her Master's thesis acknowledging me, and nearly two decades later, it is my turn. Together, we are so much more than we could ever be alone. Our adventure continues with our two wonderful children, Amrutha and Kriyansh, and I look forward to the learning and challenges ahead with gratitude and love.

To my dearest cousin, Ashok, whose absence I deeply feel every day.

Chapter 1

Introduction

This thesis puts forward a series of novel credit risk management methods for firms. Credit risk is the possibility of a loss to the lender due to a borrower's inability to meet their repayment obligations. Managing this risk implies an ability to predict, over a certain period, the likelihood of borrowers defaulting, quantify the size of the losses if they do default, and understand various common risks inherent in such borrowers. The timing of these risks may also be studied to manage them better. A risky firm could be denied credit right at the application stage, or a firm's health could deteriorate after credit has been granted, implying that models are needed for credit assessment, monitoring and portfolio risk management. As credit is fundamental to economic and financial stability and thus systemically important, it is also a highly regulated area, often overseen by central banks. Hence, firms need to understand the models that are subject to regulatory requirements as part of their credit risk management processes. Effective and transparent credit risk model development is key to advancing all of these areas of credit risk management.

Firms are typically classified based on their size, according to revenue or number of employees. In this thesis, we focus on listed mid-cap firms and small and medium-sized enterprises (SMEs). Figure 1.1 provides a comprehensive categorisation of firms by size, with higher layers in the pyramid being associated with fewer firms. In the literature, we see that large-caps are often the focus of study as clean data is available for such listed firms. However, due to their strong competitive position, their defaults are less susceptible to market factors and are more due to isolated governance issues. This means they have low default rates, and with a low volume of firms, they lend themselves less to statistical analysis. Mid-caps may be better suited for credit risk studies, but they do present data availability problems. Also, they may be prone to changing their accounting methods and could exhibit illiquid trading patterns. Therefore, current credit risk models for listed mid-cap firms tend to still be primarily qualitative, with only a few quantitative factors. For example, they may combine industry, business, and governance risk factors into a single firm rating or feature qualitative judgements made by analysts. In contrast, SMEs are generally private enterprises with similar information asymmetry issues, and timely data is even more complex to acquire for research purposes. Credit risk models for SMEs typically have more in common with quantitative models for individual borrowers, as SMEs are far more numerous than listed large-cap firms. By studying both mid-caps and SMEs, albeit in different papers, we are able to redeploy some of the techniques we propose for listed mid-caps to overcome some of the data issues for SMEs. Our research thus starts with default prediction for mid-caps and ends with a similar problem for SMEs, employing methods that are connected primarily through deep learning and network models.

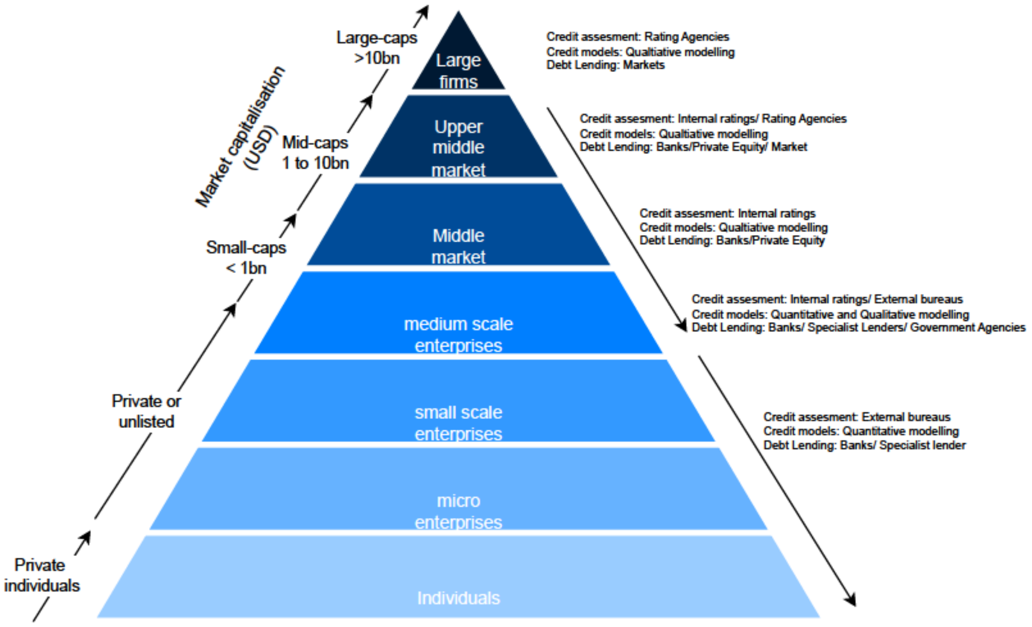

Large-caps
>10bn
Market capitalisation
(USD)
Mid-caps
1 to 10bn
Small-caps
< 1bn
Private or
unlisted
Private
individuals
Large
firms
Upper
middle
market
Middle
market
medium scale
enterprises
small scale
enterprises
micro
enterprises
Individuals
Credit assesment: Rating Agencies
Credit models: Qualtiative modelling
Debt Lending: Markets
Credit assesment: Internal ratings/ Rating Agencies
Credit models: Qualtiative modelling
Debt Lending: Banks/Private Equity/ Market
Credit assesment: Internal ratings
Credit models: Qualtiative modelling
Debt Lending: Banks/Private Equity
Credit assesment: Internal ratings/ External bureaus
Credit models: Quantitative and Qualitative modelling
Debt Lending: Banks/ Specialist Lenders/ Government Agencies
Credit assesment: External bureaus
Credit models: Quantitative modelling
Debt Lending: Banks/ Specialist lender

FIGURE 1.1: Types of firms by size and typical credit risk models and main credit or debt providers for such firms. Sources: Modified and adapted from multiple sources (Chan and Lazzara, 2015; Commission, 2021; Jayroe, 2024)

Deep learning models are a subset of Machine Learning (ML) methods, which are, in turn, an important class of Artificial Intelligence (AI) methods. Deep learning models have produced state-of-the-art results in various domains, aided by breakthroughs in computational techniques. They are particularly adept at extracting relationships where the data is available on a large scale but could be sparse or assume different formats such as textual, networks or image data, all of which are unsuitable for earlier classes of AI methods. In this thesis, we further build upon these models to adapt them for credit risk problems and, in doing so, improve the performance of credit risk models and also find novel ways to build deep learning architectures that use different forms of data. Furthermore, we empirically test them over extended time frames or a large population of firms.

Starting with all US mid-caps over a period of 30 years, the first paper addresses empirical evidence in credit risk pricing suggesting that existing models underestimate the market-implied risk. This shows that further improvement in the modelling is necessary, and we seek to address that by taking a more holistic view and considering different data sources with varying frequencies of observations. With only deep learning models able to effectively handle daily data combined with annual statements, we adapt a transformer model and develop a training routine for it to extract maximum value from the data. We use a single deep learning model to establish the whole term structure of default, which is impossible with many traditional models. This study also shows us that deep learning models are suitable where data is large-scale but sparse, with missing data.

Improving the accuracy of default prediction for each firm, using firm-level data, is only one part of the process. All of these firms operate within a shared market environment and face similar business and competitive pressures. This leads to correlated risks, which creates new risks for portfolio management. In the second paper, we work on the same dataset of mid-caps, but now analyse their relationships more extensively and seek to build portfolios that can outperform traditional portfolio optimisation methods. Using a combination of network filtering and graph neural networks, we develop novel methods that go beyond conventional linear covariance matrices to decide on the capital allocation for such firms. Unlike previous studies that commonly ignored defaulted firms, we explicitly include them in our universe of firms, which adds to the practical appeal of our method.

For the third paper, we combine some of the methods of the first and second paper and apply them to a unique large-scale SME dataset. We partnered with one of the largest financial institutions to source the data, which covers a large collection of SME firms and explicit networks between them. Unlike in the first two papers, the problem we concentrate on is credit monitoring, for which use the firms' behavioural data. We propose a deep learning multimodal architecture that combines the graph neural networks used in Paper 2 with the architecture from Paper 1, to create a robust model. This is able to distill information from large-scale networks with multiple relationship factors. This study thus seeks to quantify the predictive value of these networks for SMEs and the stability of using such complex signals for default prediction.

The rest of this introduction chapter is structured as follows: We start with the economic contributions by SMEs and the challenges they face, in Section 1.1. In Section 1.2, we further characterise their environment by looking at the current state of credit risk models (Section 1.2.1) and the leading players in this industry (Section 1.2.2), and by giving a brief overview of regulations that govern these credit risk models (Section 1.2.3). Looking for solutions, next, we provide a brief overview of deep learning models (Section 1.3), with the history in Section 1.3.2, different data inputs they work with (Section 1.3.3), model assessment, and finally surveying some relevant existing studies that use deep learning in finance (Section 1.3.7). In Section 1.4, we then introduce the primary objectives of this thesis, bringing together all the questions raised up to that point. Finally, Section 1.5 concludes the introduction by setting out the contributions made by each of the papers in answering those research objectives.

1.1 SMEs and the challenges they face

Worldwide, Small and Medium-scale Enterprises (SMEs) are the largest job generators and significant contributors to any national economy. The SME definition varies by country and region; in Europe, these are firms with revenue of less than 50 million euros and fewer than 250 employees. In the US, however, the definition varies by sector, but generally, they have fewer than 500 employees. Typically, they are unlisted or private firms and have relationships with banks as their primary credit providers. As employment generators, they provide around 60% of employment in most countries; in Europe, they are reported to account for 65% for jobs and contribute 52% of the value generated by all firms (Di Bella et al., 2023).

Compared to larger firms, there is, however, a substantial gap in their access to finance and especially the cost of capital, which impacts SME growth (Ayyagari et al., 2007; Rao et al., 2021; OECD, 2024). The process lenders use to score SMEs sits somewhere between the personalised approaches used for large corporates and the fully automated credit approval process in place for consumers, as illustrated in Figure 1.1. This dual approach adds to the cost for the banks and financial intermediaries that provide SME credit (Munro, 2013). The lack of structured data, ownership issues, legacy banking systems, and limitations of the models themselves contribute further to this problem (Beck and Demirguc-Kunt, 2006; Moscalu et al., 2020).

Another risk is their exposure to market or environmental shocks. The 2008 financial crisis and the 2021 pandemic showed a widening growth gap, with larger firms emerging much stronger after such shocks. Demirgüç-Kunt et al. (2020) show significant deleveraging of SMEs after the global financial crisis, indicating a lack of market access. In contrast, large-caps could access the markets as and when support mechanisms developed by the government were deployed. This shows a different mechanism is needed for SMEs. There are structural issues with credit access, which means that the most innovative firms continue to suffer while non-innovative firms within SMEs face the cyclical risk of lack of access (Lee et al., 2015). Similarly, the pandemic has restricted access to capital for SMEs compared to other small firms or even larger firms with credit constraints (Zhang and Sogn-Grundvåg, 2022). Adian et al. (2020) find larger financing gaps as more information partnerships are required to be able to provide financing. Although shocks such as these could be quite infrequent, they constitute considerable risks that models must be able to cope with. As a result, traditional models appear less suitable to such firms.

1.2 Credit markets for SMEs

Economic growth is strongly linked to credit growth and the maturity of credit markets. Studies show asymmetry where the credit flow is towards larger enterprises, especially during periods of high volatility, which forces SMEs and even larger unlisted firms to deleverage (Demirgüç-Kunt et al., 2020). This deleveraging impacts growth on the upturn and job growth. Credit growth is possible when there are developed institutions that can take credit risk. Among these, financial institutions, particularly banks, have credit risk as their most significant risk-taking activity. Financial markets have distributed this credit risk across diverse financial investors, promising improved risk management by diversifying and matching the risk and enabling even more growth in the credit markets. Sound credit risk management is essential as these markets evolve, as several bodies of regulation, such as those based on the Basel Accords, show. Credit risk management aims to maximise the return on capital for the lending institution by adjusting for the risk exposure. This exposure is not limited to loans, but can also stem from other obligations such as guarantees in trade finance or dealing with several counterparties in different financial products such as derivatives, foreign exchange, and equity.

High rates of default of businesses have various multiplier effects and impact a country's economy as a whole. Given these costs, large-scale investments continue to improve corporate credit risk management techniques. Over the years, professional practice has evolved with many businesses providing different services. They include, for example, rating agencies predicting a firm's credit risk, data providers aggregating various data sources, and financial institutions such as banks and investors such as pension funds or mutual funds developing models of credit risk suitable for their own purposes. Academic research also continues to develop by studying various factors that contribute to the failure of firms at the micro and macro level and by introducing novel models for predicting default. While credit risk models exist for individuals and firms, modelling practice, too, has evolved, catering mostly to large companies or individuals. Rating agencies, using a mix of quantitative and qualitative factors, look to predict the credit risk of borrowers. Their rating process is costly, however, and so is limited to larger companies (Langohr and Langohr, 2010; White, 2013). Consumer credit risk is also well understood, with a small number of credit bureaus, such as Equifax and Experian, rating individual borrowers' credit risk in various countries. There is, however, a widely reported gap for smaller and medium-scale enterprises whose credit needs are not met within the current setup, as we have seen in studies mentioned earlier for SMEs.

The difference between the demand for credit and the supply of credit available to SMEs, including micro-enterprises, globally is a staggering USD 5.2 trillion (Bruhn et al., 2017). This gap is higher in developing countries where just the credit gap accounts to around 19 percent of GDP. Later studies have confirmed these gaps; in the informal sector, these gaps may be even higher. In the UK, a similar situation exists regarding SMEs, with data suggesting that only 17% of business loans in the UK were given to SMEs, whilst 83% were made to large corporations (Lu, 2018). The main solution is a collaboration with technology providers and data providers, which points towards the need for advanced models (Bruhn et al., 2017).

1.2.1 Corporate credit risk models

In this section, we briefly examine advances made with respect to three main types of credit models and summarise some of the literature on company default prediction. We highlight some of the issues with these models and their applicability to the problems we focus on.

1.2.1.1 Statistical models

Initial research on default prediction models in the academic area starts with the univariate analysis by Beaver (1966). Altman (1968) introduced multivariate analysis, which continues to be developed until today.

Beaver used a set of 30 financial accounting ratios and looked to predict the one year ahead probability of default of a company using five years of history of these ratios. The data set was a sample of 158 large firms, thus missing out on the defaults in smaller firms. An optimal cut-off was found for each ratio, which would reduce the classification error. The cash flow to debt ratio was found to be the most significant ratio in this prediction model. Altman's Z-Score model used a set of 22 financial accounting ratios and a few market equity-based ratios of the past year to predict one year ahead default probabilities. Five variables were selected using correlation analysis and significance testing of all the variables. These were used as inputs for the discriminant analysis models. The discriminant analysis produces a score from a linear combination of the ratios, called Z-score. The data set was smaller having 66 firms in total, all in manufacturing. However, recognising that large companies do not fail as often, they were excluded from the analysis, and as smaller companies did not have the required data, the dataset consisted of mid-size firms. This model was extended by Altman et al. (1977) into what was called the ZETA model, which made modifications to the original Z-Score model. A number of studies in between applied the model to different industries. The new model focused on more prominent companies with an average of USD 100 million in assets and a longer horizon from one year to five years. The dataset consisted of 111 firms. Unlike the original model, the ratios were adjusted by standardising the variables. The model used seven ratios and although evidence was found of non-linear relationships, the test sample still performed well with a linear discriminant analysis model.

Ohlson (1980) developed a logit model for default prediction for similar companies and called it the O-Score model. These logit models have since been deployed across a large set of credit risk models for individual consumers and firms. The resulting models are also widely studied and were extended to private companies using book values of equity rather than market ratios of initial models and to emerging market companies with similar ratios. Similar statistical models to predict credit ratings issued by rating agencies were developed, which are essential in day-to-day practice as the transition from one rating category to another impacts the portfolio mark to market losses. In 2007, for a smaller dataset of US-based SMEs, and again in 2010, for a more extensive set of UK-based SMEs, Z-score model variants were developed that address the challenge of the non-availability of financial statements (Altman and Sabato, 2012; Altman et al., 2010). They quantified behavioural variables such as audit quality and delays in financial statements or court proceedings, which improved the model. In our current approach, we deploy techniques that can better use non-financial information and disparate sources of information instead of explicitly coding them, thus reducing the cost of producing such information.

Variants of these statistical models such as Moody's RiskCalc, which was developed using more than 1.5 million firms and features extensive transformations to the financial ratios, are successfully used in practice (Falkenstein et al., 2000; Dwyer et al., 2004). The Altman-based SME models have led to the emergence of fintech firms that developed these models commercially (Sanga and Aziakpono, 2023). The logistic regression-based models are extensively used within banks and credit bureau agencies to calculate a risk score for firms and individuals. Most academic studies now use logistic regression as a benchmark when developing new models, as we do as well in our first paper.

1.2.1.2 Structural models

A second set of models are structural models, which use a combination of accounting and pricing information within an option theoretic framework. Merton (1974) developed the first such model using Black-Scholes options theory. The latter theory involves viewing the firm's equity value as the price of a call option on the firm's value, with the exercise price equal to the firm's debt level. At maturity, the option pays out if the firm's value is greater than the debt, or the equity value is zero if the firm's value is below the debt level. This pricing model can be converted to predict the probability of default as equivalent to the probability of the option being exercised from the debt holder's perspective, and also to determine the credit spreads for the debt instruments, generally bonds.

Options theory enables asset pricing using the underlying asset's volatility. However, the volatility of companies is observable only for public companies from the trading information of the equity prices. Hence, in its original form, this approach applies to large companies with public trading information. Also, a firm's default (PD) and recovery (RR) is assumed to depend on capital structure, which does not hold in real-world scenarios. The initial model was easily extended, relaxing certain assumptions such as flat interest rates (Black and Cox, 1976; Geske, 1977). Black and Cox later worked on allowing immediate default instead of waiting until maturity. Vasieck developed a model to price short-term loans (Vasicek, 1984). Kim et al. (1993) developed a corporate bond valuation model which takes away some of the assumptions and incorporates real-word features by modelling interest rates and incorporating other forms of default where a liquidity crunch could precipitate default even as the valuation of the firm could be high. Their main objective was to model the credit spreads more closely to the observed spread information in the real world, incorporating as many drivers of default. Longstaff and Schwartz (1995) showed how credit spreads of firms with similar default risk can vary significantly if the firms' assets have different correlations with changes in interest rates. They further extended their models to price floating rate credit risk. Further studies looked to find the determinants of credit spread changes. Duffie et al. (2007) used the structural approach to predict multi-period corporate default, in contrast to most other studies which focused on one year ahead default prediction. In the first paper of this thesis, we have a similar focus on producing multi-period probability of default predictions, as we believe the term structure of default probabilities is more useful in various applications.

In professional practice, Moody's KMV model uses a structural model approach to calculate the expected default frequency, with a time horizon of one year (Crosbie and Bohn, 2003). Compared to the statistical models, one of the difficulties that remains with this type of model, though, is how to adapt it to private companies, as, for those firms, no listed equity information is available.

1.2.1.3 Reduced-form models

Reduced-form models avoid many of the assumptions imposed by structural models. They do assume that a firm's default time is exogenous and that a default intensity function drives this default. This function can comprise latent state variables, accounting ratios, or market-related variables. One of the appeals of doing so is that, instead of having to estimate the probability of default over a specific time frame, intensity models can also give the duration to default. For example, Lane et al. (1986)

applied the Cox proportional hazard model to produce default probabilities for a dataset of 464 US banks. Their model uses similar ratios as many of the statistical models discussed in section 1.2.1.1 and was found to perform similarly to discriminant analysis based statistical models. Jarrow and Madan (1991) developed a hazard rate model to predict the bankruptcy of non-financial firms, over a period of thirty years. This paper used the ratios employed by previous statistical methods and combined them with additional market-based data; the hazard rate model outperformed others when using market-based data. Jarrow and Turnbull (1995) later extended these models allowing both the interest rate term structure and credit spread term structure to be stochastic, unlike previous structural models which assumed interest rates as fixed. These models were further adapted to use credit rating information and transition information. The advantage is that no capital structure information is needed, and the default is modelled as a Poisson process. Hull and White (2000), using the term structure of default probabilities, derived bond prices, applying these reduced-form models to price credit derivatives.

In professional practice, these models have been mainly applied to price credit derivatives like Credit Default Swaps (CDS). One of their advantages is that macroeconomic factors can be used to model the default process, making these models quite flexible. However, as they rely on public information, which needs constant trading information and bond prices, difficulties in applying them to private companies or companies with illiquid trading patterns, and especially those with non-tradeable debt, generally make them unsuitable for SMEs or mid-caps.

1.2.1.4 Private company models

Unlike the statistical models discussed earlier, other types of credit models cannot easily be extended to SMEs or private companies. Companies like Dun & Bradstreet and Experian do provide business scores for private companies, but they are targeted towards trade credit for suppliers and purchasers who seek reassurance as to whether a company can meet the very short term obligations that arise out of these contracts. In this section, we look briefly at the main models for private companies.

Developed by Moody's, RiskCalc for private companies (Falkenstein et al., 2000) uses a statistical approach to determine the most critical accounting ratios to estimate the probability of default. Their solution is intended for firms too large to be considered a simple extension of an individual, but without publicly traded equity information. Each accounting ratio variable is transformed using non-parametric approaches by interpolation mapping. This is just a univariate analysis for each ratio, which, when ordered by percentile, has a similar distribution to the default frequency. Lower-rated firms have exponentially increasing default rates compared to higher-rated firms; the ratio should exhibit similar characteristics to be included in the model. These transformed ratios are used as inputs to the generalised probit model, which generates unadjusted probabilities of default. The latter are subsequently transformed to expected default frequencies, which are finally mapped onto Moody's rating scale. The time horizon used by RiskCalc is either 1 year or a longer term of 5 years. The longer term may be more relevant considering that, at the time a loan is originated, problematic credit applications were unlikely to be approved.

These models tend to use only accounting information and are trained on a large data set of private companies, making them more robust. They have later been extended to include economic indicators as well (Dwyer et al., 2004). In the first part of our thesis, even though we deal with public companies, they are somewhat similar to private companies in that financial accounting information is more readily available than equity pricing information.

1.2.1.5 Portfolio credit models

Through the 1990s, as credit markets developed innovative products catering to different kinds of firms and investors, the credit risk management of all such different products became quite complex. There was no single measure of risk for different products that would allow one to compare a portfolio of bonds and CDS with counterparty credit risk. Credit value-at-risk models were developed mainly at banks, under the impetus of newly introduced regulations, to solve this complexity. These are portfolio-level models and different to the models discussed earlier as they aggregate the modelling outputs of individual credit models to the portfolio level. Two types of value-at-risk models have emerged: default mode models and mark–to–market models. In the first model, losses occur only when the credit defaults; in other words, it considers a binary outcome. Mark-to-market models consider a wider variety of outcomes, namely the transitions between different rating categories of the borrowers in the portfolio. Losses thus arise whenever the borrower's creditworthiness deteriorates or the correlations change between borrowers.

Two well-known credit value-at-risk models are CreditMetrics and CreditRisk+. CreditMetrics is a methodology developed by JP Morgan (Gupton et al., 1997). The model aims to arrive at a portfolio value at risk by taking as inputs the user's portfolio, credit rating transition matrices, present credit spreads of individual credits, recovery rates for each credit, and modelled correlation changes between the credits. In so doing, it looks to capture the diversification benefits of a portfolio and recognise concentration risks that are not easy to capture with firm-level default prediction models. A structural model is generally used to determine the transition matrices for each rating to help calculate an individual firm's credit risk. Given a one-year horizon, possible credit prices are calculated and discounted to the present time. Correlations across the credits are used to calculate the portfolio's market risk, and the difference is credit value at risk. These models are deployed at a high level of decision-making to make efficient use of capital and for regulatory reporting to understand the whole risk of a bank.

CreditRisk+, developed by Credit Suisse Boston, focuses on default as the main factor influencing the portfolio's credit risk (Giese, 2003). It does not use any structural model assumptions, and models default as an individual Poisson process, similar to reduced form models. It considers information relating to an exposure's size and maturity, as well as the credit quality and systematic risk of an obligor. The inputs to the model are the credit exposures, borrower default rates, volatilities and recovery rates. The model's output can be used to determine the level of economic capital required to cover the risk of unexpected credit default losses. Given the minimal inputs, standard default rates, and sector information, the model is easily scalable to large portfolios and computationally efficient. The default rate process is independently modelled using macroeconomic factors, making it more responsive to market changes.

1.2.2 State of the industry

Rating agencies are essential in credit markets for firms, just as credit bureaus like Experian, Equifax and TransUnion play a similar role for individual consumers. The impact of the rating firms on the wider industry has been well studied (Frost, 2007; Agarwal et al., 2016; Camanho et al., 2022). At a sovereign level, the actions of rating agencies have been scrutinised in public. For corporates, they have been criticised for over-generous ratings that fuelled the financial crisis.

The main output provided by credit rating agencies is a credit rating scale, denoted by alphanumeric characters, reflecting the potential risk of a borrower. Each such rating could then be mapped to the probability of default over a medium term of 3 to 5 years. However, the same agencies have largely been unable to cater to the SME segment of the market, as they use qualitative and quantitative benchmarks to arrive at a rating. This process is not easily scalable and is limited to the large corporations or sovereign nations that can afford this process. In our thesis, we look at developing a comprehensive model to eliminate qualitative criteria that are generally related to market conditions by incorporating various data sources and allowing the model to learn these attributes.

1.2.3 State of the regulations

The need to regulate financial institutions increased as credit markets became more complex and started to pose a systemic risk to the global economy. With the failure of a few institutions well documented, the need for global regulation of credit risk became essential. Basel I first standardised the risk weights for different classes of credit, as, previously, every bank was modelling the risk differently. The Basel II accord fine-tuned these risk weights while allowing more complex models to be built, borrowing in particular from value-at-risk models. It allowed for internal ratings-based or standardised approaches (the latter of which penalised banks for not developing more sophisticated models). The Basel III regulations developed since the financial crisis impose increased capital requirements and also regulate the overall leverage and minimum liquidity that needs to be maintained within each bank.

Credit risk management is not just about adjusting returns for the risk taken but also relates to several factors, such as identifying inherent risk, creating tolerances, limits and adequate credit controls. Firstly, a credit-granting strategy needs to be developed. This implies modelling credit risks to understand each firm's credit status before any credit is granted. Furthermore, appropriate credit monitoring tools must be designed to track individual credit exposures. These could again involve models that capture behavioural information. Additional models may be developed to understand concentration risks inherent in the portfolio, and appropriate provisions must be made. Not optimising on any of these metrics would lead to costly capital provisions. An internal rating system is necessary to comply with the Basel Accords and deploy the capital economically to generate better risk-adjusted returns.

Given all these requirements and considering some of the problems highlighted earlier, it may be costly for banks to develop even more sophisticated models for SMEs. Therefore, in our thesis, we set out to develop methods that external institutions could also explore to model some of the risk in SME credit portfolios. Using advanced computational methods to increase the precision of these risk estimates could also decrease the rejection rates for these firms as well as increase the profitability of institutions engaging in SME lending.

1.3 Deep learning models

This thesis's main contributions lie in applying advanced deep learning models to large-scale credit risk management problems. Whereas hitherto techniques in this domain used to be limited by computing power, as computing availability increased drastically, credit risk techniques have yet to capitalise on these improvements fully. Moreover, the types of data sources have also become more diverse, and many alternative data sets are now available that do not just consist of tabular, numeric data. However, they are typically associated with higher model complexity, and data processing for such large alternative data sets is again not straightforward.

Seeking to leverage these computational advances, deep learning models have produced state-of-the-art results in several domains and created groundbreaking applications in several industries. In the research presented in the thesis, we look to apply them to credit risk management problems, showing what adaptions need to be made to make recent methods in this field suitable to each setting and improve the predictive power of the models or their risk management properties.

In this section, we will first introduce deep learning methods and trace the history of some of the earlier studies. We will then discuss their application to different data types, thereby indicating how we may adapt them to our own data sources. Where possible, we will also discuss the explainability of these models.

1.3.1 Advances in computational methods

Being part of the artificial intelligence branch of computational studies, deep learning is a form of computational learning of concepts from the data using successive layers. For example, the model is trained on the data in supervised learning settings to detect relationships between the desired output and the input. This is achieved through developing a hierarchy of concepts or data representations. New concepts or relationships are learnt as the model processes the data from one layer to another. The layers closest to the input data tend to capture lower-level concepts, such as boundaries in images, while the layers closer to the output capture higher-level concepts, such as whether a face is embedded in the image. As these layers learn through the data that is served, no formal knowledge discovery is required. We delve into a brief history and relevant foundational algorithms that make this possible and link this to the relevance of modern architectures.

Most problems we encounter are in a form where the exact relationship is unknown, so we must approximate it. In the most simplest form,

$$Y = F(X;\theta). \tag{1.1}$$

Equation 1.1 shows the basic form whereby we are interested in finding the output Y using the input features X, and we have to determine $F(\cdot)$. In the credit risk literature, Y could be a binary variable indicating default or in portfolio optimisation. It could be a vector of asset weights. X are a set of features that we, as modellers, believe may be relevant to the output we need. θ is the set of parameters that can define $F(\cdot)$. Hence, our task is to find values for these and thus model the relationship between inputs and output. In this, deep learning models serve the same purpose as traditional models. In non-deep learning methods, however, certain assumptions and constraints are imposed on the $F(\cdot)$; for example, a linear relationship is assumed when applying linear regression methods. This form may be useful when the data is limited or only certain data features are selected. Most traditional models also assume a certain distribution of the data, which makes calculating first and second-order moments in the data tractable and, hence, builds an approximate relationship. Deep learning methods, in contrast, are more flexible and do not require making as many simplifying assumptions such as linearity.

This research focuses on two types of deep learning models: sequential and graph neural network models. Sequential models were first designed for natural language problems such as language translation, sentiment analysis and vision problems, but they have a natural application in time series data, too, such as that used for market analysis and credit risk modelling. In particular, self-attention models like transformers, originally proposed for language translation tasks, produce state-of-the-art results and generalise well to other sequential problems (Vaswani et al., 2017). Secondly, the interest in graph neural networks has also rapidly grown in recent years due to their ability to exploit the topology of the data to produce state-of-the-art results in network analysis problems (Scarselli et al., 2009). They are, for example, suitable to understand the system dynamics of financial networks. In the papers that make up this thesis, we aim to use these models as a base model and extend them to capture complex risk structures.

1.3.2 History of deep learning

Deep learning, a subset of machine learning, has attracted considerable attention in the last decade, significantly influencing a wide array of domains. Marked by several key milestones, technological advancements, and a series of breakthroughs, its computational techniques, which mimic human learning, have permeated fields such as healthcare, finance, and technology.

The perceptron was the first neural network model, developed by Rosenblatt (1958) based on earlier work by McCulloch and Pitts (1943). This model proposed a single perceptron which combines the inputs through weight parameters and has an activation function. However, it could solve only linearly separable problems. The next significant breakthrough came with the development of the backpropagation algorithm (Rumelhart et al., 1986). This addressed the limitations of the simple perceptron by enabling efficient training of multi-layer neural networks. The algorithm uses gradient descent, which adjusts the weight parameters in each model layer and minimises the error by propagating it backwards through the layers.

The 1990s saw the development of architectures being built for different data types to solve particular problems. Convolutional networks were developed by Yann LeCun, with the first version of LeNet demonstrating the effectiveness of convolutional neural networks for image recognition tasks, particularly handwritten digit classification (Le Cun et al., 1990; Lecun et al., 1998). They introduced the concepts of local receptive fields or convolutions, shared weights, and spatial subsampling (pooling). In parallel, Recurrent Neural Networks (RNNs) were introduced by Elman (1990) for sequential tasks such as language modelling, speech recognition and language translation. They were further extended by Hochreiter and Schmidhuber (1997) who developed Long Short-Term Memory (LSTM) networks. Whereas initial RNNs had the problem of vanishing gradients due to the earliest layers' weight parameters being very slow to update, this was resolved largely with LSTM models.

In the next decade, Hinton et al. (2006) introduced Deep Belief Networks (DBNs), which is a generative model that learns to mimic the input data so that new synthetic data can be generated. For a specialised task, this generative model is fine-tuned to produce the required outputs. This utilises initially unsupervised pre-training, followed by supervised fine-tuning for specific tasks such as classification. This work created a resurgence in interest in deep learning as it showed the flexibility of different training regimes on the same model.

In 2012, AlexNet won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by a large margin, marking a significant resurgence of CNNs (Krizhevsky et al., 2012). AlexNet incorporated many novel features at the time, which are common now. The first model made efficient use of Graphical Processing Units (GPU) for training the model, which reduced the training times drastically while also increasing the complexity of the model in terms of the number of parameters that could be trained. It also incorporated different activations such as ReLU (short for Rectified Linear Unit), and for regularisation, included dropout layers. GPU-based acceleration introduced so many improvements that larger models continued to be built with AlexNet, containing up to 62 million parameters (compared to 60 thousand in LeNet models earlier). Generative Adversarial Networks, introduced by Goodfellow et al. (2014), revolutionised the generative modelling ability of deep learning models, which laid the foundations for the realistic image and video generation that are commonplace today with the latest diffusion models. The next significant breakthrough came in the form of the transformer architecture, introduced by (Vaswani et al., 2017). This eschewed recurrence in favour of self-attention mechanisms, leading to more efficient training and superior performance on various Natural Language Processing (NLP) tasks. These transformer architectures have become the main models for the deep learning progress made in recent years.

Deep learning models have enabled state-of-the-art systems that are also widely used, such as Google's Assistant, Apple's Siri or Amazon's Alexa. They are commonly deployed in some form in the recommendation systems we encounter in online e-commerce or music streaming platforms. Computer vision-related deep learning has enabled major advancements in object detection, segmentation and recognition, encapsulated by systems like YOLO (You Only Look Once) (Redmon, 2016). Similarly, the NLP field has been transformed with models like BERT (Devlin et al., 2018), GPT (Generative Pre-trained Transformer) (Radford et al., 2018), and T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2019), which have pushed the boundaries of language understanding, translation, summarisation, and question answering.

There are still many active areas of development in deep learning, as further challenges need to be overcome. Importantly, deep learning models are often treated as "black boxes", making it hard to understand their decision-making process. This brings the interpretability of these models' predictions into question. Work in progress on explainable AI (XAI) techniques aims to provide insights into model predictions (Barredo Arrieta et al., 2020). Another challenge is linked to the realisation that deep learning models are computationally intensive and require significant resources, notably as the models increased in complexity over the past few years. Research into model compression, efficient architectures, and pruning are areas of further interest in this space. A final example of an area of active interest is the development of fair models and establishment of ethical guidelines for AI application development as well as AI research (Cheng et al., 2020; Liang et al., 2021). For instance, a key concern here is that models are shown to inherit bias from the training data, which can lead to unfair outcomes for sections of the population (Sahin et al., 2024).

1.3.3 Data inputs

Deep learning has proven especially successful in utilising new (non-traditional) data sources. Historically, many of these alternative data sources were not amenable to quantitative analysis. With the latest developments, though, they have become a powerful source of information. In various application areas, the growing amount of high-dimensional data, combined with an increase in computation power and new learning algorithms (i.e. deep learning), are thus allowing data scientists to solve complex problems. They are, for example, being applied in image and speech recognition, reconstructing brain circuits, new drug molecule discovery, natural language understanding and many other fields (LeCun et al., 2015).

Here, we briefly review different types of data, including alternatives to traditional tabular data, and identify the deep learning methods designed to handle them.

1.3.3.1 Images

The selection and handling of image data types are often critical in designing robust and efficient deep learning models. The choice of image data type can affect everything from preprocessing options to the architecture and performance of the model. There are several ways in which every pixel in an image can be encoded (Nixon and Aguado, 2019). One common type are binary images, where each pixel in an image is 1 (white) or 0 (black). This is ideal for basic segmentation tasks and is helpful in optical character recognition and in medical imaging to highlight specific structures. Grayscale images represent intensity values where colour information is not necessary, but detailed image information is needed for tasks such as facial recognition or texture analysis. Colour images consist of multiple channels, usually three (Red, Green and Blue, in short RGB), each of which is similar to the earlier grayscale encoding. RGB images are used more broadly for image classification and object detection. More advanced versions can add multiple channels, such as multi-spectral images or depth images, which capture data across multiple wavelength bands and distance information from the viewpoints. These are used in remote sensing, agriculture, autonomous driving, and various other applications that require an understanding of spatial relationships. These image data are often further processed by applying normalisation, which scales the values to a defined range, usually [-1, 1]. This helps train the model with better convergence and stability. The image is also resized to fit the model input requirements. Techniques like rotation, scaling, and flipping can provide the model with diverse training datasets. This also helps the model be robust and generalisable. In the first paper, we also employ the concept of channels but instead of image data, whose channels tend to have the same dimensions, we have financial panel data in which the number of dimensions differs per channel. Hence, we tweak the models to adjust for these differing dimensions. The complexity comes in with the model architecture and training regimes.

CNNs have been the core models to process image data types. They are designed to automatically and adaptively learn spatial hierarchies of features from input images. The basic structure consists of a convolutional layer, activation and pooling layers. The convolutional layer applies a set of learnable filters to the input image to produce multiple feature maps. This is done by computing the dot product between the filter and the image sections. A non-linear activation function is applied on these feature maps. ReLU (Rectified Linear Unit), i.e. $f(x) = \max(0, x)$, is most commonly used, but Sigmoid and Tanh activations are also possible options. In this thesis, we used the ReLU activation function. For a survey of various activation functions, we refer the reader to Apicella et al. (2021). Finally, the pooling layers downsample the dimensions of the inputs and reduce the spatial size. This is done by either max or average pooling, whereby the maximum or average values in the filter map are used, respectively (Gholamalinezhad and Khosravi, 2020). This process is often repeated a few times to allow learning of different aspects of the data. The model typically will have fully connected layers where the convolutional feature maps are flattened, and activation functions such as softmax or sigmoid layers allow the model to be used for classification or other tasks. Some of the foundational models in this space are: LeNet-5, which uses two sets of convolutional and pooling layers and one fully connected layer; AlexNet, which has five convolutional and pooling layers and three fully connected layers using the ReLU function; VGGNet, a simplified version which performed better; the Inception models, which utilised multiple convolutional filters sizes to produce efficient spatial hierarchies; and, finally, ResNet which introduced skip connections or residual blocks, both of which enable gradients to flow directly thus enabling very deep networks without any performance degradation (He et al., 2015). CNNs' ability to automatically and adaptively learn spatial hierarchies of features makes them extremely powerful for image recognition. Transformer models are also being increasingly applied in this space (Han et al., 2023).

1.3.3.2 Sequential data

Sequential data is characterised by dependencies between observations in time or space. The observations are ordered and dependent on previous time steps. This is in the form of multivariate time series (e.g. stock prices over time of different stocks), panel data (e.g. financial statements over time across multiple firms) or language (e.g. words in a sentence). This form of data is pervasive across many domains, including NLP, finance, healthcare, and more. Another common feature of this kind of data are variable-length sequences, which can vary in the history and depth of each sequence, requiring models that can handle such variance. The context is important here, so the meaning of neighbourhood and current data points depends on the most recent historical data. This means the models require memory and methods to pass essential turning points in the series. Data with time as one of the dimensions is what we mostly encounter in this thesis, as we dealt with large sets of firms observed over extended historical periods to arrive at predictions for these firms.

Deep learning models designed to handle sequential data commonly excel in extracting temporal or spatial patterns, making them indispensable for speech recognition, language translation, and time-series forecasting tasks. LSTM models are the foundational models that can learn these long-term dependencies and can handle variable-length sequences. Gated Recurrent Units (GRUs) are a simple alternative to the LSTM models, with fewer gates and parameters, providing similar performance and reduced computational complexity. Transformer models have been the breakthrough models in this area, processing entire sequences in parallel rather than sequentially, leading to huge improvements in sequential data tasks (Vaswani et al., 2017). They are superior in handling long-range dependencies, as well as highly parallelisable, leading to reduced training times. Their use of attention mechanisms allows the models to focus on different parts of the input sequence when creating the outputs. These have found applications in NLP tasks such as text classification or sentiment analysis (in which models learn to categorise text and assess its sentiment). Furthermore, compared to earlier models, the transformer model improved language translation and speech recognition capabilities to a degree that is closer to human-level performance. In the area of time-series analysis, they have been integrated in weather forecasting and anomaly detection (Mousavi et al., 2020; Wen et al., 2022). They also improved the areas of healthcare and genomics in predicting newer protein structures and gene expression and can detect anomalies much faster from medical time series (Chandra et al., 2023).

Transfer learning is widely used with these models, as pre-trained models, previously trained on large tasks, can be fine-tuned for specific domain tasks. For example, models such as BERT, trained on general language datasets, have been fine-tuned for NLP tasks in the healthcare or finance domains. This cuts the training cost for practitioners by a large margin and helps explain the surge in the adoption of these models. The performance achieved across such a variety of domains shows the general learning ability of these types of models. Nonetheless, understanding different models' nuances, strengths, and appropriate applications remains key to leveraging deep learning for sequential data tasks.

1.3.3.3 Tabular data

Tabular data is one of the common data types encountered in many domains. It consists of rows and columns, where each row represents an observation and each column represents a feature or attribute. While deep learning has traditionally excelled in handling unstructured data such as images, text, and audio, recent advancements have made it increasingly effective for tabular data.

Tabular data can comprise different, heterogeneous features where columns could be of a different data type. It can thus contain a mix of numerical, categorical and ordinal features. A database of firms, for example, might include features such as revenues (numerical), sector (categorical) and market capitalisation range (ordinal). Often, tabular data also contains missing values that need to be addressed. It is common as well to see the features being related or interacting with each other. These patterns could often be complex and require sophisticated modelling techniques to be captured effectively.

Preprocessing and feature engineering are required to prepare these data as inputs to the models. Normalisation and standardisation are techniques for scaling the features to a specific range or distribution. This ensures uniformity and improves convergence during training. Min-Max scaling, which transforms features to the [0,1] range, and standardisation, whereby features are rescaled to a mean of zero and a standard deviation of one, are common techniques. Categorical variables need to be encoded, typically using one-hot encoding, which expands the categorical variable into several binary features. Design decisions may be taken when multiple categorical values are possible for a single feature, which may involve grouping them first and then one-hot encoding them.

Missing values can, for example, be addressed by imputing the mean, median or mode. Outlier features can affect some of these, but they may also be winsorised. Simply removing rows or columns with missing values can lead to data loss; to minimise this risk, we can impute the missing value but also create a binary categorical feature for each feature to represent whether the data was replaced. This technique we have used throughout the thesis.

Fully Connected Networks (FCNs) are the simplest neural network architectures that deal with tabular data, and they have been shown to work well for regression and classification tasks on structured data. Wide and deep learning techniques where the wide component captures feature interactions and deep components capture complex patterns have been deployed in recommendation systems and predictive analytics (Cheng et al., 2016). Further use cases have been used to generate embeddings that can be used in the models, thus reducing the dimensionality and capturing semantic relationships.

Traditional models that assume linear relationships might not always work well for tabular data, but deep learning models have not been as successful here as on other data types (Gorishniy et al., 2021; Shwartz-Ziv and Armon, 2022). In contrast, tree-based ensemble models such as random forests and XGBoost are consistently found to perform very well in this context (Grinsztajn et al., 2022). The latest deep learning model developments are reporting higher performance, but there is no comprehensive evidence yet that deep learning models will outperform other models like we often see with other forms of data (Hollmann et al., 2025). Hence, while deep learning has for some time been dominant in unstructured data domains, its application to tabular data is still evolving.

1.3.3.4 Graph data

Graphs or network data are structures used to represent relationships among entities. They appear in numerous domains, including social networks, biological networks, transportation systems, and the Internet. Graphs consist of nodes (vertices) representing entities and edges (links) representing relationships between these entities. For example, they could connect users in a social network or firms in a relationship network. The graph structure varies with each case. Depending on the setting, the edges can be undirected, directional or bidirectional. The edges can also have weights, signifying the strength of the relationship. In many graphs, the nodes and edges also have attributes, which give a set of feature vectors. For a user in a social network, for example, the node attributes could be profile information, whilst edge attributes could include the number of interactions and the kind of interactions between friends.

There are many techniques to represent the graphs. Especially for larger-scale graphs, efficiently representing them becomes important as the computational complexity increases. An adjacency matrix is a square matrix used to represent a finite graph. Each matrix element indicates whether there is an edge between the corresponding nodes. The edge list is another form of representation in which each edge is represented as a tuple. Node and edge attributes could be converted to low-dimensional vector representations using different techniques while preserving the graph topology and node features. DeepWalk applies random walks to generate node sequences and then uses skip-gram to learn embeddings (Perozzi et al., 2014). Node2Vec extends this model by introducing bias to the random walks (Grover and Leskovec, 2016). These representations can be used as inputs to deep learning models that are trained to provide suitable outputs from them.

Deep learning models that were designed to handle graph data, often called Graph Neural Networks (GNNs) (Scarselli et al., 2009), have shown substantial promise in learning from these data structures. Graph Convolutional Networks (GCNs) extend traditional convolutional neural networks to non-Euclidean graph data (Bruna et al., 2014). Convolutions are applied over graph nodes, aggregating feature information from the node's neighbours. They have been used in node classification and community detection. Graph Attention Networks (GATs) use the attention mechanism to assign different weights to the nodes in a neighbourhood, focusing on the most relevant nodes during aggregation. These are effective in scenarios where the importance of neighbouring nodes varies. For example, firms in a network can be influenced more by the major firms' financial health than some of its other neighbours. GraphSAGE (Graph Sample and Aggregation) generates node embeddings for previously unseen data using inductive learning, aggregating features from a sample of a node's neighbours. They are used in large-scale graphs and dynamic graphs where nodes or edges change over time.

GNNs have been deployed in various domains. In social network analysis, they are used for link prediction (for example, a suggestion for adding friends) and community detection (identifying groups of nodes with similar properties or interests) (Fan et al., 2019). In recommendation systems, they may be applied for collaborative filtering and content-based filtering to recommend similar items to users (Ying et al., 2018). They are used in fraud detection to identify malicious behaviour in social networks or fraudulent transactions in financial networks (Xu et al., 2021). In transport systems, they may be used for predicting traffic or for route optimisation (Peng et al., 2020). In many of these settings, GNNs face key challenges regarding scalability and the dynamic nature of the real world. Hence, techniques such as sampling, efficient sparse matrix operations, and dynamic embeddings constitute areas of further research.

Since this thesis uses several of the data types discussed in the previous sections, ranging from panel data to graph data, we will next discuss how to integrate and learn from these different data modalities.

1.3.4 Multimodal models

Multimodal learning involves integrating and processing data from multiple modalities, such as text, image, audio, video, and sensor data (Ngiam et al., 2011). This approach leverages the complementary information from different data types to enhance the model's performance and robustness. Multimodal architectures have found applications in a variety of domains, including NLP, computer vision, healthcare, and human-computer interaction. For a full review and applications of these models, we refer to Jabeen et al. (2023).

Different modalities possess different properties and structures (e.g., the text is sequential, images are spatial, and audio is temporal), but they also provide complementary information, which enhances the overall understanding and context. As an example, combining visual and textual information helps to understand an image caption better. However, they bring their own challenges, which include the problem of dealing with redundant or noisy information. Hence, they require robust mechanisms to filter and integrate useful data.

There are several fusion techniques in multimodal learning (Sahu and Vechtomova, 2021; Pawłowski et al., 2023). Early fusion combines raw data from different modalities at the initial stage before feeding it into the model. This is the traditional method and easy to implement. However, it might cause information loss as one channel might dominate the other. Late fusion combines the outputs only at the final level while keeping the modalities separately trained until then. This, however, loses any interactions that could be learnt. Hybrid fusion combines both approaches by combining intermediate representations and then further training together.

Deep learning-based multimodal architectures are widely being developed. Multimodal Deep Boltzmann Machines (MDBMs) extend the earlier DBM by using separate layers for each modality and then using a shared representational layer (Srivastava and Salakhutdinov, 2012). Multimodal autoencoders combine separate encoder networks to generate latent representations, which are then decoded into the original modalities. These effectively denoise the data (Zhu et al., 2019) and image-text alignment tasks (Liu et al., 2023). Multimodal transformers are also quite effective as the attention mechanism is applied across different modalities and used for various audio-visual and text tasks (Xu et al., 2023). These have been used in tasks such as automatic image captioning, visual question answering, and cross-modal retrieval such as image search via text.

The aforementioned models present their own challenges in properly aligning and synchronising data from different modalities with varying temporal and spatial resolutions. They increase the computational requirements and memory footprint, which calls for efficient architectures and parallel processing techniques. Notwithstanding these challenges, ongoing research and advances in multimodal learning promise to further expand its applications to complex real-world problems, enabling more efficient, accurate and interpretable models.

1.3.5 Loss functions

Loss functions, or objective or cost functions, are critical components in neural networks and deep learning algorithms. They need to be differentiable for the gradient descent and backward propagation algorithms that deep learning models rely on in their optimisation, to work. In supervised learning, they generally quantify the difference between the predicted and actual output in absolute or probabilistic terms. The change in the loss value for each training batch guides the optimisation process on how to update the model's parameters. Selecting an appropriate loss function is crucial, as it directly impacts the model's performance and convergence. A number of loss functions are commonly used, depending on whether the problem is classification or regression.

For classification tasks, the cross entropy loss (CEL) is the most common choice, which is a log-likelihood function. This is defined for each observation i and summed to obtain a batch-level loss or full sample loss. Thus, for observation i,

$$CEL_i = -\frac{1}{C}\sum_{c=1}^{C}(y_{i,c}\log(P(\hat{y}_{i,c})) \tag{1.2}$$

Where C is the number of classes, y_{ic} is a binary variable indicating whether the final output belongs to class c, and $P(\hat{y_{ic}})$ is the predicted probability for that class. In this thesis, we deal with multi-label classification and binary classification tasks. For the former, we tweaked this cross-entropy loss to make it suitable for multi-label classification. In a multi-label problem, the same observation could be in multiple classes, unlike regular classification, where each observation could belong to only one class. For binary classification, the loss function is commonly expressed as a sum, rather than a mean, as shown below.

$$CEL_i = -(y_i\log(P(\hat{y}_i)) + (1-y_i)\log(1-P(\hat{y}_i))) \tag{1.3}$$

For regression tasks, Mean Absolute Error (MAE) or Mean Squared Error (MSE) are commonly used. These are also called $L1$ and $L2$ loss, respectively.

Mean Absolute Error (MAE), as defined below, measures the average magnitude of the absolute differences between actual and predicted values over a sample of size n. This is useful when the presence of outliers is less severe or when we want to treat all deviations equally.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1.4}$$

Mean Squared Error (MSE) is the average of the squared differences between actual and predicted values:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{1.5}$$

These could also be used as optimising criteria within specific parts of the deep learning model to restrict the outputs of intermediate layers, which could help improve the overall model's performance.

1.3.6 Performance criteria

Performance criteria measure how well the models have performed over the data. The same functions discussed earlier as loss functions are typically used for regression tasks, in addition to R^2 or Adjusted-R^2. Other criteria are needed for classification tasks depending on the data and predicted outcome distribution. Most performance criteria can be derived from the confusion matrix, shown in Table 1.1.

TABLE 1.1: Confusion matrix for classification tasks

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

For each observation, a classification model either correctly classifies it, resulting in a true positive (TP) or true negative (TN), or it incorrectly classifies it, which results in a false positive (FP) or false negative (FN). We can create suitable performance criteria by taking the count of observations in each of these categories.

The accuracy of the models is widely used when the classes are balanced and can be defined from the confusion matrix as

$$Accuracy = (TP + TN)/(TP + FP + TN + FN). \tag{1.6}$$

Furthermore, we can define:

$$Precision = TP/(TP + FP) \tag{1.7}$$

$$Recall = TP/(TP + FN) \tag{1.8}$$

$$F1score = 2 \times Precision \times Recall/(Precision + Recall) \tag{1.9}$$

Recall is also called the true positive rate (TPR), whilst the false positive rate (FPR) is $1 - TPR$. These values are used to plot the Receiver Operating Characteristic (ROC) curve across all thresholds of TPR and FPR. The area under this curve is called the AUCROC, which is preferred over accuracy in classification tasks where there is a class imbalance in the data set. This is when one of the class frequencies is considerably lower than the other classes, as it is regularly encountered in credit risk modelling settings where loan defaults tend to be much rarer than non-defaults.

In addition to these widely used metrics, customised metrics could be developed depending on the problem, which might also be suitable as a loss function to optimise the model. In portfolio optimisation, for example, typically, the Sharpe ratio is the metric used, which is the average expected return of the portfolio over the risk-free rate relative to the volatility. We will further discuss this in the relevant chapter.

1.3.7 Applications of deep learning in finance

Machine learning models have a longer history in finance, having seen various applications such as using support vector machines in classification (Kim and Sohn, 2010) or K-means clustering algorithms. We refer to the comprehensive review by Nazareth and Ramana Reddy (2023) for machine learning applications in finance. The recent advances in AI and specifically deep learning models have also found applications in finance due to their predictive power, ability to extract relationships from different data types, and generative abilities (for example, when having to summarise documents).

They have been broadly studied in the finance domain in the following areas.

- **Risk management**: Deep learning can produce predictions about risk that could be extracted from existing data sets with better accuracy or from novel high-dimensional data sets. This thesis is also a prime focus as it seeks to predict the default of mid-cap companies and SMEs. These predictions could be further used to derive capital provisions in a bank or for investing decisions at other financial institutions. Other work has used deep learning techniques to analyse textual data contained in company filings (or other alternative data sets) (Mai et al., 2019; Stevenson et al., 2021) and detect sentiment or derive other numeric features that can be fed into the risk management models (Glorot et al., 2011; Wang et al., 2016).

- **Trading strategies**: Deep learning models have been used to create automated strategies as they have been found to be good at predicting the best-performing assets (Nelson et al., 2017; Jiang et al., 2024), or at detecting change points or regime switches much earlier than other models (Wood et al., 2022). They also have been used for hedging strategies (Buehler et al., 2019).

- **Portfolio optimisation**: Here, deep learning methods may be employed to find the best assets to invest in and how much of the available capital to allocate to them while considering criteria such as minimising portfolio losses or drawdowns (Hu and Lin, 2019; Zhang et al., 2020). This forms the problem setting for the second paper of the thesis.

- **Generative data**: Deep learning models have also been used to simulate market environments, generating data that could further aid model training (Wiese et al., 2020). Another application has been in setting up workflows mimicking an expert using large language models which can summarise large amounts of text, such as legal documents (Watson et al., 2024; Singh et al., 2024).

In this thesis, we research two of these areas — risk management and portfolio optimisation. Within risk management, we focus on credit risk, which constitutes the largest risk for banks. Deep learning models have attracted growing attention in the credit risk literature, both in individual (consumer) lending and corporate credit. They have been mainly studied there for default prediction or credit scoring, classifying borrowers based on their risk grade, or detecting fraud. With the increased availability of large datasets such as graphs and networks, deep learning models are being trained on a variety of input data to improve the performance of standard models. They also serve to understand the value of different alternative datasets being available. This allows institutions to decide on which data collection procedure to prioritise putting in place.

In credit risk management, explainability is key, particularly where added regulatory scrutiny applies. Deep learning models have limitations in this regard. Research is ongoing to improve them for more general use cases and credit risk management. In the thesis, we considered some of these developments and adapted them to our areas of enquiry. Hence, the following section will briefly elaborate on some relevant work here and how it applies to our work.

1.3.8 Interpretability

Being able to interpret or understand the predictions of the models is often a precondition to proceed with deploying the models in real-life scenarios. However, deep learning models are complex and are thought of generally as black-box models, especially compared to more traditional models (Sahin et al., 2024). The same properties that allow them to work on large-scale data, especially those with high dimensionality or different data modalities, bring out challenges in interpreting them. Many studies in this area have been on post-hoc explainability, where we try to understand the models' predictions after the training has happened (Vale et al., 2022). Such explanations could be model-agnostic or model-specific. Whereas the former type can be used with any type of model, the latter are useful only for a particular architecture. In this thesis, we consider both kinds of explanations to better understand the models and the risk drivers they identify.

The model-agnostic methods used in part of our work involve a form of Shapley Additive exPlanations (SHAP) analysis (Lundberg and Lee, 2017), which has been extended to answer specific questions about the relative importance of different data sources and around which historical period most influences the predictions. Doing so requires grouping the features, whilst the results also inform our choice of architecture.

Having now set out the application context of our research, and the deep learning advances that underlie our proposed methods, the next section will elaborate on the objectives of the thesis and the research questions addressed by the three papers.

1.4 Thesis objectives

The broad aim of the thesis is to advance the area of credit risk management using deep learning approaches, thereby focusing on SMEs and listed mid-cap firms. Small improvements in these risk management techniques can lead to many benefits, such as improved access to credit markets for SMEs, whilst the research may equip investors with cutting-edge tools to improve decision processes, as well as make a contribution towards making the credit markets more robust to different macroeconomic environments.

Specifically, the three papers that make up the thesis will address the following research questions.

The first paper looks to design and test a transformer-based deep learning model for predicting the probability of default for mid-cap firms over the short to medium term. One of our starting premises is that, by considering the interaction between credit and market risk, we should be better able to model the credit risk of these or other companies with limited liquidity. Hence, we will incorporate market-related signals into the credit risk models, thus combining the types of models used in consumer lending with those in corporate credit risk modelling. To do so, we will need the capability to model complex interactions inherent to deep learning models. Among several challenges related to these types of firms, we also seek to address the difficulties in producing longer-term predictions, which present models cannot handle. Furthermore, we look to contribute to this strand of research by creating not only novel deep learning techniques but also providing interpretations for the predictions of these models. In so doing, we also seek to answer some of the questions raised in the credit risk literature and provide regulatory insight.

The second paper moves away from firm-level models to understanding the system-wide dynamics of a portfolio of mid-cap firms. Its first objective is to investigate how to create a network of correlated firms from such a large universe of firms with sparse historical data. Secondly, we look to use this topology to construct an optimal investment portfolio (in terms of volatility-adjusted returns), using graph neural network models. In designing our approach, we ensure that companies that default sometime over the study's timespan do not have to be omitted; we explore why other studies do and make the case they should be included. To assess the effectiveness of our proposed graph neural network solution, we compare the risk-adjusted returns of the resulting portfolios with those generated by several established approaches and aim to understand the differences in their respective strategies. We also study the robustness of the model performance over long periods.

In the third and final paper, network models and multimodal learning are applied to a large dataset from a major international lender containing information on SMEs and their revolving credit lines. From this, we derive a series of behavioural features that we include in our default prediction models. We then aim to study whether adding network data about other related firms can improve the model's predictive capability. We thus aim to quantify the relative value of the network data for this prediction and also explain the models by identifying the most important behavioural features in the dataset. This paper thus combines some of the work from the previous two papers while also analysing behavioural data, which is less well studied in the literature. As with the first paper, the ultimate aim is that any improvements may improve capital access for SMEs and/or improve profitability for SME lenders.

1.5 Paper contributions

The main contributions from the first paper on mid-cap corporate default prediction are as follows. First, we developed a novel deep learning approach to credit risk modelling by applying an advanced transformer-based model to time series panel data. This approach also features a custom loss function and a performance metric specific to the term structure problem. Second, we put forward a framework for multimodal learning that combines the different data sources and allows for a differential training approach to train each model separately. Third, we identified a suitable method to interpret the predictions of the deep learning model. Specifically, to establish which data sources and time periods contribute most towards default prediction, we use a custom method that uses Shapley values to quantify the relative importance of groups of variables. Finally, another contribution is to the credit risk literature, as we show how a multi-horizon probability of default model can be built within a single model, taking advantage of deep learning techniques, and how this model produces good results not just in the short-term but over a medium term of three years.

Our main contributions in the second paper are on portfolio optimisation using graph neural networks. First, we developed topological portfolio optimisation models for mid-cap firms, extending the literature that had hitherto focused mostly on stochastic models for the segment. Second, we applied advanced dependence measures to a large set of assets to generate networks that capture the complex relationships that occur between the return data for different mid-caps. Finally, we showed how these networks can be fed to graph deep learning models to produce portfolio weights and compared their performance against traditional approaches. We developed custom layers that give better training consistency (smoother gradient descent), and we used the whole deep learning architecture for optimisation. We also explained the selection and capital allocation strategies and showed that the deep learning models are robust over time and, most of the time, outperform other methods whilst requiring fewer transactions. We also contributed to the literature on large-scale portfolio optimisation by using over 5000 firms and 30 years of data and including firms with high default risk.

The third and final paper extracted explicit networks from the data and generated behavioural data from a unique lending data source. We showed that the deep learning embeddings generated from the network data improve upon the behavioural predictions by utilising the state of the current network neighbours of a firm. We found that network information alone can also predict default and remains so during external shocks when the behavioural data as a whole could be affected and become less valuable. However, in a typical economic environment, behavioural data performs well. Using tabular data, we found that the performance uplift from deep learning models is insignificant. Still, once the data turns more complex, as with the graph data, the performance gap with benchmark models becomes much more sizeable, confirming earlier literature.

Overall, the thesis addressed the broader question of how deep learning may be successfully adapted to credit risk management, particularly in areas where there are specific data challenges, such as high dimensionality, different data modalities, or gaps in the data, that specific deep learning solutions are well-equipped to handle. It also highlighted the importance of network data in this domain and its role in different macroeconomic environments.

1.6 Structure of the thesis

The rest of the thesis is organised as follows. Chapter 2 contains the first paper on default prediction for mid-caps using transformer-based deep learning models. Chapter 3 contains the study on mid-cap portfolio optimisation using graph neural networks. Chapter 4 continues with the work on SME credit lines, using behavioural and network data for credit risk default prediction. Chapter 5 concludes the thesis, again highlighting the main contributions and discussing where there is scope for further research.

Chapter 2

A transformer-based model for default prediction in mid-cap corporate markets [1]

Abstract

In this paper, we study mid-cap companies, i.e. publicly traded companies with less than US$10 billion in market capitalisation. Using a large dataset of US mid-cap companies observed over 30 years, we look to predict the default probability term structure over the medium term and understand which data sources (i.e. fundamental, market or pricing data) contribute most to the default risk. Whereas existing methods typically require that data from different time periods are first aggregated and turned into cross-sectional features, we frame the problem as a multi-label time-series classification problem. We adapt transformer models, a state-of-the-art deep learning model emanating from the natural language processing domain, to the credit risk modelling setting. We also interpret the predictions of these models using attention heat maps. To optimise the model further, we present a custom loss function for multi-label classification and a novel multi-channel architecture with differential training that gives the model the ability to use all input data efficiently. Our results show the proposed deep learning architecture's superior performance, resulting in a 13% improvement in AUC (Area Under the receiver operating characteristic Curve) over traditional models. We also demonstrate how to produce an importance ranking for the different data sources and the temporal relationships using a Shapley approach specific to these models.

[1] NOTICE: This is a published work. Changes resulting from the publishing process, such as editing, corrections,structural formatting, and other quality control mechanisms may not be reflected in this version of the document. Please cite this work as follows: Korangi, K., Mues, C. and Bravo, C. (2022). A transformer-based model for default prediction in mid-cap corporate markets. European Journal of Operational Research, DOI: https://doi.org/10.1016/j.ejor.2022.10.032. This work is made available under a Creative Commons BY license.🅭🅯

Keywords:

OR in Banking, Mid-Cap Credit Risk, Default Prediction , Deep learning, Transformers, Self-Attention

2.1 Introduction

Traditional credit risk models cater to individual consumers with empirical models (built by applying statistical or machine learning methods to large datasets). In contrast, corporate credit risk models are often theory-driven or may include a qualitative component. Rating agencies play an important role in determining corporate credit risk. That rating process is costly, and it also has a strong subjective component (Frost, 2007; Rona-Tas and Hiss, 2010). The subjective component is often needed because, unlike with consumer credit risk models, the small number of firms may affect the quality of statistical models. Although this approach is appropriate for large companies, for the much larger population of small to medium-sized companies, such a qualitative assessment would not be scalable. Neither could we reapply the same quantitative approaches developed for consumer credit risk, as the default signal in the corporate setting comes from a complex combination of internal and external market conditions. In our work, we seek to both remove the subjective component of the rating process and take a different quantitative approach by incorporating various data sources such as accounting data, pricing data and general market data into a multi-channel deep learning model that predicts the default risk of mid-cap companies that are active in debt (bond or loan) markets.

Mid-cap firms (in short 'mid-caps') are defined as firms with USD 1 to 10 billion market capitalisation and are likely constituents of Dow Jones Wilshire Mid-cap index or S&P 400 Mid-cap index. Their debt has a shorter legal maturity period of around 5 to 10 years for mid-caps (over 20 years for large-caps). The effective maturity of the debt can be as short as half the legal maturity, after considering embedded options and coupon rates that tend to be higher than those for large-caps. Mid-caps also tend to differ from large caps in terms of the relative credit risk they pose. In corporate debt markets, the mid-caps typically hold a non-investment grade credit rating, implying higher credit risk. Given that the listed mid-cap companies provide public data about their financial accounts, stock exchanges publish stock prices, and default history is available, lenders have all the data required to construct sophisticated credit risk models. In this paper, we use a combination of financial accounting data, historical pricing data of the firm and general market performance data to predict the probability of default.

Despite the availability of such data, building such models presents several challenges. First, the credit spreads or prices implied by the models often differ from what is empirically observed, termed as the Credit Spread Puzzle by Amato and Remolona (2003). This means mid-cap credit risk is not accurately priced and can lead to underestimation of potential losses. A second challenge is the difficulty in separating credit risk and market risk for mid-cap firms (Jarrow and Turnbull, 2000). Finally, the covenants in debt offerings and embedded options make the maturity and capital structure dependent on market conditions (Liu et al., 2016). All these issues make it difficult for lenders or investors to assess risk on a large scale, thus limiting access to credit for the companies involved. To address this, governments have established supporting institutions providing financing to mid-caps and small and medium-sized enterprises such as the European Investment Bank (EIB) in Europe and the British Business Bank in the UK.

Another challenge in building corporate default prediction models lies in the time horizon of the prediction models. Most credit risk models study the probability of default over a one-year time horizon due to business practices and regulatory frameworks such as the Basel Accords (Basel Committee on Banking Supervision, 2003). However, the time between financial distress to an actual default could easily last longer in firms. In the capital requirement models cited above, this is reflected by the maturity component of debt, but they are not usually captured by the probability of default (PD) models. Several methods have been proposed to extend the models to longer horizons (Duffie et al., 2007; du Jardin, 2015; Altman et al., 2020). Still, multi-horizon models are not widely implemented due to the lack of large historical data under different macroeconomic conditions, changes in distribution of the variables, relationship drift between explanatory variables over time and the changes in relationship with the dependent variable (du Jardin and Séverin, 2012). Different models tend to be developed for different time horizons, and generally, an ensemble of models is used for better performance, making the modelling complex. We are interested in predicting the probability of default from a short-term horizon of several months to a medium-term horizon of one to three years, using a unified model. This is close to the effective maturity of these instruments and considers most lenders' investment horizons in this area of the market.

The techniques used for default prediction modelling have evolved over time and very much remain an active area of research (Dastile et al., 2020). Traditionally, popular linear models such as the logit model or discriminant analysis require making a large number of discretionary decisions when handcrafting a set of predictive features, such as the choice of lookback period and aggregation function, and making some restrictive assumptions about the distribution of the data or the functional form of the relationship between those features and default risk (e.g. linearity). In addition, large datasets may also require further feature selection (Jones et al., 2017). On the other hand, machine learning models allow for a large set of features and can handle non-linear relationships, which can produce predictive performance gains over linear models. However, integrating different kinds of (often diverse) data sources remains challenging as the process to represent data becomes complex (Mai et al., 2019). Such data could include non-structured data (such as text or audio) and may contain a mix of high-frequency data (such as daily price history) and low-frequency data (such as accounting information). Deep learning models (LeCun et al., 2015), however, can cope not only with large amounts of data, but, using techniques such as multimodal learning, they can also handle different types of data effectively (Ngiam et al., 2011). Furthermore, they are able to identify non-linear correlations over longer time frames, which other methods could overlook. These properties make deep learning a promising approach for the mid-cap default prediction setting, as they allow us to use different forms of data (such as high-frequency pricing data and low-frequency accounting information) alongside each other and capture how they affect default risk without the need for manual feature creation.

Within the deep learning community, Long Short-Term Memory (LSTM) models, originally developed by Hochreiter and Schmidhuber (1997), have for some time been the common method of choice for time series or sequential data. Therefore, we also include LSTMs in our study. However, transformer-based deep learning models have recently produced state-of-the-art results in tasks involving other sequential data such as text, audio and video data (Vaswani et al., 2017). We expect them to perform similarly well on time-series data (Wu et al., 2020), as they can capture long-range dependencies in the data. Crucially, they do not incorporate the position of a data point in a time series as relevant, which is a different design compared to LSTM-based models which employ recurrence as a key feature, using the present input and selected past information to arrive at a prediction. Instead, transformers use the whole past information along with the present to produce their predictions.

Although deep learning can help increase the accuracy of model predictions, interpreting how these predictions are derived presents an added challenge. We address this issue in two ways. Firstly, we will show how transformer models, although complex, are more transparent than recurrent networks, as they allow us to visually interpret the temporal relationships extracted from the data using attention heat-maps. Secondly, we will apply a Shapley approach (Shapley, 1953) to quantify the relative importance of groups of variables and the temporal importance of the data. This will allow us to get sophisticated insights about the mid-cap risk structure.

Therefore, the three key research questions addressed in the paper are:

1. Can an effective transformer-based model be developed that uses accounting, pricing and market data for mid-cap default prediction?
2. Can this architecture accurately predict a term structure for the probability of default over a short to medium-term horizon (3 months to 3 years)?

3. Which data sources and past time periods contribute most to the default risk estimates?

The remainder of the paper is organised as follows. Section 2.2 presents a literature review on corporate default risk modelling, discussing the popular models, studies on specific mid-cap issues and relevant machine learning research. Section 2.3 describes the data used in the paper. The proposed models and the baseline models against which they are compared are described in Section 2.4. Section 2.5 discusses the experimental design, custom metrics, the Shapley group method and hyper-parameter tuning strategies. Section 2.6 presents the results and highlights some discussion points relevant to mid-cap companies.Finally, Section 2.7 summarises the contributions and suggests future steps.

2.2 Literature Review

Corporate default prediction research has thus far focused on three types of approaches. All of these have also seen commercial implementations,by rating agencies such as Standard & Poor's (S&P), Moody's and Fitch Ratings. First, statistical models for default prediction use accounting information from financial statements and apply econometric techniques. These models initially used univariate analysis (Beaver, 1966), later multivariate analysis (Altman, 1968), and they continue to be developed to the present day (Altman et al., 2020). S&P and Fitch use this approach commercially and augment the models with expert opinions and industry-specific metrics. There are, however, limitations to these models. Accounting information could be restated by management or discretionary changes limit the predictive power of these models when companies are under financial stress (Beaver et al., 2012). The second set of models are structural models, which use a combination of accounting and pricing information, within an option theoretic framework. Merton (1974) developed the first such model using Black-Scholes option theory. Structural models are used in commercial applications such as Moody's KMV model (Crosbie and Bohn, 2003). Despite their ability to use current market price information to predict default, there are some limitations to these models as well. Assumptions on asset volatility need to be made as they are not observable and the firm capital structure needs to be simplified to quantify the value of debt as an option on the firm value. Also, default of the firm is endogenous to the model and occurs when the asset value drops below debt outstanding. The real picture, however, is much more complex for mid-cap companies (Jarrow and Turnbull, 2000). The third type of models are reduced form models. They use mainly market information and especially credit spread information of public companies, applying arbitrage-free valuation techniques. Jarrow and Turnbull (1995) first introduced these models where both the interest rates term structure and credit spread term structure are stochastic, unlike previous structural models which assumed interest rates as fixed. Their main use has been in the pricing of credit derivatives of large firms. However, as they rely on public trading information and bond prices, they cannot be applied to private companies or companies with illiquid trading patterns or non-tradeable debt, which makes them unsuitable for mid-cap companies.

Mid-cap companies present their own specific challenges to these credit risk models. Amato and Remolona (2003) first reported the phenomenon of the credit spread puzzle; i.e., they found that the difference between the model-based credit risk estimates and the empirical risk increases as credit ratings drop below investment grade, which is where most mid-cap companies are rated. De Jong and Driessen (2012) and Lin et al. (2011) have suggested the existence of a liquidity premium as one possible factor impacting the credit risk estimates for these companies. Beckworth et al. (2010) found monetary policy shocks to be another factor determining credit spreads, together with economic conditions. Acharya et al. (2013) further explain the puzzle by adding shocks to economic conditions through liquidity, especially for mid-cap companies with non-investment grade ratings. Later studies by Feldhutter and Schaefer (2018) found the credit spread puzzle to be more pronounced for high yield or mid-cap companies, while large firms were less affected. Du et al. (2019) reduced the difference between model and empirical credit spreads by further improving the structural models, including uncertainty from asset risk. Bai et al. (2020) reject the existence of the credit spread puzzle, but their report uses credit default swap spreads, which is a different market to the bond markets used in previous research. The bond market is more relevant to mid-cap firms as they need to raise debt in bond or loan markets.

The third set of challenges that complicate mid-cap credit risk modelling arise from market risk factors. For any firms whose debt is traded, credit risk is not easily separable from market risk. This holds even more for mid-cap companies, whose debt is more correlated with equity indices than with treasury rates, which are representative for debt markets (Jarrow and Turnbull, 2000).

As the aforementioned studies show, modelling mid-cap credit risk is complex and different approaches consider a variety of factors. In this paper, we aim to bring together some of these strands by looking at accounting factors, general market factors and firm equity performance to estimate the probability of default or credit risk. What's more, we propose to tackle this problem with deep learning models and make a case for why they are more suitable.

In default or bankruptcy prediction, Tam and Kiang (1992) was one of the first to use neural networks(shallow) which had better performance against linear models like logistic regression models and Zhang et al. (1999) also demonstrate robustness to unseen data using neural networks.Kim and Sohn (2010) applied Support Vector machines(SVMs) to small and medium scale enterprises default prediction and reported greater accuracy. Later research continued with ensemble of model predictions. Alaka et al. (2018) reviewed different predictive models such as multi-layer neural networks, support vector machines, rough sets, case-based reasoning, decision trees, genetic algorithms, logistic regression and discriminant analysis models in the domain of bankruptcy or default prediction. They found that ensemble models performed better but integrating them is a challenge. Dastile et al. (2020) performed a meta-analysis of the literature and found that in general ensemble of classifiers performed better.They also found deep learning models to show promising results.

Compared to the former machine learning techniques, there is a much smaller but growing number of papers in the area of credit risk modelling that have applied deep learning models, such as LSTMs, convolutional neural networks, and, most recently, transformers. Kim et al. (2021) applied LSTM models to bankruptcy prediction for all US firms between 2007-2019 and found LSTM and ensemble models to perform best in identifying bankruptcies accurately. Given LSTMs common use in other domains as well, we have included them as one of the baseline models in our work. Mai et al. (2019) applied convolutional neural networks to a large dataset containing textual data (from the 10-K reports on financial performance and risks submitted by company management) along with other accounting data and found deep learning models to perform better. Stevenson et al. (2021) applied BERT (Bidirectional Encoder Representations from Transformers) to predict default in micro, small and medium-sized enterprises. They found textual data provided by a loan expert to be predictive of default.

Our approach differs from the above work by considering time-series panel data and modifying transformer models to analyse such data (as opposed to the textual data to which they are most often applied). Vaswani et al. (2017) first developed the transformer model, which introduces a multi-headed self-attention mechanism. This mechanism removes recurrence so that the whole data input can be used. Also, it allows interactions between inputs when extracting relationships. Multiple heads also allow different relationships to be learned. Transformer-based models have since then significantly outperformed LSTM-based models in natural language tasks (Lakew et al., 2018) and speech-related problems (Karita et al., 2019). Moreover, this performance improvement should be extendable to tasks that require taking advantage of complex non-linear relationships that vary temporally (such as the evolution of markets, prices and fundamentals that we study in this work).

Hence, the first contribution of our study is that we are the first to propose a transformer-encoder model for corporate default risk modelling. To adapt this model to our problem, we propose a custom loss function and a performance metric specific to the term structure problem. Second, we develop a framework for multi-modal learning that can combine the different data sources and allows for a differential training approach, where we can train each model separately.

Machine learning models improve predictions but come at the cost of reduced interpretability, which hinders their application in highly regulated areas such as credit risk (Alaka et al., 2018). Transformer models, even though they are complex, are arguably more interpretable compared to other deep learning methodologies. For example, Wiegreffe and Pinter (2019) studied the attention weights after training the model and found them useful for explaining the model's predictions. Although these weights are useful to understand the impact of individual variables, we are also interested in understanding the relative importance of each of the three data channels. For that purpose, we adopt an additional methodology based on Shapley values. Several methods based on Shapley values have been proposed to interpret a model (Lundberg and Lee, 2017), but as we aim to quantify the importance of a group of variables, we follow the approach by Nandlall and Millard (2019). In so doing, we are able to make a third contribution, which is to answer questions about the relative importance of different data sources and study how these relationships vary over time.

Our fourth and final contribution is to the credit risk literature, as we show how multi-horizon probability of default estimates can be produced using a single deep learning model, and how this model produces good results not just in the short term but over a medium term of up to three years.

To benchmark the predictive performance of our proposed transformer model against other methods, we consider a series of methods including logistic regression, shallow neural networks, machine learning classifiers such as XGBoost, and other deep learning alternatives such as LSTMs and Temporal Convolutional Networks (TCN). XGBoost, a scalable decision tree-based ensemble learning algorithm developed by Chen and Guestrin (2016) has achieved state-of-the-art results in many machine learning competitions, especially in classification tasks using structured data (Nielsen, 2016). The same technique applied to bankruptcy prediction also produced good results (Zięba et al., 2016). Second, Temporal Convolutional Networks (TCN) are another deep learning model which combines a series of techniques used in both sequence and image processing models. TCNs have been successfully used to classify time series data in health (Sun et al., 2015; Lea et al., 2017) and other domains (Pelletier et al., 2019). We use the version of TCN developed by Bai et al. (2018) — a generic architecture that can be adapted to our task. Similarly to transformer models, TCNs have not yet been applied to default prediction in consumer or corporate credit risk either, as far as we are aware. Hence, by comparing our proposed transformer model to several powerful benchmark models, we add the necessary robustness to the findings of our study.

2.3 Data

We collected 30 years of data related to mid-cap companies listed in the US from 1990 to 2020, from the following sources: CRSP/Compustat for accounting data and pricing data, Bloomberg and CRSP for default information, and Datastream for market performance data. We exclude financial firms as their leverage and accounting measures are different from non-financial firms, following the standard practice in the corporate default prediction literature (Shumway, 2001).

2.3.1 Data channels

We distinguish between three different data sources (channels):

(i) Fundamental channel: This provides quarterly accounting data expressed as ratios observed at different time points. Sampling is done quarterly instead of over yearly intervals, as the latter would miss the accounting periods' seasonal volatility. The quarterly data is annualised using the last twelve months' metric such that all data is comparable. This data source is useful in capturing the firm's state at a specific time period or understanding how changes in those ratios may affect default risk. We refer to Appendix A.1 for more details about the ratios included and Figure 2.1 shows how data is processed.

(ii) Market channel: Quarterly market performance is collected over the same time period as the fundamental channel data. This data captures general market conditions and includes any financial ratios derived by combining accounting and market data. We refer to Appendix A.2 for the complete list of market indices used.

(iii) Pricing channel: Daily high, low and close history of each firm's equity prices. It consists of very few features, but they are collected at a much higher frequency than the other two channels, providing a detailed record of each firm's recent market valuation history.

2.3.2 Default definition and date selection

We define that a firm defaults if any one of the following criteria is satisfied: the firm filed for bankruptcy; the company is under liquidation; a credit event has been declared as defined by the International Swaps and Derivatives Association (ISDA) which led to the triggering of Credit Default Swaps (CDS); or the firm has failed to pay interest or principal on any of its debt instruments.

This is a broader default definition than simply identifying default on the basis of a bankruptcy filing. It is intended to capture most default scenarios at the earliest opportunity. For example, failure to pay interest or principal is an early indicator of default, which predates a subsequent bankruptcy filing (if any). CDS events also sometimes capture defaults earlier, as the market participants independently determine them. A CDS trigger might not push a company towards bankruptcy, but it could mean losses to its debt holders. This definition makes the predictive modelling more challenging as the firm's financial data might not yet have deteriorated to the same extent as with the traditional bankruptcy or liquidation filing approach. However, it is a more useful approach as this replicates the real-world scenario. For bankruptcy and liquidation data, we used a combination of Bloomberg and Compustat data. For the rest of the data, Bloomberg, CRSP and Datastream were used.

The timestamp that we record for each reporting event is also important to note. Here we take a different approach to the literature, by using the actual reporting date on which the financial results are published, which may differ from company to company. This approach avoids having to add an extra lag to the financial information as is typically done.

2.3.3 Target vector and data structure

To be able to predict default over a short to medium-term horizon, we create a multi-label target vector consisting of binary variables, Y_t, of the form

$$Y_t = [default_{3m}, default_{6m}, default_{9m}, default_{1y}, default_{2y}, default_{3y}],$$

with 1 denoting that a default event occurred over the corresponding time period, or 0 otherwise. For example, if default occurred 10 months after the timestamp, the vector would hold the values $[0, 0, 0, 1, 1, 1]$. This creates an incremental multi-label classification problem, where, as the time horizon increases, the class imbalance decreases. However, a longer time horizon makes the event harder to predict.

The observed inputs, X, for each firm are a matrix of dimensions $w \times f$, where w is the maximum number of historical time periods and f denotes the number of features (input variables) in the data. The input variables collected from the three channels are further preprocessed using standardisation and by treating outliers and missing data. We normalise the data using median and interquartile values and winsorise the data for values beyond 6 times the interquartile ranges. This limits the impact of severe outliers on the model parameters. We replace missing values with the median and add dummies to mark those replacements, since reporting gaps more frequently occur when firms are under financial stress and, thus, these data might not be missing at random. The process chart in Figure 2.1 shows the raw data conversion from various data sources and the preprocessing steps taken to make them suited for the models we apply.

2.4 Models

In this section, we describe our novel Custom Transformer Encoder model (CTE), as well as another recent deep learning model against which it will be benchmarked, i.e. Temporal Convolutional Networks (TCN). For brevity, we omit describing our other baseline models (i.e. shallow neural networks, LSTM, XGBoost and logistic regression) as those are more widely known. The inputs to both the models is a matrix of type NxT where N is the features of the company over T time periods and the output is a vector of size six representing probability of default over 3 months,6 months, 9 months,1 year, 2 year and 3 years.

2.4.1 Custom Transformer Encoder (CTE)

Transformers have thus far been used mainly in the field of Natural Language Processing (NLP). These models incorporate a *self-attention mechanism* store learned patterns. When looking at sequential data, this mechanism ensures that each data point is related to every other data point in the sequence. The architecture further allows for multiple attention heads, each of which can focus on a different aspect of the input, thereby extracting complex non-linear relationships. This ability makes Transformers different in how they handle sequence data. Unlike earlier sequence models based on Recurrent Neural Networks (RNNs), such as LSTMs, transformers take the whole sequence as an input and focus on multiple disjoint sequences to generate patterns. The standard model consists of an encoder and a decoder, as is typical in sequence-to-sequence models. During training, the encoder takes the numerical input and each of its head learn different input aspects, thus creating a higher-order representation. The encoder output is transferred to the decoder. The decoder applies a similar attention mechanism to the output sequence and further applies one more attention combining the encoder representation and earlier output sequence representation. This is passed through a dense feed-forward network to produce the final target vector. This architecture has produced state-of-the-art results in language translation tasks as cited earlier.

The type of data that transformers are designed to handle, i.e. sequence data, makes them suitable not just for natural language problems but also for time-series data. In the NLP setting, the output could be a translated text in a different language (multi-output), sentiment analysis (single output), or other output. The language input has a sequence-like structure due to the grammar and context of the sentence.

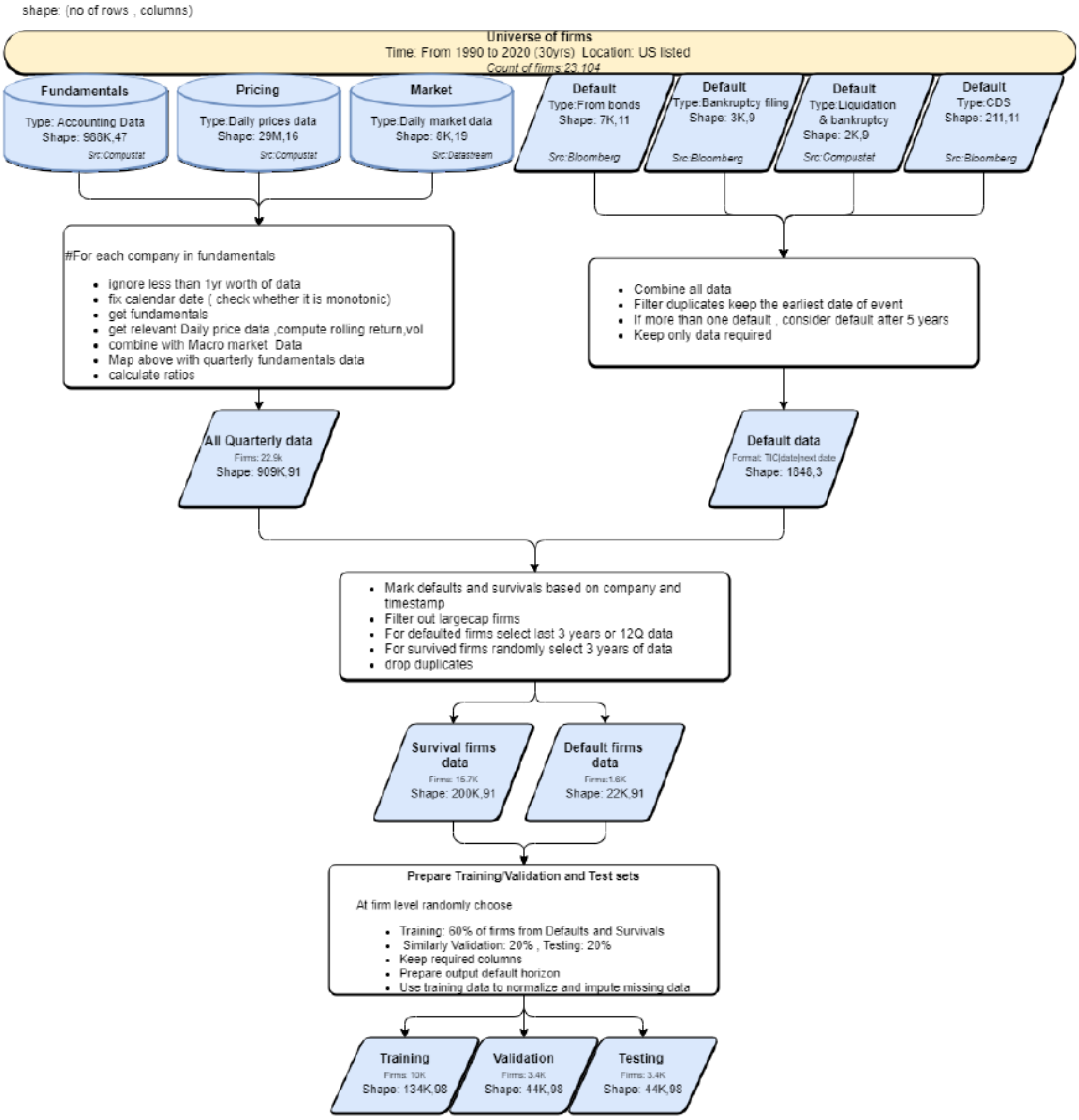

shape: (no of rows , columns)
Universe of firms
Time: From 1990 to 2020 (30yrs) Location: US listed
Count of firms:23,104
Fundamentals
Type: Accounting Data
Shape: 968K,47
Src:Compustat
Pricing
Type:Daily prices data
Shape: 29M,16
Src:Compustat
Market
Type:Daily market data
Shape: 8K,19
Src:Datastream
Default
Type:From bonds
Shape: 7K,11
Src:Bloomberg
Default
Type:Bankruptcy filing
Shape: 3K,9
Src:Bloomberg
Default
Type:Liquidation
& bankruptcy
Shape: 2K,9
Src:Compustat
Default
Type:CDS
Shape: 211,11
Src:Bloomberg
#For each company in fundamentals
ignore less than 1yr worth of data
fix calendar date (check whether it is monotonic)
get fundamentals
get relevant Daily price data ,compute rolling return,vol
combine with Macro market Data
Map above with quarterly fundamentals data
calculate ratios
Combine all data
Filter duplicates keep the earliest date of event
If more than one default , consider default after 5 years
Keep only data required
All Quarterly data
Firms: 22.9k
Shape: 909K,91
Default data
Format: TIC|date|next date
Shape: 1848,3
Mark defaults and survivals based on company and timestamp
Filter out largecap firms
For defaulted firms select last 3 years or 12Q data
For survived firms randomly select 3 years of data
drop duplicates
Survival firms data
Firms: 15.7K
Shape: 200K,91
Default firms data
Firms:1.6K
Shape: 22K,91
Prepare Training/Validation and Test sets
At firm level randomly choose
Training: 60% of firms from Defaults and Survivals
Similarly Validation: 20% , Testing: 20%
Keep required columns
Prepare output default horizon
Use training data to normalize and impute missing data
Training
Firms: 10K
Shape: 134K,98
Validation
Firms: 3.4K
Shape: 44K,98
Testing
Firms: 3.4K
Shape: 44K,98

FIGURE 2.1: Data processing

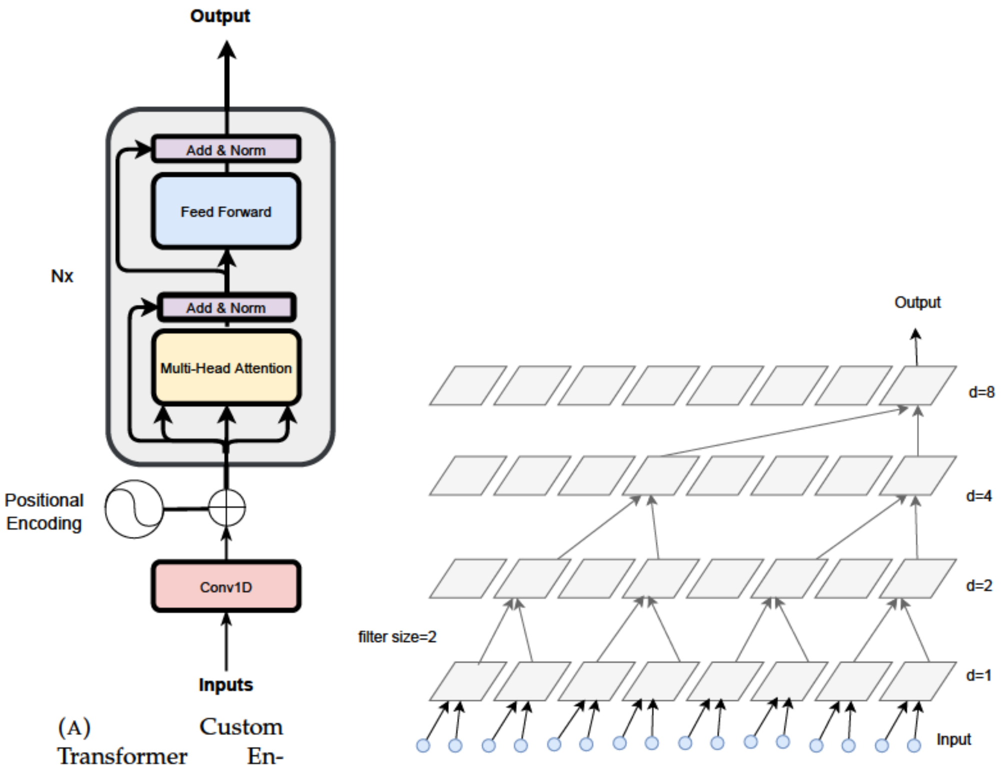
Each word in a sentence can be seen as analogous to a time period in our data. In natural language applications, each word is converted to a vector of integers based on spelling, meaning, and other language attributes. Similarly, for each time period, we have many features that represent the financial state of the firm. When applied to language tasks, transformers apply multi-headed attention to each sentence and learn the sentence's relationship to the output. Here we apply a similar process over time series (sequence) data to learn to predict default probability.

Further advances are being made regarding the application of transformers to time series forecasting (Li et al., 2019a; Wu et al., 2020). In this paper, we modify the original Transformer, by using only the encoder part to form a representation of the input data. As our problem is a multi-label classification task, instead of using the decoder, we then use the encoder output combined with max pooling layer which picks the maximum weighted representation and traditional dense layer to better suit our prediction target. This way, our transformer model encodes our set of time series into several feature vectors, which provide a detailed description of the company and its market context. From the original transformer, we also modify the initial layer by replacing the embedding layer with a 1D convolutional layer as shown in Figure 2.2a. This helps us in two ways. Firstly, unlike textual data that needs to be converted to numerical data accessible to the model, the time series data is already available in a numerical format. Secondly, transformer models have a fixed model size, which ensures a constant size flow of the input representation through each layer of the model. The initial convolutional layer modifies the time-series input to match the model size of the transformer model. This makes it possible to combine different data sources and model outputs, as we will show later. As the performance of the transformer proved sensitive only to the model size and number of layers other aspects of the encoder are left unchanged.

2.4.2 Temporal Convolutional Network (TCN)

A temporal convolutional network is a generic architecture for sequence data (Bai et al., 2018) which was found to give better results over benchmark models such as LSTMs and provides a good trade-off between model complexity and performance. TCNs can store a longer memory than traditional LSTMs and hence perform much better when there are long term persistencies in the data like in financial performance of a company where losses or weak performance could persist over time.

TCNs build up a hierarchical memory over a sequence of data. Initially, they look at nearby relationships for data points and build up a shorter representation of the data as shown in Figure2.2b, near the inputs. Each block consists of dilated convolutional layers with weight normalisation and dropout. This representation is again passed through another block to build a higher-level representation which is the row of blocks above Input in Figure 2.2b. Unlike transformers that focus on all data simultaneously, TCNs build representation in a traditional sequential manner. They achieve this through convolutions. This makes them closer to image recognition models such as Convolutional Neural Networks (CNNs) but applicable to sequential data, including time-series data.



(A) Custom Transformer Encoder(CTE)

(B) Temporal Convuolutional Network(TCN)

FIGURE 2.2: Deep learning models

2.4.3 General architecture

One of our paper's contributions is that it develops a framework to add multiple data sources and combine them. To enable this, a multimodal approach is proposed the architecture of which is given in Figure 2.3.

We could train this multimodel in three different ways: train one data channel at a time; or train each model to its input but simultaneously; and finally, the differential training approach, which utilises the flexibility of our architecture by training each model and data channel separately and still incorporating all the data. We study all three approaches. When a model is trained separately, the models' weights or parameters are reused in the multimodal model. As the parameters are tuned for a specific model, we could either freeze the parameters or use them as initial weights for training in the multimodal setup. Freezing the parameters does not update them

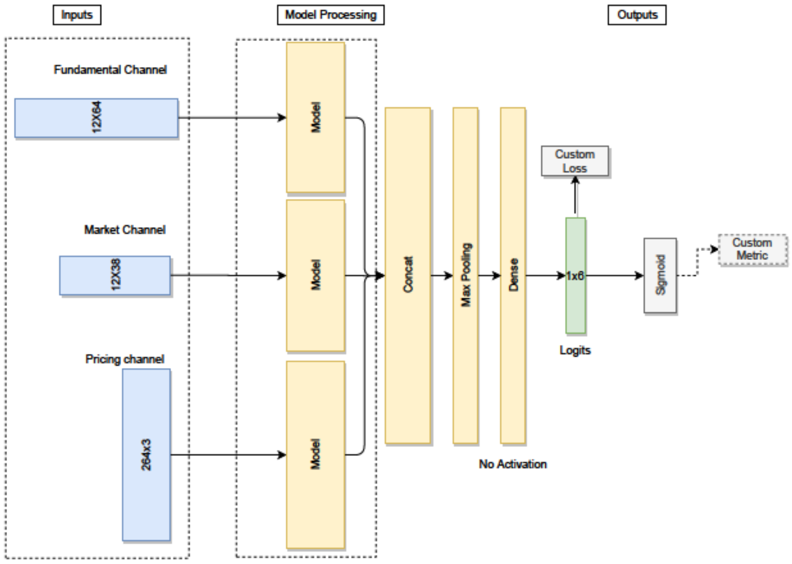


FIGURE 2.3: General deep learning model architecture

while training. This also improves the training time of the multimodal models (Glorot et al., 2011; Lu et al., 2015).

The arrows in Figure 2.3 highlight the general data flow structure from inputs to outputs in Figure 2.3. The dotted lines around inputs or models mean they could be combined or run individually based on the analysis that we are looking to run. For example, if we are looking to use the fundamental channel data only, the other inputs will be disabled, and only one model will be used. The specific model that will be used for this data could be either CTE, TCN or LSTM, but the setup is easy to extend to other forms of data and models as well.

2.5 Model training and experiments

This section describes the loss function, the custom Shapley method we developed for interpretation of the models, the hyperparameter tuning strategy, the optimisation measures used during training, and the two testing strategies used.

2.5.1 Loss function

Here we define the loss function used during training for our models to find the weights or parameters. As we are dealing with an incremental multi-label classification problem, we define a custom loss metric. With the last layer of the network outputting the logits ($\hat{y}_t$) for our respective time horizons (i.e. 3 months to 3 years), we enter each of those outputs into a sigmoid cross-entropy with logits function, defined as follows:

$$loss = -(y_t * log(sigmoid(\hat{y}_t)) + (1 - y_t) * log(1 - sigmoid(\hat{y}_t)) \tag{2.1}$$

where y_t denotes the true default outcome (0 or 1) for that outcome period. To obtain a loss value for the entire observation, we sum the loss values over all those time horizons. This loss function is different from the typical cross-entropy loss function for multi-class classification, as, instead of only one class having a positive outcome, we often observe multiple such outcomes depending on when the default occurred. Note that we do not have strict independence among the binary target vector variables as some combinations are not possible by definition. While we have not enforced that limitation in our current models (which could be done, e.g., by penalising the weights), it did not lead to incorrectly specified probabilities in our results.

2.5.2 Shapley variable group importance

We use Shapley values, a solution concept from game theory, to explain the relative importance of channels and the models' temporal dependence (Nandlall and Millard, 2019). Shapley values are calculated for the multimodel by framing the problem in the form of a cooperative game. Playing a game is analogous to using the model to predict. Maximising the prediction metric is the objective, called the score function.

The players in the game are the data channels defined earlier. If a channel is selected, it is denoted by 1, and 0 otherwise. For G channels, the universe of possible combinations is denoted by T where $|T| = 2^G$, and each combination is a profile p_i where $i = 1, 2, .., 2^G$. $|p_i|$ is the number of channels selected. When $|p_i|$ is 1, the profile is denoted by e_i, implying only one channel among the G channels is selected.

The Shapley set (Q_g) of a channel g is all the sets in T in which channel g is not selected (g^{th} element is 0).

The score function is a characteristic function taking only values between 0 and 1, a higher score indicating a more favourable outcome. While accuracy is often used as a score, in our setting, model performance is more often measured using the Area Under the ROC Curve (AUC). A higher AUC value suggests better ability to discriminate between defaults and non-default, but unlike accuracy, AUC does not take a zero value (but rather a value between 0.5 and 1), so to turn it into a valid score function, we need to rescale it into the so-called Gini coefficient (equal to 2*AUC-1) and use this as our score function.

The marginal contribution of the i^{th} channel is dependent on the profile. For a profile p_n where the i^{th} channel is not included, the marginal contribution is the difference in score when the channel is added:

$$m(p_n, e_i) = s(p_n + e_i) - s(p_n) \tag{2.2}$$

The Shapley value for channel i, $S(i)$,is now defined as

$$S(i) = \sum_{p_n \in Q_i} m(p_n, e_i) * (|p_n|)!(|G| - |p_n| - 1)!/(|G|)! \tag{2.3}$$

In other words, $S(i)$ is the (weighted) average contribution of the i^{th} channel to the game, weighing all possible combinations to which the channel can be added appropriately. A higher score implies a higher contribution of the channel's data towards the predictive power of the model.

2.5.3 Hyperparameter tuning

We used a grid search to tune the hyperparameters for each model, using a validation data set covering 20% of the total data. To speed up the search, we used parallel processing techniques.

For logistic regression, we used the saga solver with L2 penalty, as it is easier to optimise than the L1 penalty but performed similarly in our experiments in terms of predictive performance.

The XGBoost model hyperparameters were tuned with a grid search for the learning rate {0.001, 0.01, 0.1}, maximum depth {2, 3, 4}, number of estimators {50, 100, 250, 500} and alpha {0.1,. . . ,0.9}. These numbers were chosen so...

For the deep learning models, we found the batch size and number of epochs to be less important as we trained the models with early stopping, as explained later in section 2.5.4. The shallow neural network consisted of two hidden layers and one output layer. The first two layers were tuned over different number of units in the range of {50,100,150,200} and {10,20,30,40,50}, respectively. In the LSTMs, we tuned the number of units, over the range {16,32,64,96,128,150}, the dropout rate {0.1,0.2,0.3} and the optimiser {'adam','sgd'}. The TCN's hyperparameters are different as it is a convolutional network-based model. There, we conducted a grid search on the number of filters {16,32,64,128}, kernel size {1,3,6}, the activation function {'tanh','relu'} and dropout {0.1,0.2,0.3}. Finally, in the proposed CTE, the model size and number of layers are the key parameters that need to be determined. We tuned the model size (M) over {6,12,18,24,36,48,54,72,84,96,102} and based on validation data performance set it to 72. The number of layers (l) was tuned over {1,2,3,4,6,12}. Once the layers and model size are fixed, h or the number of heads is defined as M/l. All the other hyperparameters in the model were unchanged from their defaults as the impact of further tuning them proved marginal.

To select the window size for the accounting input data, we experimented by training LSTM and TCN models with different window sizes of 4, 8 and 12. This represents lookback periods of 1, 2 and 3 years, respectively, as each year has four quarters of accounting data. Both models performed better with larger window sizes, implying that using a longer time span of financial data benefits deep learning, and that these methods have the capacity to process it.The same window size was applied across all models and combinations later on.

As for the pricing channel, this has daily prices covering the previous two years, making the potential lookback period quite deep. We used a grid search for the appropriate window size for each model, trying window sizes of 3, 6, 9, 12 and 24 months. In the results section, we will report how the performance of each model changes with the choice of window size.

2.5.4 Training settings

To prevent overfitting the data, we trained the models with early stopping, whereby training is stopped when the validation set loss metric no longer decreases. To avoid local minima, a patience setting of five (eight) was selected for the multimodal (single-channel) model setup, respectively. We apply more patience to single channel training as it is expected to take a larger number of epochs compared to the multimodal model whose parameters have already been tuned. This is especially true for the pricing channel where single-channel training ran for 30-40 epochs in our analyses, while the multimodal training only required 3 to 5 epochs.

All models were first assessed on an independent test set (20% of the data), using AUC as the performance criterion. Furthermore, to assess the robustness of the model performance estimates, we also carried out a stratified 10-fold cross-validation procedure. This ensures the model is tested to various changes in variable distributions and relationship or concept drift over time. Instead of the traditional procedure which would simply divide the training observations into 10 folds, we define the folds by assigning different companies to different folds; this ensures that observations linked to the same company appear in the same fold. We will report the average performance and variance across all folds.

2.6 Results and discussion

In this section, we present three sets of results. The first subsection summarises and compares the performance of all models, on the test set. The next subsection shows the results of our robustness checks. Thirdly, we show how the transformer model's multi-head attention weights can provide a partial model explanation and we compare the importance of the three channels using the Shapley approach.

2.6.1 Model performance results

2.6.1.1 Single channel, quarterly fundamental data

First, we consider all models built using only the fundamental channel data as input. Table 2.1 shows the AUC for each resulting model and each time horizon (e.g., d_{1y} is the AUC score for the estimated probability of default in one year). The first column takes the average AUC over all time horizons.

TABLE 2.1: Model performance: single channel, fundamental data(best result in bold)

Input: Quarterly fundamentals only	**AUC**						
Model	Average	d_3m	d_6m	d_9m	d_1y	d_2y	d_3y
CTE	**0.785**	**0.824**	**0.811**	**0.797**	**0.775**	**0.756**	**0.747**
TCN	0.780	0.814	0.804	0.793	0.775	0.750	0.743
LSTM	0.777	0.808	0.799	0.792	0.772	0.753	0.742
NN	0.756	0.768	0.774	0.769	0.762	0.747	0.738
XGB	0.715	0.749	0.739	0.731	0.713	0.684	0.676
Logistic	0.702	0.643	0.681	0.702	0.719	0.732	0.733

With an average AUC of 0.785, the transformer (CTE) model shows the best performance, but it is closely followed by the sequential deep learning models TCN and LSTM. All of these models outperform a shallow neural network model (NN). A potential explanation could lie in emerging complex structures which deeper models are better able to capture. XGB did not give competitive performance, which suggests it probably requires a larger number of observations to extract predictive patterns. As expected, logistic regression, being a relatively simple linear classifier, has the weakest performance.

2.6.1.2 Single channel, quarterly market data

The market data channel contains general market prices of several indices as well as some company-specific data, which differentiates the data observed for different firms in the same time period. Table 2.2 summarises how each of the models performs on this second data source.

TABLE 2.2: Model performance: single channel, market data(best results in bold)

Input: Quarterly market data		**AUC**					
Model	Average	d_3m	d_6m	d_9m	d_1y	d_2y	d_3y
CTE	0.767	0.786	0.790	0.777	0.759	0.742	0.748
TCN	0.767	0.779	0.782	0.776	0.761	0.748	0.754
LSTM	0.770	0.786	0.784	0.775	0.762	**0.753**	**0.762**
NN	**0.772**	**0.787**	**0.790**	**0.782**	**0.765**	0.752	0.754
XGB	0.752	0.760	0.763	0.756	0.745	0.743	0.749
Logistic	0.741	0.771	0.766	0.751	0.728	0.713	0.715

As shown, the deep and shallow neural network models all perform similarly. They are, however, better than the XGB or the logistic model. This could be linked to the high level of repetition in data that is linked to the same time period, which the former models are better designed to handle. With the AUCs being somewhat lower than in Table 2.1, there is clear value in the market data but less than in the accounting data.

2.6.1.3 Single channel, daily pricing data

The pricing channel contains just three features but has more frequent data than the fundamental or market channels. The first question is which look-back period or window size of past data to select.

TABLE 2.3: Pricing model performance for different lookback window sizes (best results in bold)

Test AUC		**Window size**			
Model	3m	6m	9m	1y	2y
CTE	0.698	0.710	0.711	**0.716**	**0.736**
TCN	0.702	**0.715**	**0.726**	0.701	0.731
LSTM	0.588	0.654	0.626	0.570	0.657
NN	**0.702**	0.703	0.702	0.705	0.708
XGB	0.681	0.693	0.701	0.707	0.715

Table 2.3 shows that, as the pricing data's window size increases, the transformer model's AUC consistently improves, from 0.698 to 0.736. TCN, LSTM and, though to a lesser extent, NN tend to improve with larger window sizes, too. Note that we dropped the logistic regression model from the analysis as its performance on the pricing data was close to random.

Based on these results, a two-year window is selected. Next, employing this two-year window, Table 2.4 shows how well each model can predict default.

TABLE 2.4: Model performance: single channel with pricing data(best results in bold)

Input: Pricing data	AUC						
Model	Average	d_3m	d_6m	d_9m	d_1y	d_2y	d_3y
CTE	**0.736**	0.756	0.747	**0.759**	**0.735**	**0.716**	**0.700**
TCN	0.731	**0.761**	**0.750**	0.747	0.729	0.706	0.694
LSTM	0.657	0.692	0.681	0.671	0.643	0.625	0.632
NN	0.708	0.733	0.733	0.726	0.708	0.681	0.669
XGB	0.715	0.749	0.739	0.731	0.713	0.684	0.676

The CTE and TCN are clearly superior to the other methods in this setting. In other words, they can extract more meaningful information from daily mid-cap equity prices. However, the overall performance remains lower compared to the other channels, indicating that there is less predictive value in this type of data. The models also take longer to train: while the models for the other two data channels converged in 8 to 10 epochs, the pricing data took 30-40 epochs. As there is much more noise in daily pricing, it takes longer to derive a useful signal for default prediction.

Combining high-frequency pricing data with low-frequency accounting data is not straightforward. Directly combining such data would require resizing the input matrices (e.g. by turning quarterly data into daily values). Instead, deep learning provides several alternatives for building multi-channel models, as discussed next for the best performing model type identified thus far, i.e. the CTE model.

2.6.1.4 Multi-channel, all data

The multi-channel model is designed to use data from all three channels, using the architecture proposed in Figure 2.3. As this allows that the three sets of inputs are fed to the network separately, they can have different dimensions. We consider the different approaches to training described in Section 2.4.3. That is, either we can jointly train the full model, or train separate model components and then freeze the weights for those channels. The results for the corresponding training options are presented in Table 2.5.

TABLE 2.5: Multi-channel CTE model performance, for different training methods.

Input: Quarterly and daily data channels	AUC						
Method	Average	d_3m	d_6m	d_9m	d_1y	d_2y	d_3y
Training together	0.811	0.827	0.823	0.818	0.804	0.795	0.800
Pricing channel freeze	0.816	0.838	0.828	0.820	0.810	**0.802**	0.800
Market and Pricing channel Freeze	**0.821**	**0.844**	**0.839**	**0.826**	**0.811**	0.800	**0.803**

The prediction model clearly benefits from including all three channels, as the AUC is larger than for all previously trained models. showing how long time frames, varied data, and a complex data flow can lead to better results. A multi-channel model with a differential training approach yields the best AUC (average of 0.821), outperforming a simultaneous training strategy (0.811). This suggests the former approach is better at handling the structural differences between the three input sets.

2.6.2 Robustness check: 10-fold cross validation

To test the robustness of our findings, we performed a 10-fold stratified cross validation check for the multi-channel model, with the strategy that produced best result, assigning firms to different folds as described in Section 2.5.4.

TABLE 2.6: Stratified k-fold cross validation: mean AUC (standard deviation)

Stratified 10-fold cross validation			**AUC**			
Average	d_3m	d_6m	d_9m	d_1y	d_2y	d_3y
0.869 (0.011)	0.881 (0.025)	0.884 (0.016)	0.880 (0.013)	0.871 (0.010)	0.854 (0.008)	0.846 (0.010)

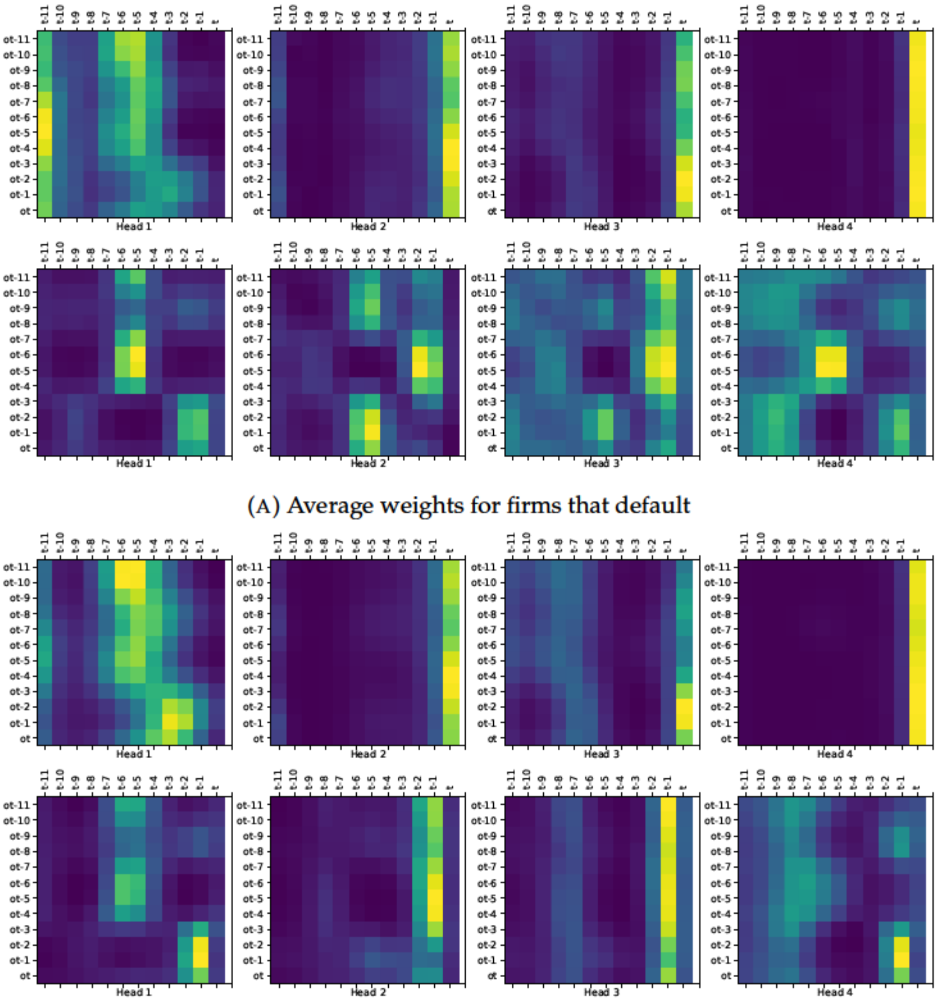
Table 2.6 confirms that the proposed CTE architecture produces excellent default predictions, regardless of the time horizon. These results support the idea that the learning is able to detect true patterns as opposed to noise, which successfully generalise to previously unobserved companies. Furthermore, the deep learning model can efficiently combine multiple information channels with limited preprocessing.

2.6.3 Interpretability of the architecture

Although the CTE was shown to produce highly accurate predictions, one challenge lies in providing a suitable interpretation of what factors led to those predictions. Hence, to better understand the model, we will first demonstrate how to interpret the transformer model's multi-head attention weights; the second subsection will then discuss the insights gained from the Shapley approach outlined earlier.

2.6.3.1 Multi-headed attention weights

Transformers models, for all their complexity in design, do provide an interesting layer of interpretability. Each head in each layer of the transformer encoder is expected to learn a different input data aspect. This kind of interpretation has been previously used in the NLP domain for translation tasks. Here we adapt the idea to



(A) Average weights for firms that default

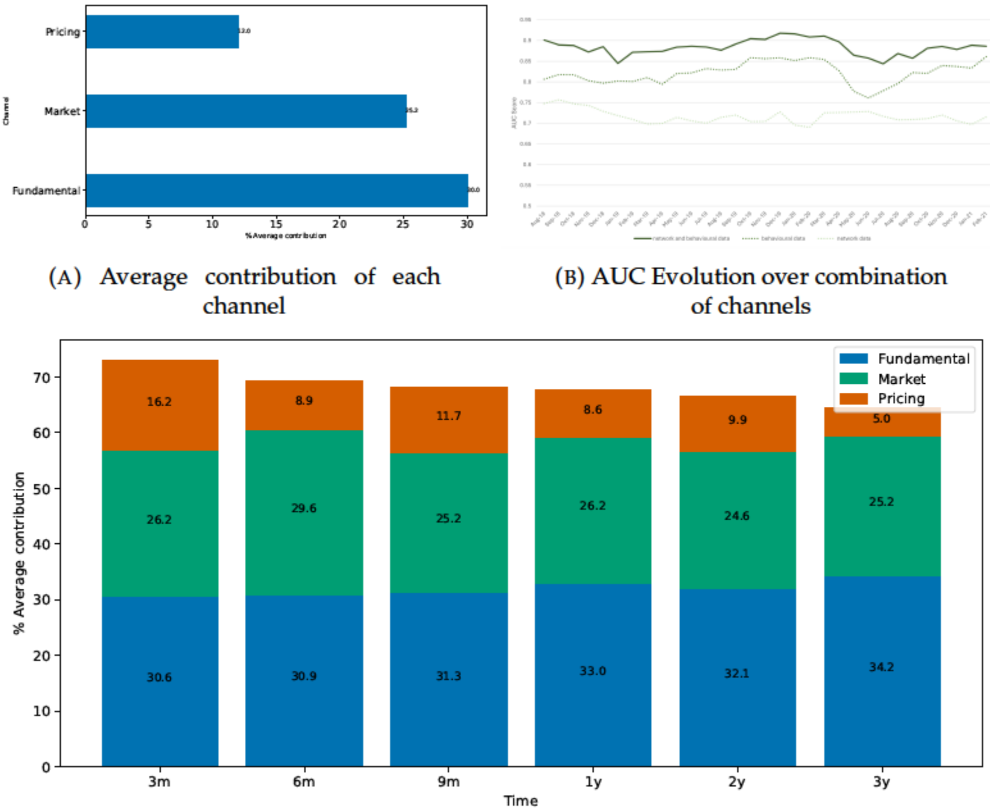
(B) Average weights for firms with no observed default

FIGURE 2.4: Attention weights mapped to time periods, over defaults and survivals

time series data. To illustrate this, we select the fundamental channel data only. This gives a direct interpretation between the CTE output and input.

Each plot in Figure 2.4 visualises the attention weights for one of the four heads (see the figure columns) in one of the two layers (rows) of the transformer model trained earlier. The horizontal axis in each plot divides the input data according to time quarter; the vertical axis is the output representation. This mapping thus shows which time period is given a higher weight by the head; the highest weights are shown in yellow, the lowest are in deep blue. To understand how the model distinguishes between default and non-default outcomes, we compare the average weights for firms that default (top panel) with those that do not (bottom panel).

The first layer of defaults and survivals exhibits few differences. However, the second layer does show differences: The second head in the second layer for defaulted firms focuses on data from the $t-5$-th period and $t-2$-th period while the same head for survived firms, looks at the $t-1$ period. This can be interpreted as follows: if a firm



(A) Average contribution of each channel

(B) AUC Evolution over combination of channels

(C) Channel importance over time

FIGURE 2.5: Channel Importance Interpretation

has certain financial ratios in the last quarter of accounting data ($t-1$), it will be more likely to be classified as a survival. However, if it does not satisfy this, the model looks at previous financial year data ($t-5$) to check for specific patterns to classify the firm as a default. This shows the model extracting complex temporal relationships. Some heads in both cases mainly use the present time period ($t-4$ to t) to extract relationships. In the next section, we proceed to quantify the importance of present data over past data.

2.6.3.2 Relative importance per channel

Using the Shapley derived method defined earlier, we present the results for each channel's relative importance. The method allows us to see how each combination of the inputs has impacted the AUC score.

In Figure 2.5a, the fundamental channel has the highest relative importance, 30%. This means that, on average, the inclusion of fundamental data into the model improves the model's AUC metric by 30%. Figure 2.5b reports the AUC values for different data combinations. For example, using just the fundamental channel, we achieve an AUC of 0.791. Adding the Pricing channel improves the performance slightly to 0.807, while the Market channel improves the AUC metric by 5.3%, to 0.833. From both of these, it is clear the Market channel adds a statistically significant contribution compared to the pricing channel.

To take a closer look at the impact of the pricing channel, we look at how the relative importance of channels varies over each prediction horizon in Figure 2.5c. In the short term, the pricing channel plays some role with 16.2% importance and decreases to just 5% in the three-year horizon. We could infer the pricing channel provides some signalling in the short term. Still, over the medium term, fundamentals and the general market environment play a larger role in determining the probability of default. This follows intuition and somewhat aligns with the weak market efficiency hypothesis: prices reflect the market's current belief, taking into account short-term fluctuations, but true long-term estimation ignores these blips caused by events that may prove meaningless in hindsight.

The temporal aspect of the results is another important factor that needs to be understood. Which time period contributes most towards the results could be calculated using the same Shapley method. We group the variables into yearly data, starting from the quarterly data. The twelve quarters of data are grouped into two groups. The first group contains one year worth of data, which is the firm's present data, while the rest of the data is grouped as past. Each channel is then evaluated on test data to get their relative importance.

TABLE 2.7: Shapley Contribution of each channel over time (%).

	Shapley values	
Channel	Present 1 year	Past 2 years
Fundamental	52.3	12.4
Market	35.1	20.0
Pricing	38.4	9.3

The results in Table 2.7 show the importance of the latest time period over previous years data across all channels. In the fundamental channel, over 52% of the performance comes from the present time period data. The previous time periods still contribute positively to the model's predictions. However, in the market channel, the temporal aspect is relatively more important as 20% of the contribution comes from past data. The changes in the environment or market cycles make an impact on firm performance. The present state of the market is impacting a firm at an individual level with a lag. This could be expected as it takes some time for uncertainty in the macroeconomic environment to impact firms. In the pricing channel, 2 years of past data is more important compared to previous year data. This implies a long term relationship being extracted. Firms in decline have a general under-performance in their equity, and these could be known over a long term than short term price movements.

2.7 Conclusion

The paper has shown deep learning techniques are powerful in default prediction for mid-cap companies when carefully engineered, predicting complete term structures, going beyond one-year predictions hitherto common in the area. The increasing complexity of the models does increase predictive power. To be able to get this increased performance, however, new strategies were needed. Combined multimodal architectures were designed, as they were much better than a single large model. This architecture also gave the flexibility to treat each data source differently and take advantage of selective learning mechanisms.

Custom learning methods needed to be devised specifically for the problem. We developed a custom loss metric for training purposes that is relevant to the incremental multi-label classification problem. An efficient setup is important as several models with different data combinations and different hyperparameters' choices need to be tuned. Using measuring performance consistently and increased the training speed by applying on the whole training set instead of at the batch level is traditionally done. Such options need to be carefully considered when deploying end-to-end deep learning models.

While the training strategy and the custom loss functions can be applied in any deep learning model, such as TCN or LSTM, we also developed a transformer-based model and showed how to adapt them to handle structured data. The increase in this model's performance over large amounts of data shows promise in handling complex non-linear relationships over long time frames. CTE handled lower-frequency data with many related features and handled high-frequency data when other models had significant drops in performance.

On the contribution towards financial analysis, our results show that deep learning models apply to mid-cap companies, probable more so than traditional approaches applied to large-cap companies. The former are companies where data could be missing or not as extensively followed as the latter companies. Their prices could be more volatile and have a higher default rate compared to large-cap companies as well. We were also able to show that accounting data still has more value in predicting default. However, pricing data can provide valuable signals provided we develop a specific strategy to handle this source of information.

Predicting the probability of default over a multiple time horizon was accomplished within the same model architecture. This information would be useful to understand the term structure of credit spreads. Instead of tuning multiple models to different data sources, we could find a single model to produce multiple outputs, as in ensemble models.

We developed a custom methodology to interpret a deep learning model using Shapley values for groups of variables. We could infer pricing information is of limited, time-decaying, usefulness, while the market context is much more important. Also, we were able to visually understand the differences between defaulted firms and surviving firms from the heatmaps derived from the CTE model. With increased performance and better interpretability, we believe deep learning models could add significant value in the default prediction space for corporates or possibly other credit risk domains.

Another potential avenue for future research, expanding multimodal learning, could be further extended to develop new kinds of scorecard models for credit risk. Adding further channels to predict business sector performance, management, and other factors would be interesting. This paper dealt with only structured data and more towards financial and market data, complementing previous research that used unstructured data such as text. The framework given here could be extended to any other type of data to improve the model further. Adding unstructured data like textual documents to the channels could further improve the model, e.g., incorporating management discussions information to the fundamental channel and news feed with relevant company news to the pricing channel.

Acknowledgements

This work was supported by the Economic and Social Research Council [grant number ES/P000673/1]. The last author acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [Discovery Grant RGPIN-2020-07114]. This research was undertaken, in part, thanks to funding from the Canada Research Chairs program.

The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

Chapter 3

Portfolio optimisation using Graph neural networks

Abstract

Apart from assessing individual asset performance, investors in financial markets also need to consider how a set of firms performs collectively as a portfolio. Whereas traditional Markowitz-based mean-variance portfolios are widespread, network-based optimisation techniques offer a more flexible tool to capture complex interdependencies between asset values. However, most of the existing studies do not contain firms at risk of default and remove any firms that drop off indices over a certain time. This is the first study to also incorporate such firms in portfolio optimisation on a large scale. We propose and empirically test a novel method that leverages Graph Attention networks (GATs), a subclass of Graph Neural Networks (GNNs). GNNs, as deep learning-based models, can exploit network data to uncover nonlinear relationships. Their ability to handle high-dimensional data and accommodate customised layers for specific purposes makes them appealing for large-scale problems such as mid- and small-cap portfolio optimisation. This study utilises 30 years of data on mid-cap firms, creating graphs of firms using distance correlation and the Triangulated Maximally Filtered Graph approach. These graphs are the inputs to a GAT model incorporating weight and allocation constraints and a loss function derived from the Sharpe ratio, thus focusing on maximising portfolio risk-adjusted returns. This new model is benchmarked against a network characteristic-based portfolio, a mean variance-based portfolio, and an equal-weighted portfolio. The results show that the portfolio produced by the GAT-based model outperforms all benchmarks and is consistently superior to other strategies over a long period, while also being informative of market dynamics.

Keywords

Portfolio optimisation, mid-caps, correlation networks, distance correlation, filtered graphs, deep learning, graph attention networks

3.1 Introduction

Portfolio optimisation is crucial in financial risk management, as performance correlations between firms in a portfolio bring unforeseen risks. Given that each investor's risk profile differs, the portfolio construction model must also account for different objectives. One such type of model, based on the classic work by Markowitz (1952, 1959), is mean-variance optimisation, which trades off maximising returns against minimising volatility. In this paper, too, we look to optimise with a mean-variance objective, but we do so over a large set of firms that could also default or go bankrupt. This problem must be solved for an investable universe that is expanding, as financial markets continue to develop, and emerging and private markets are becoming increasingly accessible. All these assets have different risk and liquidity profiles. Against this backdrop, which firms to select and what proportion of capital to allocate to each is an increasingly high-dimensional problem.

Portfolio optimisation often involves estimating the expected returns and covariance matrix and then using a constrained optimisation method to find the asset allocation weights that maximise the portfolio objective. The classical mean-variance measure is not without its problems, though, and portfolios optimised using it have been shown to exhibit poor out-of-sample performance (Siegel and Woodgate, 2007). Assumptions about the normality of returns and absence of transaction costs, as well as the presence of regimes in markets, make the classical model difficult to implement (Guidolin and Ria, 2011). Even if these assumptions are fulfilled, the expected mean of the portfolio returns and the covariance matrix cannot be readily estimated as they are not observed in practice, which means that, instead, the sample mean and covariance matrix are commonly used (Ao et al., 2019). Furthermore, it is challenging for such models to cope with high dimensionality, a common characteristic of modern portfolios (DeMiguel et al., 2009b). To better address these challenges, new methods to solve the portfolio optimisation problem continue to be developed, borrowing from different techniques in other domains, such as fuzzy programming (Arenas Parra et al., 2001), cluster analysis (Puerto et al., 2020), quantum annealing (Venturelli and Kondratyev, 2019) and deep reinforcement learning (Shi et al., 2022a).

Of particular interest to our work are topological or network studies for portfolio optimisation (Pozzi et al., 2013; Li et al., 2019b). Network models exploit graph data structures to identify relationships that may be impossible to detect by Euclidean data-based models. In our case, the network nodes are the firms, and each edge represents a relationship between two firms. More formally, the network at a given time t is represented as an undirected graph $\mathcal{G} = (\mathcal{V}_t, \mathcal{E}_t)$ where $\mathcal{V}_t$ are the nodes or firms and $\mathcal{E}_t$ is the set of edges, often represented by an adjacency matrix A of dimension $|\mathcal{V}_t| \times |\mathcal{V}_t|$.

The aforementioned network studies consistently find that allocating capital to firms in the peripheries of the networks produces higher returns, due to low correlations with other parts of the network. Similarly, to produce a diversified portfolio, mean-variance models also tend to prefer firms in the peripheries of the network (Onnela et al., 2003). However, studies using the former methods have only been deployed to small portfolios or were limited to a specific sector. Here, we look to extend these topological analyses to the whole market of US mid-cap companies, a much more challenging and realistic problem setting.

Mid-cap firms (in short 'mid-caps') are defined as firms with a 1 to 10 billion USD market capitalisation and are likely constituents of the Dow Jones Wilshire Mid-cap or S&P 400 Mid-cap indexes. Modelling the performance of these firms is complicated by the low-volume and, at times, illiquid nature of trading, which makes their return distributions non-normal (Castellano and Cerqueti, 2014). They are also far more numerous compared to large-cap firms, which makes mid-caps less suitable for analysts to cover. However, as they behave as a separate autonomous asset class, they can further improve the diversification aspect of a portfolio if included (Petrella, 2005). Over the long term, mid-caps also provide a premium in return for the same risk, which is desirable for any portfolio seeking financial returns (Ge, 2018). Given the large number of companies in the mid-cap universe, simple index replication strategies, such as those implementing the popular market-weighted methodology of the Russell 2000 Index, can be costly though. Furthermore, investors may have different horizons and risk tolerance, whilst constituent churn can negatively affect performance (Cai and Houge, 2008; Cremers et al., 2020). Hence, a comprehensive approach is needed to generate portfolio weights for mid-cap firms that yield better risk-adjusted returns. Another area of practical interest to which our study may be applicable is the development of new automated ETF strategies for such companies. The latter would require large-scale portfolio optimisation models incorporating the strategy constraints of the particular ETF.

Studying the correlation between firm's returns or volatilities is an integral part of any portfolio optimisation procedure. Pearson correlation approaches, generally used by mean-variance models, can only capture linear dependencies and pairwise correlations. Instead, in this work, we employ the distance correlation measure (Székely et al., 2007). This is able to capture non-linear relationships between pairs of firms. Sun et al. (2019) compared the use of distance correlation with Pearson correlation for portfolio optimisation, and found that the distance correlation strategy indeed performs well. Furthermore, most previous studies did not allow for natural churn in the portfolios, the presence of which we believe makes distance correlation an even more attractive option. Thirdly, as mid-cap companies are more illiquid compared to large-caps, their available trading history may be shorter, less uniform, or missing for some period of time. The distance correlation measure can handle such time series features and still produce a quantitative measure of the relationship between firms. This allows us not to drop any firms, thus avoiding selection bias. Hence, we use the covariance matrix generated by distance correlations to produce a fully connected initial network of firms. Subsequently, we apply the Triangulated Maximally Filtered Graph (TMFG) method introduced by Massara et al. (2017) to filter this dense matrix. This process results in a network with fewer edges, representing the strength of the relationships between firms with minimal loss of information. The added sparsity yields more meaningful initial relationships that subsequent models can build upon.

Once networks are formed and stock price data are included, traditional methods fall short on handling the complex resulting data structure. Deep learning techniques, however, with their ability to create higher-order representations of any available data, commonly excel at this. As they do not impose restrictions on the data distribution and can handle non-standard data types by design, they are applicable in a wide variety of settings. For example, they have produced state-of-the-art results in several domains, such as speech recognition, natural language processing, object detection, drug discovery, and genomics (LeCun et al., 2015). Furthermore, deep learning models tend to scale well to high-dimensional datasets, Avramov et al. (2023) applied deep learning techniques such as feed-forward neural networks and conditional autoencoders to identify mispriced stocks consistent with most anomaly-based trading strategies. A recent study on midcap default prediction has also shown (Korangi et al., 2022), they can be designed to find optimal solutions for various problem types with different objectives or constraints. This suggests that deep learning methods may be well-suited to mid-cap portfolio optimisation.

In this paper, we propose employing a class of deep learning methods, Graph Neural Networks (GNN), which can learn non-linear, complex representations of the firms. More specifically, we use Graph Attention networks (GAT), a variant of GNNs that employs attention mechanisms to weigh the importance of a firm's neighbouring nodes (Veličković et al., 2018). Unlike previous topological studies, which have shown that network structure can play a vital role in portfolio optimisation, GNNs distil information from the relationships to produce the portfolio weights without relying on a few static measures. GATs, in particular, work on specific sub-structures of graph data, while the graph data can be dynamic. This makes them an attractive option for portfolio optimisation, as the relationships between firms can change over time and may behave differently under different macroeconomic environments. The optimisation procedures of deep learning architectures also work well in higher-dimensional space, such as the historical returns of a large number of firms.

Like other deep learning architectures, GATs can also be trained to optimise any chosen objective function and are flexible in terms of the types of output that can be produced.

To measure portfolio performance, we use the widely applied Sharpe ratio, i.e. the ratio of returns over volatility (Sharpe, 1966). We employ a similar criterion for model training, by using a custom loss function derived from the Sharpe ratio. This allows us to target generating optimal portfolio weights, without having to predict individual returns in the same way that, for example, Zhang et al. (2020) did. To cope with the dynamic nature of the problem and allow the set of active mid-cap firms to vary over time, we use a rolling window approach which allows graph inputs and past returns data to vary as we move between forecast periods.

Therefore, the three key research questions addressed in the paper are the following:

1. Can an effective network topology be constructed from sparse historical data on a large collection of firms?
2. Are graph attention networks able to generate higher-order representations of this network that enable constructing an optimal portfolio for mid-caps?
3. How does the model perform under different market conditions, and can we infer useful strategies from the model results?

In so doing, the paper makes three main contributions. First, we develop topological portfolio optimisation models, extending the literature that hitherto focused chiefly on stochastic models, and applying the resulting approach to the challenging set of mid-cap firms. Second, by using GPUs and parallel computing, we are the first to be able to apply the distance correlation measure and subsequent TMFG filtering at this scale. Finally, we use the resulting networks as inputs to graph-based deep learning models and show how these are capable of producing portfolio weights for the large number of firms we are faced with.

The remainder of the paper is organised as follows. Section 3.2 reviews the relevant literature, focusing on large-scale portfolio optimisation, graphs and graph-based deep learning models. Section 3.3 describes the data and the process by which this data is converted into graphs, as well as defining several measures used in our work. The proposed models, and the baseline models against which they are compared, are further described in Section 3.4. Section 3.5 elaborates on how we set up the empirical analysis, summarising the different steps and types of model comparisons made. Section 3.6 then presents and discusses the results. Finally, Section 3.7 summarises the main insights gained from our study and suggests some future research.

3.2 Related literature

In this section, we discuss relevant literature for the study, focusing on large-scale portfolio optimisation studies and GNNs.

3.2.1 Correlation networks and portfolio models

Portfolio optimisation has been the subject of a large body of research, and novel methods, with various constraints and objectives, continue to be developed (DeMiguel et al., 2009a; Branch et al., 2019). Of particular interest to our work are studies that have focused on optimising large-scale portfolios effectively. Perold (1984) was the first to do so, by considering the specific nature of dense covariance matrices, and recommending strategies to make them sparse so that their analysis becomes computationally feasible. More broadly, the first branch of large-scale portfolio optimisation studies followed a similar approach by devising algorithms to reduce computational time and memory space requirements for the classical mean-variance approach.

Later studies on large-scale portfolio optimisation have focused on proposing model extensions and computational methods to solve them, measuring the performance of the resulting allocation weights using performance metrics such as the Sharpe ratio. Extending the mean-variance framework by adding a probabilistic constraint requiring that the expected returns exceed a chosen threshold with a high confidence level, and introducing additional trading constraints, Bonami and Lejeune (2009-05/2009-06) proposed a novel exact solution for their resulting model, which they tested on a portfolio of up to 200 firms. Demonstrating their approach for pools of stocks of up to 100 S&P 500 firms, Ao et al. (2019) proposed an unconstrained regression representation of the mean-variance portfolio problem, which they estimated using sparse regression techniques. Bian et al. (2020) and Dong et al. (2020) used regularisation methods for portfolio optimisation as, without such methods, a large universe of stocks would lead to overly small or unstable allocations (and, hence, high transaction costs). They found that these techniques improved portfolio performance (in terms of the Sharpe ratio) compared to the standard model. Performance-based regularisation, whereby the sample variance of the estimated portfolio risk and return is restricted, also performed better on several Fama-French data sets (Ban et al., 2018). Recently, Bertsimas and Cory-Wright (2022) reviewed the size of portfolios that previous large-scale portfolio optimisation studies were able to handle and proposed a ridge regression-based regularisation algorithm that speeds up the convergence of sparse portfolio selection. They showed that their method can select up to 1000 stocks from the Wilshire 5000 equity index. However, they did so without reporting the Sharpe ratio of this selection.

Despite these computational advances, we argue that, in all of the aforementioned work, either there was further scope for increasing the portfolio size, or the performance analyses to measure the efficacy of the proposed algorithms were restricted to using simulated data. Additionally, in all of these studies, the universe of firms or assets from which to select was always kept constant, and firms that default or those that are acquired or liquidated were thus omitted, eliminating an important driver of idiosyncratic portfolio risk. Instead, we take a dynamic approach and allow both the universe of selectable stocks and the chosen portfolio to change over time. In so doing, we avoid selection bias and ensure that our approach more closely mirrors a real-world scenario wherein investors wish to invest in a certain market sector or asset class. We seek to make an optimal decision considering all of the firms available within that asset class at any given time.

In order to create a sparse covariance matrix, another alternative explored in the literature is information filtering using graphs or network data. How to build sparse networks that represent information contained in large data sets is an active area of study in various domains such as internet search (Xie et al., 2018), social networks (Berkhout and Heidergott, 2019) and, similarly to our paper, finance (Fan et al., 2013). New applications also continue to emerge, such as social network analysis for link prediction (Zhang and Chen, 2018), recommender systems (Fan et al., 2019), or the study of object interactions in complex systems (Battaglia et al., 2016). In previous work related to ours, starting from a graph representing correlated assets, Onnela et al. (2003) and Cho and Song (2023) showed that investing in the peripheries identified by a filtered subgraph provided benefits for portfolio diversification. They used a static slice of S&P 500 companies, starting at a larger base than previous studies, but the firms again remained unchanged over the long time frame of 20 years that they studied.

In any of these application settings, graph filters aim to maintain the most relevant information by constraining the topology of the graphs. For example, identifying the Minimum-Spanning Tree (MST) is a filtering mechanism for dense graphs that keeps the edges with the highest weights and allows no cycles or loops in the graph (Mantegna, 1999). The Planar Maximally Filtered Graph (PMFG) imposes a different constraint on the graph's topology, requiring it to be planar; i.e., there should be no edge crossing on a plane (Tumminello et al., 2005). Compared to MSTs, PMFGs were found to be more robust for financial market networks as market conditions change, without losing much information content (Yan et al., 2015). Alternatively, Triangulated Maximally Filtered Graph (TMFG) is a more computationally efficient algorithm since, unlike PMFG, it can be parallelised (Massara et al., 2017). In this work, we adopt the latter method for correlation networks of stocks, making them suitable for the large datasets we work with.

As we mentioned earlier, we use the distance correlation measure to account for the strength of the relationship between firms. Alternatively, Diebold and Yilmaz (2012, 2014) developed the connectedness metric, using VAR (Vector Auto Regression) decomposition methods, to measure the pairwise relationship between firms. Their approach can identify individuals or clusters of firms crucial to networks and quantify the direction of risk spillovers. However, these methods have limitations for large networks, such as those for mid-cap firms, due to the amount of historical data they require. Given these limitations, we instead used the distance correlation measure for pairs of firms, and TMFG for filtering, to provide the sparse network which serves as the input to our deep learning-based models. For a more extensive survey encompassing time-series correlations and network filtering in financial markets, we refer the reader to Marti et al. (2021).

3.2.2 Graph Neural Networks

GNNs were first proposed by Scarselli et al. (2009) for node classification tasks. Similarly to recurrent neural networks (RNNs), the first generation of GNNs employed recursion, which they used to learn higher-order representations for a node from its neighbours. As the deep learning field evolved with the emergence of RNNs for sequential data, Convolutional Neural Networks (CNNs) for primarily image processing, and attention-based models (Vaswani et al., 2017) for spatial analysis of unstructured data, non-euclidean data models based on GNNs also developed in parallel. Graph Convolutional Networks (GCNs) borrowed concepts from CNNs, such as kernel filter size and stride, to generate representations for graphs (Kipf and Welling, 2017). They produced state-of-the-art results on popular graph datasets such as citation networks and knowledge graphs, outperforming semi-supervised or skip-gram-based graph embeddings, label propagation and regularisation approaches. Graph Attention Networks (GATs) further improved on these results by introducing variable aggregation of neighbours, and they also proved successful in transductive learning tasks where the data is not fully labelled (Veličković et al., 2018). We refer the reader to a comprehensive survey by Wu et al. (2021) for a general introduction to GNNs and their various flavours.

Thus far, GNNs have been applied in some application areas related to finance or financial markets. For example, they were used in consumer finance for fraud detection (Xu et al., 2021) and credit risk prediction (Óskarsdóttir and Bravo, 2021). Feng et al. (2022) used a combination of a GCN model and self-attention (unlike in a GAT, which has self-attention as part of its own architecture) for making stock recommendations, i.e. to predict the top 3 stocks (out of 738 stocks) for the next period.

Overall, we argue that deep learning methods, such as GATs, are better deployed at scale, on a large set of illiquid, risky firms such as mid-caps, than on a small set of established firms, as they are well equipped to extract complex relationships but need a large enough amount of data to produce stable solutions.

3.3 Methods

In this section, we describe the data used for this study, the distance correlation measure and the graph filtering algorithm, which provide the inputs for the GNN and other benchmark models.

3.3.1 Data

We collected the daily closing prices of all mid-cap companies listed in the US over 30 years, from 1990 to 2021. A total of 16,793 firms were active for at least part of this period. For portfolio selection, we use a three-year rolling window, which limits the allocation problem to around 5,000 firms at any given time point. This number changes substantially over time due to defaults and firms entering or leaving the mid-cap universe, as can be seen on a year-by-year basis from Figure 3.1. Here, default is when a firm has entered liquidation/bankruptcy or experienced another credit event that adversely affected the firm's equity. The default rate varied around 1.5% in any given year; over the full period, 8.5% of total firms faced some form of default.

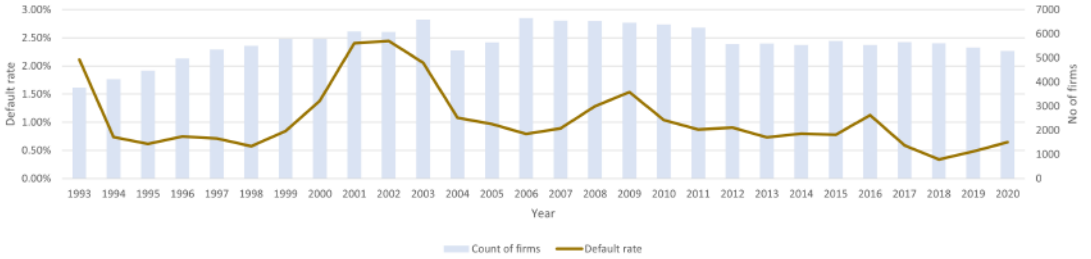


FIGURE 3.1: Number of firms and year-on-year default rate in the sample

We convert the prices to daily returns for each firm, denoted by r_{ut} for a firm u at time t. These return series are divided over time into a training (50%), validation (25%), and test set (25%), with the most recent data serving as the test set. We have more details of this split in Section 3.5.1 To enable capturing relationships between firms, we calculate the return-volatility series using the standard deviation of these returns and a 30-day lookback period. Using return-volatility series is in line with previous work on connectedness by Diebold and Yilmaz (2012, 2014), who, like us, employed return volatilities instead of returns to look for such correlations. In financial market settings, the return-volatility series has some interesting properties compared to the return series, such as exhibiting pronounced co-movements during short risk-off periods but showing weaker relationships under more benign market conditions. This series also tends to have strong serial correlation and closely mirrors market investor sentiment (e.g. being positively correlated with the volatility index, VIX), which are essential for identifying crises (Huang et al., 2019). Those are precisely the circumstances in which the performance of different portfolio allocation strategies can diverge substantially.

More formally, for a firm u at time t, return volatility is defined as $c_{ut} = \sigma(r_{ut}, r_{ut-1}, \ldots, r_{ut-29})$. We let $\mathcal{V}$ represent the total universe of firms, which we observe over N periods, with $\mathcal{V}_t \subseteq \mathcal{V}$ denoting the set of firms active at time t $(t = 1, 2, \ldots, N)$. Using a lookback period of T (set to 3 years) , each firm $u \in \mathcal{V}_t$ has a daily return series,

$$x_{ut} = (r_{ut-1}, r_{ut-2}, \ldots, r_{ut-T}) \in \mathbb{R}^T, \tag{3.1}$$

and a daily return-volatility series,

$$l_{ut} = (c_{ut-1}, c_{ut-2}, \ldots, c_{ut-T}) \in \mathbb{R}^T. \tag{3.2}$$

To prepare the input to the models, we stack the individual return series from (3.1) for all firms in $\mathcal{V}_t$, to obtain the feature matrix $X_t \in \mathbb{R}^{|\mathcal{V}_t| \times T}$, i.e.

$$X_t = [x_{1t} \; x_{2t} \; \ldots \; x_{ut} \; \ldots \; x_{|\mathcal{V}_t|t}]^\top. \tag{3.3}$$

3.3.2 Distance correlation

We quantify the strength of relationship between two firms by evaluating their return volatilities and applying the distance correlation measure. Distance correlation is a generalised measure of dependence, which is capable of capturing non-linear dependencies and is known to perform well in domains such as signal processing (Brankovic et al., 2019) and computational biology (Mendes and Beims, 2018). Starting from the volatility series l_u, l_v for two firms u, v (see equation (3.2) but omitting t for brevity) , our distance correlation measure, $dcor(u, v)$, is derived as follows.

For each firm, we consider the absolute change in volatility between any times i and j over some (lookback) period of length T, and then double-centre the resulting $T \times T$ matrix. Each such rescaled firm-level change matrix can now be compared against the matrices of the other firms, to derive the distance correlation between each pair of firms. More formally, we first define two matrices $A = (a_{i,j})$ and $B = (b_{i,j})$, for a pair of firms u and v, respectively, as

$$a_{i,j} = ||l_{ui} - l_{uj}||$$

$$b_{i,j} = ||l_{vi} - l_{vj}||$$

where $i, j \in \{t-1, t-2, ..., t-T\}$, and $||\cdot||$ is the Euclidean distance. Each such matrix captures times when the corresponding firm has higher or lower volatility compared to other time periods. By doing so across all times, we obtain quantified values of the firm's volatility changes over the observed time period. This sets up a comparison with another firm over the same period. To make the values comparable between two firms with different risk characteristics, we define two further matrices $A^{'} = (a^{'}_{i,j})$, $B^{'} = (b^{'}_{i,j})$, $i, j = 1, 2, ..., T$, that normalise the matrices A, B,

$$a^{'}_{i,j} = a_{i,j} - a_{.j} - a_{i.} + a_{..}$$
$$b^{'}_{i,j} = b_{i,j} - b_{.j} - b_{i.} + b_{..}$$

where $a_{.j}$ is the mean across rows, $a_{i.}$ is the mean across columns and $a_{..}$ is the mean across all values in matrix A (and similarly for B). The distance covariance, $dcov(u, v)$, now is the average over all entries of the element-wise multiplication of $A^{'}$ and $B^{'}$, from which the distance correlation can then be obtained, as follows:

$$dcov(u, v) = 1/T^2 \sum_{i=t-1}^{t-T} \sum_{j=t-1}^{t-T} a^{'}_{i,j}\, b^{'}_{i,j}$$

$$dcor(u, v) = dcov(u, v) / \sqrt{dcov(u, u)\, dcov(v, v)}. \tag{3.4}$$

This measure has some valuable properties that are relevant to our problem (Székely et al., 2007):

1. $dcor(u, v) = 0$, if and only if l_u and l_v are independent.
2. $0 \leq dcor(u, v) \leq 1$, unlike Pearson correlation which instead captures the linear dependence as a number between -1 and 1. This is useful because we are interested in the strength of the dependence rather than the direction of the dependence.
3. The measure can produce a value for two series of unequal length. Given that firms can drop in and out of the universe, resulting in different histories, this feature allows one to nonetheless consider the relationship between them.

A naive implementation of distance correlation calculation has $O(T^2)$ time complexity for a pair of firms. Optimisation methods exist to implement this in $O(T \log T)$ when faced with two series. Still, the main problem is to calculate correlations between many firms and over multiple periods (Huo and Székely, 2016). In our experiments, we chose to implement the required computations in a distributed architecture, using a mix of GPUs and CPUs. Parallelising the pairwise comparisons between firms (using workload manager SLURM and High Performance Computing) facilitates the calculation of large correlation matrices. The resulting code can handle the problem size in a reasonable time.

Writing $d_{i,j}$ as a shorthand for the distance correlation, $dcor(v_i, v_j)$, computed between any two nodes $v_i, v_j \in \mathcal{V}_t$, the resulting dependency matrix, $D_t = (d_{i,j})$, generally leads to a complete graph in which each firm is connected to all other firms.

Some studies have used threshold conditions to remove weaker edges (those associated with small correlation values) from this graph, but the choice of threshold value is arbitrary, and the composition of our portfolio changes over time, further complicating the use of a global threshold mechanism. Instead, we use a more advanced technique from graph theory to filter D_t and remove weaker connections from it.

3.3.3 Graph filtering

The filtering technique we chose is the Triangulated Maximum Filtered Graph (TMFG) method proposed by Massara et al. (2017), which, like PMFG, imposes a planarity constraint on the graph but is more scalable to larger datasets such as ours. Using the topological features of the graph as a constraint, a planar graph retains most of the information with fewer edges. A planar graph can be drawn on a plane (or a sphere) without any two edges crossing. Such graphs have attractive features making them tractable for analysis, by, for example, simplifying cluster or community detection.

We denote by $\mathcal{K} = (\mathcal{V}_t, \mathcal{F}_t)$ the dense graph before filtering, wherein, for any two nodes $v_i, v_j \in \mathcal{V}_t$ $(i \neq j)$, we let $(v_i, v_j) \in \mathcal{F}_t$ if and only if $d_{i,j} > 0$. Using the distance correlation values from Equation (3.4) as edge weights, TMFG filters $\mathcal{K}$ by searching for a (near-)maximal planar subgraph $\mathcal{G}$, i.e., one with the highest possible sum of retained edge weights. Planarity constraints reduce the edges from $|\mathcal{V}_t|(|\mathcal{V}_t| - 1)/2$ in $\mathcal{K}$ to at most $3(|\mathcal{V}_t| - 2)$ in graph $\mathcal{G}$.

The TMFG algorithm grows this planar graph by optimising a chosen score function at each iteration step. In this paper, we select as score function the sum of edge weights between pairs of firms, as given by our distance correlation measure. The procedure starts by identifying a clique of four firms that have the largest such sum of correlations among all firms. A clique is when all the distinct vertices in the sub-graph have an edge between them, i.e, all clique members are connected. This happens in most cases in our volatility networks, as all firms are correlated and unlikely to have a zero value. Next, out of all remaining firms, the algorithm looks for the node (firm) that has the largest sum of correlations (here: distance correlations) with any connected subset of three nodes that are already in the network (so, initially, three within that clique of four) and extends the network with this node and the extra three connections. It keeps repeating this until all firms are added again to the network. Doing so guarantees that the new graph is planar and sparse. To improve computational efficiency, the algorithm maintains and updates incrementally a cache of possible combinations of these sub-graphs. At the end of this process, a sub-graph $\mathcal{G} = (\mathcal{V}_t, \mathcal{E}_t)$, in which $\mathcal{E}_t \subseteq \mathcal{F}_t$, is created.

The resulting graph will serve as input to the deep learning models and graph-based portfolio models. Its planarity property also aids visual representation. Figure 3.2 depicts a sample graph extracted from our data. This shows, for an example time period, the topology of the generated network as being characterised by a fairly small number of central nodes and a large periphery. As we roll over the data window, we construct such a graph for each subsequent time period.

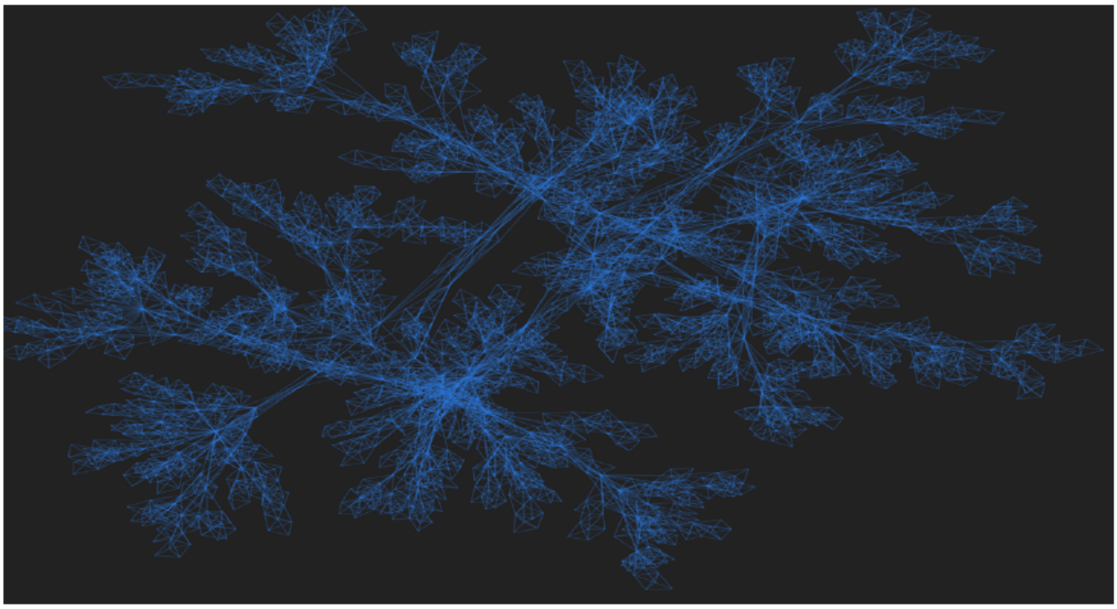


FIGURE 3.2: Snapshot of network of mid-cap firms showing the remaining node connections after TMFG filtering.

3.3.4 Network measures

The graphs created after TMFG filtering could be used directly for portfolio optimisation, as some studies (Li et al., 2019b; Pozzi et al., 2013) have done. Rather than employing deep learning in a subsequent step, these prior approaches basically involve investing in the graph's most 'peripheral' assets. As we believe it is helpful to compare our new GAT-based approach against such a simpler network-index based approach, we next outline an inverse peripherality score, p_i, for each node (firm) $i \in \mathcal{V}_t$ in graph $\mathcal{G}$, which will later be used in section 3.4.2.2 to determine the allocation weights for this benchmark model.

Specifically, we propose a composite score incorporating three common centrality measures: a node's degree, betweenness and closeness. These metrics have been studied since the 1970s (Freeman, 1977) and remain popular in the network analysis literature (Óskarsdóttir and Bravo, 2021). For the first of these, i.e. the degree centrality of node i, we take the number of nodes to which i is directly connected (also referred to as the node's neighbours) relative to the total number of nodes (other than i itself). More formally, having defined the set of neighbours of i as

$$\mathcal{N}_i = \{v | (i, v) \in \mathcal{E}_t\} \tag{3.5}$$

the degree centrality, dc_i, for firm i, is defined as

$$dc_i = |\mathcal{N}_i| / (|\mathcal{V}_t| - 1) \tag{3.6}$$

The second measure, betweenness centrality, can be thought of as an indicator of the amount of activity that passes through a graph node when any changes occur in the network. To measure this activity level, it considers the shortest paths between all pairs of nodes. The betweenness centrality of a node is the fraction of these shortest paths (other than those starting or ending in the node) that pass through that node.

For a firm i, this is defined as

$$bc_i = \sum_{u,v \in \mathcal{V}_t} \frac{s(u, v|i)}{s(u, v)} \tag{3.7}$$

where $s(u, v)$ is the number of shortest paths between (u, v), and $s(u, v|i)$ is the number of these shortest paths that pass through i, with $s(u, v|i) = 0$ if $i = u$ or $i = v$, and $s(u, v) = 1$ if $u = v$.

Our third centrality measure is closeness centrality, cc_i, which, for a given node i, considers the reciprocal of the average length of the shortest paths to all other nodes that are reachable from that node. Hence, for a firm i,

$$cc_i = \frac{(|\mathcal{B}_{it}|)}{(|\mathcal{V}_t| - 1)} \frac{(|\mathcal{B}_{it}|)}{\sum_{u \in \mathcal{B}_{it}} d(u, i)} \tag{3.8}$$

where $\mathcal{B}_{it}$ contains the set of nodes that one can reach from i, not including i itself, and $d(u, i)$ is the shortest distance (in terms of edge count) between (u, i); for example, $d(u, i) = 1$ if there is an edge that directly connects both nodes. In our setting, the TMFG graph ($\mathcal{G}$) still connects all the nodes, so all nodes are reachable from one another, but, unlike in $\mathcal{K}$, the distance now varies. As with the other two measures, higher scores for cc_i imply higher centrality, with values ranging between zero and one.

We use the networkx library implemented by Hagberg et al. (2008) to calculate these three centrality measures. Averaging them produces a simple inverse peripherality (centrality) score, p_i, for each firm i,

$$p_i = (dc_i + bc_i + cc_i)/3. \tag{3.9}$$

3.3.5 Model performance and loss metric

We adopt the Sharpe ratio as the final performance measure for the models (Sharpe, 1966). This is a well-studied metric to measure portfolio performance. Whilst there have been studies proposing further refinements to deal with some of its limitations (Lo, 2002; Farinelli et al., 2008), we use the widely accepted form of the metric. For an individual firm u, we estimate the Sharpe ratio from the sample mean and variance of the returns (defined in (3.1)). The same method also applies to portfolio returns. To produce the latter, we take a weighted sum of the individual returns using the corresponding allocation weights in the portfolio, giving a return series that has a similar format to an individual firm's return series.

Thus, for a series with T daily returns, a firm's mean return and return variance are given by:

$$\mu_u = \frac{1}{T}\sum_{t=1}^{T} r_{ut}$$

$$\sigma_u^2 = \frac{1}{T}\sum_{t=1}^{T}(r_{ut} - \mu_u)^2.$$

From these quantities, the Sharpe ratio, SR_u, can be easily estimated as

$$SR_u = \frac{\mu_u - r_f}{\sigma_u}$$

where R_f is the risk-free interest rate. For ease of calculation, we set this rate to zero as a constant baseline for all. Therefore, the Sharpe ratios produced here cannot be compared to those reported by other studies, but they do allow for a direct comparison between the models in this study. After training, we use them to test model performance and report how they evolve. For the model training itself, we take a different approach outlined below.

Most supervised deep learning models have a prediction target, but in our problem, we do not aim to predict but provide a score for each firm, i.e. the weight to be assigned to that firm in the portfolio. To enable this, we can convert the Sharpe ratio to a suitable loss metric, which the deep learning models will seek to minimise during training. Maximising the Sharpe ratio is equivalent to minimising the negative logarithm of that ratio, which gives us a more convenient loss function. A similar approach was used by Zhang et al. (2020), using gradient ascent and a differentiable Sharpe ratio. The loss function is thus expressed for a portfolio p with a daily returns r_{pt} at time t as

$$LF = -\ln(\mu_p) + \ln(\sigma_p). \tag{3.10}$$

where r_{pt} is calculated from the weights vector $W_t \in \mathbb{R}^{|\mathcal{V}_t|}$,

$$r_{pt} = \sum_{i=1}^{|\mathcal{V}_t|} w_i r_{it} \tag{3.11}$$

3.4 Models

3.4.1 Graph Attention Networks

Different types of GNNs have been developed to learn from graph data. In this paper, we opted for GATs rather than the earlier class of GCNs. As explained in section 3.2.2, the latter were introduced along with convolutional neural networks, which were designed for image processing. Unlike image data, however, graph data are more complex in that they may have a variety of features and node connections that vary in importance. GCNs generate a higher-order representation of input features and neighbours, by weighting the features of each neighbour based on its respective degree centrality. This choice of a single measure rules out more complex weighting mechanisms. GATs solve this problem by using the self-attention mechanism. The latter introduces learnable parameters to generate the weights for neighbours, making GATs more flexible in how they learn from the neighbours' features (Veličković et al., 2018). This is particularly attractive in many real-world settings where, like in ours, the graphs are dynamic and evolve as time passes, or in many financial settings, where market conditions also tend to vary over time. In those settings, fixed weighting of neighbours might not perform well. The mechanism also allows for different features to be learnt through multiple heads, such as short-term moves in one head and longer-term relationships in another. Furthermore, the attention operations are more efficient than alternative approaches, since they are parallelisable across node neighbour pairs.

Here, we formally define the GAT specific to our problem. Given a graph $\mathcal{G} = (\mathcal{V}_t, \mathcal{E}_t)$ as defined in section 3.3.3, and the input features defined in (3.3), GAT transforms the input features X_t into a higher-order representation H_t given by

$$H_t = [h_1\, h_2\, \dots\, h_u\, \dots\, h_{|\mathcal{V}_t|}]^\top \tag{3.12}$$

where $h_u \in \mathbb{R}^{T'}$

Note the dimensionality change from T to T' with this transformation. Specifically, for a given firm u (and, from here on, omitting the current time t for brevity), the transformation function from x_u to h_u, for a GAT with K heads, is defined as

$$h_u = \|_{k=1}^{K} F(x_u, \Sigma_{v \in N_u} a_k(u,v)(W_k x_v)) \tag{3.13}$$

in which N_u, as before, are the neighbours of firm u,. For each head k, $W \in \mathbb{R}^{T' \times T}$ is a weight parameter matrix of the model that is learnt during training, $a(u,v)$ is the weighted importance score of adjacent firm v, $F(\cdot)$ applies an activation function (ReLU), and $||$ is the concatenation operator applied to the outputs of all K heads.

The function $a(u,v)$ is where each type of GNN differs; whereas it was a convolutional function for GCNs, it is attention for GATs,

$$a(u,v) = softmax\,(\sigma(a\,[Wx_u||Wx_v])) \tag{3.14}$$

where σ is a non-linear function (specifically, Leaky Regularized Linear Unit, in short LeakyReLU), $a \in \mathbb{R}^{2T'}$ is a weight vector, and $||$ is again the concatenation operator. In our experiments, we set $T' = 24$ and $K = 8$.

Above, H_t is a higher-order representation of the input features X_t. To convert this representation to a one-dimensional portfolio weights vector, we next reduce the dimensionality by adding a series of learnable layers. We use two blocks of feed-forward networks, which first apply batch normalisation. This normalises the feature inputs in the standard way, using mean and standard deviation, which enables faster convergence and optimisation. We pass this through the first feed-forward network and then apply dropout. The dropout process helps to improve the stability of the training, reducing overfitting. The second feed-forward network uses L1, or LASSO, regularisation to further shrink and eliminate unnecessary weights. We denote the output of each layer i by s_{iu}.

$$s_{iu} = \sigma^{'}(W_{iu} * h_u + b_i)$$

where $i = 1, 2$ are the two feed forward networks, W_i and b_i are their trainable weights and bias terms, respectively, and $\sigma^{'}$ is a nonlinear function (for which we used ReLU).

The s_1 passes through a normalization layer and dropout layer before it reaches the next feed-forward block. The final output $s_2 \in \mathbb{R}$ provides the (as yet unscaled) scores for the starting universe of firms.

We introduce a final allocation layer in the model to rescale these scores to generate the final weights and meet the weight constraints, i.e., making sure that the values range between $[0, 1]$ and add up to 1. This layer is labelled 'Importance Layer' in Figure 3.3. A softmax output function would have been the standard solution to generate these weights, as used by most deep learning papers if they need the outputs to add up to one. However, given the size of our investment universe, softmax would lead to many tiny holdings of firms, which is impractical and can bring high transaction costs. Instead, we prefer to concentrate the portfolio in fewer positions, which reduces the cost of managing the portfolio. To this end, we considered two alternatives. One is to use sparsemax, which generates sparse outputs that would be more suitable to our application scope (Martins and Astudillo, 2016). Another is to introduce a weight reduction mechanism in the final layer of the model, that generates the final weights and uses the output of the feed-forward network s_2 to bring the sparsity with regularisation, thus bringing the weights to what we need and with allocations in fewer firms,

$$w_u = \frac{s_{2u}}{\sum_{v=1}^{|\mathcal{V}_t|} s_{2v}}.$$

We found our mechanism more stable and straightforward to implement than softmax and sparsemax. The training loss function reduction tended to be smoother and more consistent, with the same input producing similar weights over different runs. In contrast, the convergence path with the other training mechanisms was noisier, i.e. the loss over the epochs was more volatile. Note that this allocation layer could be easily extended to meet other portfolio constraints (e.g. choosing the top K firms).

We use standard GATs with their original parameters where possible and implemented them using the Spektral package developed by Grattarola and Alippi (2021). The full model with the added layers is summarised in Figure 3.3. The data embeddings are displayed in colour and the deep learning layers are shown as unfilled boxes in the figure. The GAT layer takes a graph as input. Each node in the graph also has the corresponding firm's return series as node features. For each of these nodes, the GAT model generates an embedding which is further processed by dense layers with non-linear (ReLU) activation, dropout and L1 regularisation, as previously discussed. These scores are then converted to portfolio weights in the importance layer, which collects the scores from earlier steps and, using the reduce-weight mechanism previously described, allocates weights. From this, we can obtain a series of Sharpe ratios for each graph, which we then average to measure the model's performance over the full time period.

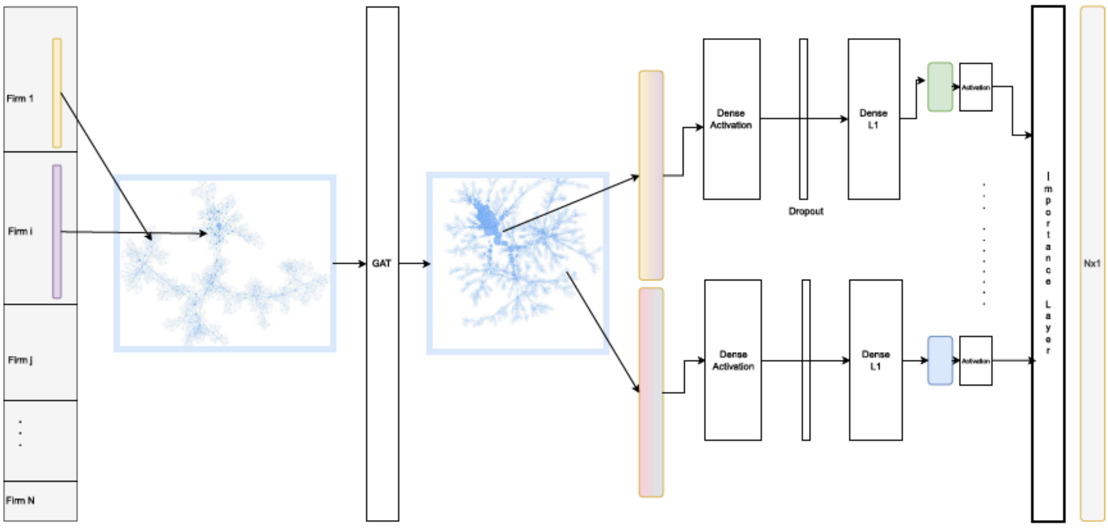


FIGURE 3.3: GAT-based model: First embeddings are obtained from GAT and for each node new representations are created before they are combined to form portfolio weights

3.4.2 Benchmark portfolios

3.4.2.1 Mean-variance model

As discussed earlier, Markowitz (1952)'s mean-variance model is widely recognised as a cornerstone of modern portfolio theory. Therefore, we include the model as one of the benchmark models. For $|\mathcal{V}_t|$ firms active at time t, the model assists in determining the optimal weights w_i for each asset i in the portfolio, thereby requiring that $\sum_{i=1}^{|\mathcal{V}_t|} w_i = 1$. In so doing, one looks for an optimal trade-off between the expected (here, quarterly) returns and the portfolio's volatility.

In its classical implementation, the expected return and the variance of the portfolio, $E(R_p)$ and $V(R_p)$, are estimated from the sample using the sample mean and covariance,

$$E(R_p) = \sum_{i=1}^{|\mathcal{V}_t|} w_i \mu_i \tag{3.15}$$

$$V(R_p) = \sum_i \sum_j w_i w_j cov(i,j) \tag{3.16}$$

where μ_i is the mean return of i^{th} asset and $cov(i,j)$ is the sample covariance between the return series for firms i and j.

We solve the mean-variance problem using an optimisation library that uses quadratic programming (Martin, 2021). Intuitively, the model is expected to have a preference for lowly correlated firms to achieve diversification benefits. A high-dimensional portfolio poses computational challenges, however, which cause our model run times to increase considerably.

3.4.2.2 Network index model

As previously discussed studies have shown, portfolios invested in peripheral assets tend to outperform portfolios containing more central firms. One of our benchmark models, referred to as the network index model, uses the peripherality measure defined in Section 3.3.4 and allocates capital according to the inverse of this score. We rescale these weights so that they sum up to one. Thus, for each node or asset i in the network,

$$\begin{aligned} w_i &= 1/p_i \\ w_i &= \frac{w_i}{\sum w_j}. \end{aligned}$$

The model takes a graph as input at each iteration and calculates the weights as shown above. To test this approach's performance, the resulting portfolio's daily returns are then calculated using the observed individual firm performance over the next three months.

3.4.2.3 Equal-weight portfolio

The equal-weighted portfolio is an important benchmark strategy against which to compare any models with high dimensionality, such as those in our mid-cap universe. The strategy consists of simply assigning equal weights to all the firms in the portfolio. For each firm i in a set of firms $\mathcal{V}_t$, the weight w_i thus corresponds to

$$w_i = |\mathcal{V}_t|^{-1}. \tag{3.17}$$

This allocation may not be practical when developing a portfolio strategy for a large number of firms, as the transaction costs increase considerably. For simplicity, however, we assume that there are no transaction costs[1], and use this equal-weight portfolio as the market benchmark in our study. It is widely reported in the literature that the equal-weighted strategy, is difficult to beat, especially as the portfolio size increases, because the risk of model misspecification error increases for models that use complicated strategies (DeMiguel et al., 2009b).

3.5 Experimental setup

In this section, we elaborate on how we set up our analyses and compared the different models.

3.5.1 Empirical analysis: overview and training settings

At each step of our rolling window procedure, we go through a series of steps preparing the data and building and comparing our different models. This pipeline is depicted in Figure 3.4. All models are measured on the same test period, and using the same set of firms. The equal weighted portfolio, for example, does not need training, but to facilitate a meaningful comparison, this strategy is assigned the same test set as the GAT-based model.

The first stage is the preparation of model inputs, which sees the raw data being converted to returns and volatilities. For each time period, the volatility data is then used to create the dense distance correlation matrix, which is filtered by the TMFG algorithm to create a sparse graph.

The second stage in the pipeline is when the models receive the appropriate inputs for training. The mean-variance model and equal-weighted model are fed the expected returns data. The GAT model, in addition to the return series, also receives a filtered graph as additional input. When training this model, we adopt early stopping to prevent overfitting. A patience of 15 epochs is applied to avoid local minima. Lastly, the network index model is given the graph input, and calculates the peripherality score for each node.

All models generate investment weights at the final output stage, and their performance is measured using (unseen) test data, which is two quarters ahead of training data and one quarter ahead of the validation data (i.e., all models are tested on an out-of-time sample). The Sharpe ratio is calculated at portfolio level for each quarter. These steps are repeated as we slide to the next window of returns, resulting

[1] As discussed earlier, our method would perform well in the presence of transaction costs, since it is designed to produce a sparser portfolio. Later results will show that this no-cost assumption actually favours our benchmarks as their portfolio turnover is higher.

in an updated set of correlation values and a new graph, which are then fed into another model run. We track the performance of each of these series of models over time and report the results in Section 3.6.

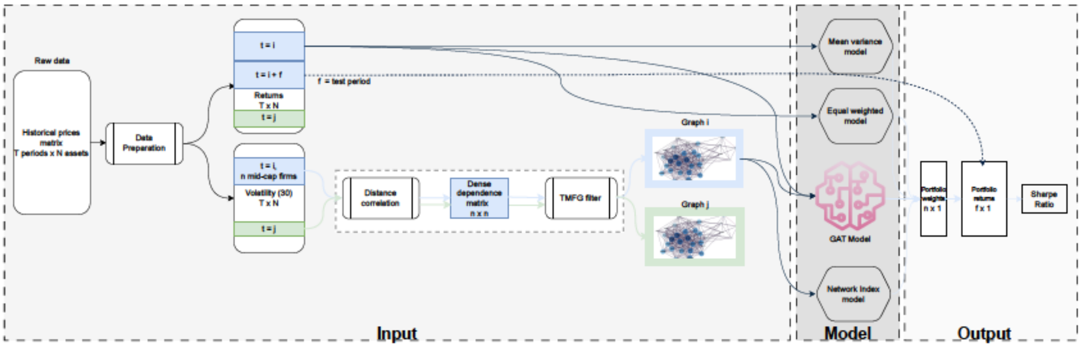


FIGURE 3.4: Empirical setup: overview

3.5.2 Evaluation metrics

The Sharpe ratio provides a suitable measure of portfolio performance. However, to further explore how the respective portfolio solutions differ, we employ some additional metrics.

Firstly, we use two of the node centrality measures discussed earlier, specifically betweenness and degree centrality, to calculate portfolio-level centrality scores. For each of these scores, we take the weighted average over all firms in the portfolio, using the allocation percentage as respective weights. This will show how peripheral the nodes selected by each model tend to be.

Secondly, we compare the industry-level composition of the portfolios, as different strategies might overweight certain industry sectors.We calculate each sector weighting by summing the portfolio weights of each holding in that sector.

Lastly, bearing in mind the dynamic nature of the problem, we also report turnover statistics. In theory, there are four possibilities for each position in a portfolio: either it is newly added, unchanged, closed, or modified (i.e. its allocation increased or decreased), compared to the previous period. Due to the size of the mid-cap universe and its natural turnover, we chose to focus on two types of changes. Specifically, we will consider the newly added or closed positions in a portfolio, relative to the natural rate of change. The latter can be easily derived from the equal-weight portfolio, as this will create or close positions only if the companies are new to or have exited the mid-cap universe, respectively. Thus, for each model, we calculate the number of new (closed) positions in the portfolio, subtract from this the number of new (close) positions found in the equal-weight portfolio, and divide by portfolio size. We then define relative portfolio turnover as the sum of both percentages.

3.6 Results and discussion

We begin this section by reporting the mean performance of each model according to the Sharpe ratio, followed by how this ratio varied over a 30-year period. Subsequently, we explore how the resulting portfolios differed in terms of the peripherality of the chosen holdings, sector distribution, and turnover.

3.6.1 Model performance

Table 3.1 shows the mean portfolio performance of the four strategies, over the training, validation and test data splits. We annualised the shape ratio as we have a horizon of one quarter. On the test data, the GAT models have the highest Sharpe ratio of 1.082. In other words, the GAT models tend to offer better risk-adjusted returns than all of our benchmark methods. This suggests that deep learning models can learn intricate relationships based on the provided input features and the adjacency information derived from the volatility networks, which allows them to beat our benchmark strategies. In line with earlier studies confirming that even well-designed portfolio models find it hard to outperform equal-weighted portfolios (especially when the portfolios are large) (DeMiguel et al., 2009b), we can see the latter having the second-highest Sharpe Ratio, closely followed by the network index benchmark. The Mean-Variance model exhibits the worst performance, confirming similar findings in studies on large-scale portfolio optimisation (Ao et al., 2019). One explanation may be that their inputs, i.e. the expected returns and (linear) covariance matrix, do not have the same amount of information contained in them as do the returns time series and filtered adjacency matrix that serve as the inputs to the GAT models.

TABLE 3.1: Portfolio optimisation performance results

Sharpe Ratio (annualised)	train	val	test
Equal	0.825	0.925	0.830
Network	0.817	0.917	0.820
Mean-Variance Portfolio	0.779	0.785	0.700
GAT	**1.819**	**1.480**	**1.082**

Whereas the rightmost column of Table 3.1 reports the average ratio over all the test data sets, further insights can be gained from plotting a four-quarter moving average of these quarterly ratios (showing the last 12 months' performance). Figure 3.5 details how the models thus perform over time. Apart from an initial period of five years over which all the models appear to perform similarly, this plot shows the GAT models performing consistently well from there on. We can also observe that the mean-variance models, including in the recent past, have underperformed. Although the relative performance gap between the GAT models and the equal-weight strategy and network-index based models appears smaller (especially so near the end of the time span), the GAT models tend to more often have the edge over these two strategies as well. As the market went through several cycles during this extended time period, these findings appear robust to general market conditions and regime changes.

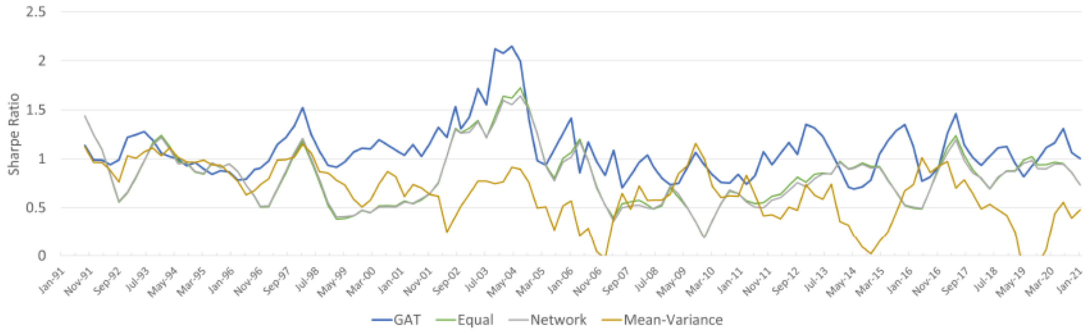


FIGURE 3.5: Model performance over time

3.6.2 Strategy differences: peripherality of holdings

Next, we further examine some of the factors that may explain the performance differences observed above. Firstly, Figure 3.6 shows how two of the (weighted) average centrality scores defined earlier differ between the portfolios produced by each strategy. The set of bars on the left (right) shows betweenness (degree) centrality, respectively. The black lines represent the standard deviation of these scores over time.

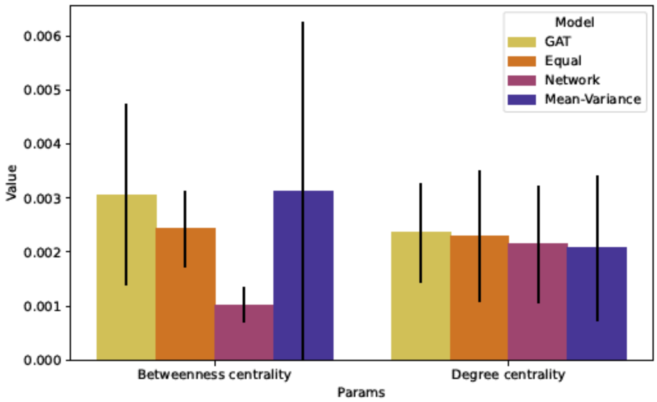


FIGURE 3.6: Weighted centrality of portfolios

Although the GAT and mean-variance models appear similar in terms of their mean betweenness centrality, this score varies heavily over time for the mean-variance model, showing that the latter strategy is undecided in choosing between nodes that are either more or less peripheral. The betweenness scores for the GAT model show much lower variability over time, suggesting that the latter more consistently prefers companies from certain parts of the graph structure. The network index benchmark model chooses the most peripheral nodes according to the same criterion, but the previous section showed that the resulting portfolios underperform against those selected by the GAT model. This could be due, in part, to a relative overweight on peripheral nodes. Also, the earlier chart in Figure 3.1 showed that mid-cap firms carry around 1.5% default risk, which will impact the portfolio quite drastically and might eliminate any possible performance gains from selecting the more peripheral nodes. This is a different conclusion from previous studies, especially those based on topological information for portfolio optimisation (Li et al., 2019b). However, those studies often excluded firms that defaulted over the study period and also tended to remove any firms that do not have the complete data available.

From the degree centrality scores shown on the right, we can draw a similar conclusion; i.e., the GAT model is associated with the smallest variance in centrality. Although the actual means are fairly close, the GAT model tends to select nodes that have marginally higher degree centrality than those in the equal-weighted portfolios or those for the other benchmarks.

To shed further light on this, we revisit the filtered network previously shown in Figure 3.2, adding the portfolio weights allocated by the GAT model to Figure 3.7. This shows the allocations focusing on select branches instead of merely selecting the most peripheral nodes. The darker pink tones show where the model did not allocate any capital at all. As we move higher up in the colour scale, we see larger weights directed towards a few peripheral firms (see dark green tones) and smaller weights distributed across numerous central firms (white and light green tones).

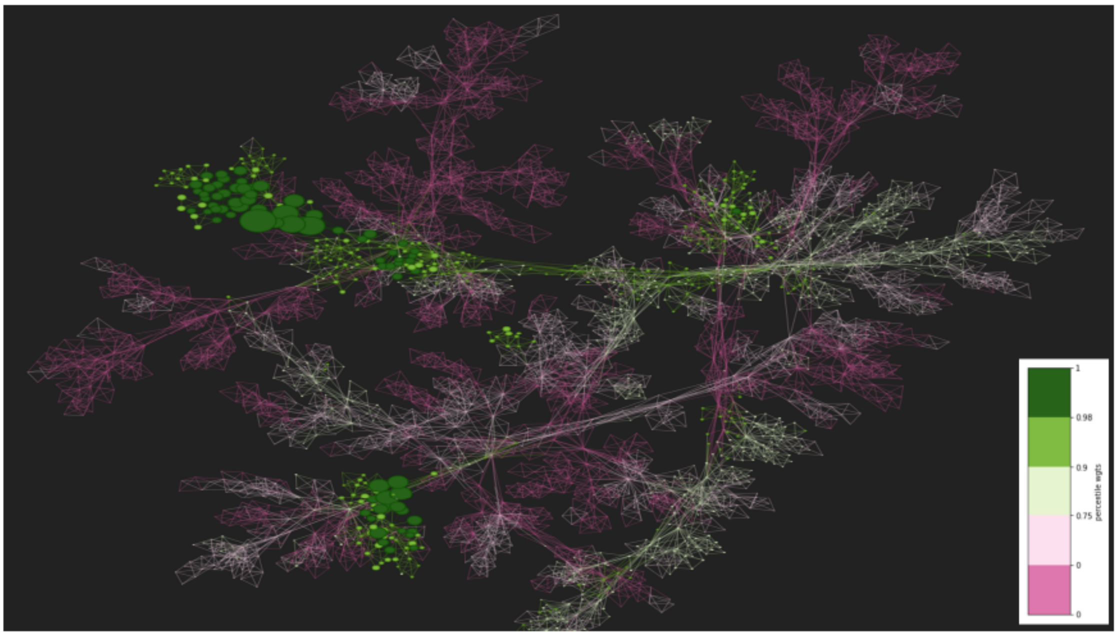


FIGURE 3.7: Distribution of GAT portfolio weights over nodes of mid-cap network snapshot

3.6.3 Strategy differences: sector allocation

Figure 3.8 shows the share of the portfolio's capital that is allocated to each of the different industry sectors. The equal-weighted portfolio (orange bars) provides a useful baseline for comparison, as it shows how the overall population of mid-cap firms is distributed. The largest sector is manufacturing, with around 45% of firms. The network-index model portfolios have a similar sector composition to the equal-weighted portfolio (but as seen in Figure 3.6, they choose more peripheral nodes). The mean-variance model shows greater variance in industry weightings than other models. This is somewhat expected for a large-scale portfolio, as, with the universe of firms changing every quarter, mean-variance portfolio optimisation can be unstable. Here, it tends to underweight the larger manufacturing sector and overweight the transportation and public utilities sectors. As seen from their relative variation in industry allocations, the GAT model, over time, rebalances sector weightings more extensively than the equal and network-based portfolios, but less so than the mean-variance model. This may put the GAT model in a better position to achieve stable returns under different market conditions.

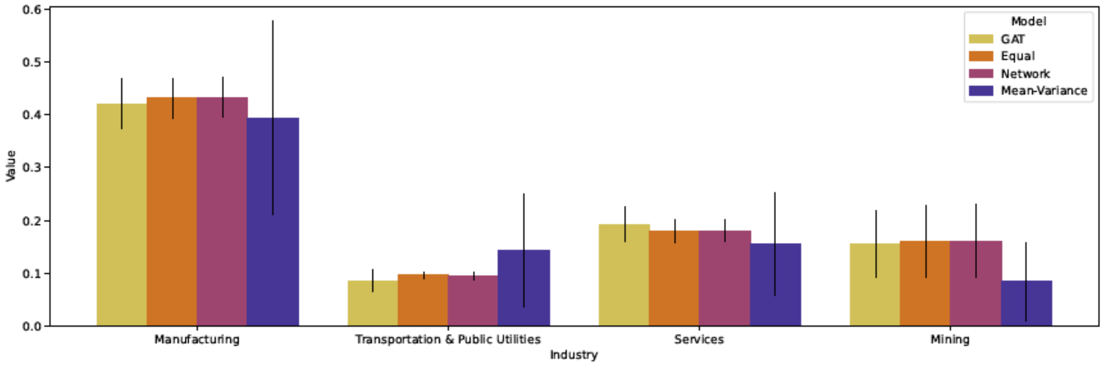


FIGURE 3.8: Industry allocations per model

3.6.4 Strategy differences: portfolio size and turnover

To better understand some of their cost implications, we end the analysis by examining the resulting portfolio size and turnover for each strategy.

Firstly, to help us compare how sparse the portfolio selections are for the GAT and mean-variance models, Figure 3.9 plots, for each time period, the percentage of firms from the corresponding universe to which no capital has been allocated. Note that the network index and equal-weight strategies were left out from this chart, as both will assign non-zero weights to all of the firms. As can be clearly seen, the GAT model requires holding far fewer firms than our mean-variance implementation, which is unsurprising given that the former includes a regularisation mechanism that is lacking in the latter. The resulting difference in unallocated firms is substantial and consistent over time. The changes over time observed for the GAT model may be indicative of changing market conditions.

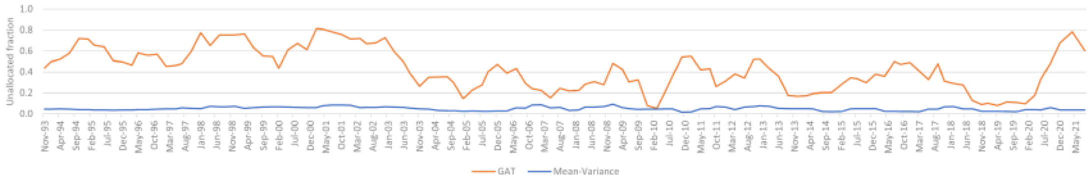


FIGURE 3.9: Fraction of firms that do not have any capital allocated, over time

Secondly, we evaluate the quarterly turnover of each strategy. Unlike the capital allocation reported earlier, this relates to the count of positions changing between periods (or number of trades). Figure 3.10 shows, for the different types of portfolios, the mean proportion of holdings that were new positions, as well as how many positions were closed since the last quarter as a percentage of portfolio size. Turnover is the sum of these two, represented by the combined bar height. The equal-weight portfolio sees 16% turnover on average, with some further variability indicated by the error bars. This reflects the natural churn as companies enter or leave the universe.

Among the other methods, the GAT model has the lowest turnover, albeit with a higher variance than the network model. This means that, on average, it requires fewer trades but, depending on market conditions, there are some periods where it uses substantially more than the network index model.

Table 3.2 summarises how much of each model's turnover is in excess of the natural change (by subtracting the turnover of the equal-weight portfolio). From this, we can again see that the GAT model requires little such excess turnover compared to the other strategies. For example, the portfolios produced by the GAT model, on average, add a mere 2.07% more new positions from one quarter to the next, whilst they close an additional 2.03% of existing positions (compared to 2.83% and 2.68% for the mean-variance portfolios). With an overall excess turnover of 4.10%, the GAT model is

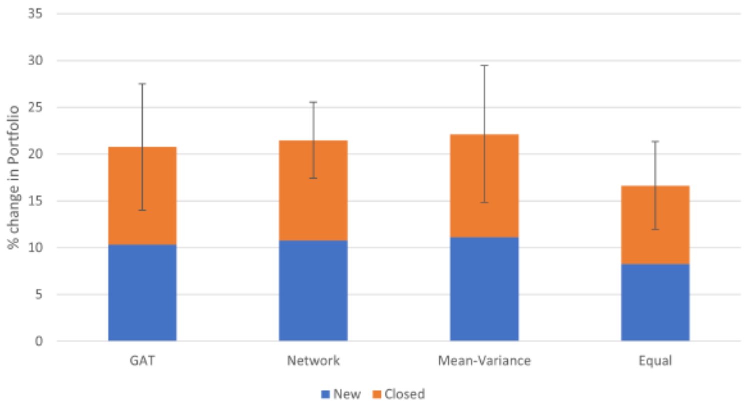


FIGURE 3.10: Trades in the portfolio as a % of total firms

thus associated with a relative reduction of 25% and 15%, compared to the mean-variance and network index models, respectively. This strongly suggests that the GAT model will have lower transaction costs compared to traditional models.

3.7 Conclusion

In this paper, we have put forward a solution for large-scale portfolio optimisation using deep graph learning. We have seen how GAT-based models, a type of model within the broader family of deep learning models, can extract intricate relationships that other traditional models can not. While most studies focus on portfolio optimisation for assets that have regular availability of prices, we focused on problems where the data is more difficult to model, by choosing to analyse the volatile mid-cap market. The study of such firms in itself is of interest for a variety of reasons. Whilst they are far more numerous than their large-cap counterparts, their smaller size makes them more vulnerable to market movements and larger correlation with the overall economy, something most structured stochastic models ignore. They also potentially offer higher returns for commensurate risk to investors, while better risk management may lead to better access to financial markets for those firms.

In designing our approach, we have linked several areas of study. We applied the distance correlation measure to firm volatility pairings, to capture more complex connections between firms than with alternative approaches. From this, we generated a sparse graph by employing the Triangulated Maximally Filtered Graph algorithm, a filtering technique that is applicable to large-scale graphs. Through this, we explicitly incorporate the interdependence of midcap companies. These filtered graphs were then presented to a GAT model, which can identify higher-order relationships. The final allocation layers of our deep learning solution were designed to optimise the embeddings generated by the GAT models, and the regularisation parameters used in the deep learning models imposed constraints on possible weights and the number of firms to which capital can be allocated. Being derived from the Sharpe ratio, the chosen loss function set a risk-adjusted return objective for portfolio performance maximisation. Other portfolio objectives could similarly be used and further constraints imposed on the portfolio allocations.

TABLE 3.2: Mean (excess) turnover by model (in excess of natural change)

% change	new	closed	turnover
Network	2.48	2.36	4.84
Mean-Variance Portfolio	2.83	2.68	5.50
GAT	**2.07**	**2.03**	**4.10**

Starting from the premise that deep learning models should be adept at optimising many high-dimensional problems such as ours, our experiments with real-world midcap data indeed showed that the GAT-based models achieved better performance than other alternatives. We also studied how these results were robust to different market conditions and looked at the distribution of firm allocations across different strategies, to identify some of the factors explaining how the GAT models differed. In so doing, we found that they tended to choose companies that are not too much in the periphery and allocated capital to fewer firms. Lastly, we observed that the typical turnover associated with the GAT models was lower than that of the alternatives, although, on fewer occasions, they did make more substantial changes to reposition the portfolio.

As for future work, further graph neural network models could be developed to predict aspects like market regimes, or produce early-warning indicators for financial networks. By changing the objectives and revising the loss function, we could also extend deep graph learning-based portfolio optimisation models towards different goals, such as the construction of Environmental, Social and Governance (ESG) portfolios, or other types of diversified portfolios, and solve problem instances on a much larger scale than before.

Acknowledgements

This work was supported by the Economic and Social Research Council [grant number ES/P000673/1]. The last author acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [Discovery Grant RGPIN-2020-07114]. This research was undertaken, in part, thanks to funding from the Canada Research Chairs program.

The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

Chapter 4

Network-enhanced credit risk models for SME credit lines

Abstract

Credit lines are borrowing facilities that firms can draw from up to an agreed limit. They are used primarily for liquidity management, as they offer flexibility within a set credit agreement. For banks, these lines of credit also create procyclical risk, as the exposure tends to increase during challenging economic conditions. As individual firms draw from these lines and signal their financial health through their repayment behaviour, this creates behavioural patterns that could signal the firm-level risk of default and deterioration in the wider market environment, thus impacting other firms. However, the prevalence of instalment-based products results in them being less widely studied than other debt instruments, such as bonds or loans. This paper uses credit line data from a large database of Small and Medium-sized Enterprises (SMEs). We generate a large set of behavioural features related to credit line usage from temporal data. We also create a dynamic network of all firms using explicit inter-firm transaction data, ownership data, and the financial transactions due to explicit supply chain relationships between firms. This allows us to capture how default risk may propagate between firms. Alongside these, we also use financial information and credit agreement terms as further inputs to our model. For the behavioural and financial information, we create panel data for each firm and use it as input to temporal deep learning models, specifically graph attention networks. Using a multimodal architecture, a deep learning model that can incorporate different data sources effectively, we combine the temporal and graph models to produce a one-year probability of default for a given firm. To gauge its performance, we compare this model with logistic regression, traditional machine learning models, and baseline deep learning models. We also seek to understand the predictive power of behavioural data by comparing with other data sources.

Keywords

Credit Lines, SMEs, Deep learning, Graph neural networks, Graph attention networks

4.1 Introduction

Small and Medium sized Enterprises (SMEs) contribute significantly to any economy. What constitutes an SME differs across countries and regions; in Europe, these are firms with revenues of less than 50 million euros and fewer than 250 employees, while, in the US, the definition changes by sector, but generally, they should have no more than 500 employees. While exact definitions may vary, the structural features of these firms are similar. They typically provide around 60% of the employment in most countries. In Europe, for example, they are reported to account for 65% of employment and contribute 52% of the value generated by all firms (Di Bella et al., 2023). Even though they play such an important role in an economy, they face higher barriers regarding access to credit, compared to individual consumers or larger firms (Ayyagari et al., 2007; Rao et al., 2021). The process used by lenders to score SMEs sits somewhere between the personalised approaches used for large corporates and the fully automated credit approval process that is in place for consumers. This dual approach adds to the cost for the banks and financial intermediaries that provide SME credit (Munro, 2013). The lack of structured data, ownership issues, legacy banking systems, and limitations of the models themselves also contribute to this problem (Beck and Demirguc-Kunt, 2006; Moscalu et al., 2020).

Revolving credit facilities (RCF) or credit lines are a form of credit provided by financial institutions to manage the liquidity needs of a firm. The firm can borrow up to a pre-agreed limit for a fixed period. Interest is paid on the drawn amount, and likely commitment fees may be paid on the undrawn amount. This allows a firm to avoid going through a time-consuming loan sanctioning process whenever it needs to meet its cash needs. As some firms operate in a cyclical business and due to cash flow delays that are inherent to their supply chains, most firms would need some revolving credit arrangement with their primary bank (Sufi, 2009). The banks, too, benefit by providing a credit line facility, as this provides behavioural insight into their client firm, while earning them fees and interest. Customers that appeared initially risky might, over time, be better understood after they start using the facility, especially if there is strong seasonality in the cash flows or a longer working capital cycle, i.e., the time it takes to convert working capital into cash revenues. This helps the banks to better serve their clients and provide other forms of credit, such as loans (Aragon et al., 2020; Acharya et al., 2021). However, these instruments also carry risks, as studies show an increase in revolving line exposure in the period prior to default (Bergeres et al., 2015; Berrospide and Meisenzahl, 2022).

For every granted loan or credit line, banks must allocate a provision, which covers expected losses, and use part of their capital to account for unexpected losses. The capital for a specific facility or a loan is a certain percentage of Expected Loss (EL), the latter of which is obtained by multiplying three factors: the loss Given Default (LGD), i.e. the complement of the expected recovery rate after default has occurred, and represented as a percentage of par; the Probability of Default of a firm (PD), generally a one-year ahead probability of default; and the exposure at default (EAD), the amount of monetary exposure of the bank to the defaulted firm, usually the full amount owed plus an extra adjustment related to the expected utilisation of the remaining credit line when the borrower is under stress. These three quantities need to be modelled by the financial institution. Although there is little reported work on PD modelling for revolving credit facilities, the EAD has received some attention. For example, Wattanawongwan et al. (2023) and Tong et al. (2016) build EAD models for retail credit card data and Thackham and Ma (2019) provided a similar study on EAD for large corporates. Our paper covers an existing research gap by both focusing on the PD of revolving credit while also tackling the challenging problem of SME lending.

To do so, we use the behavioural data of revolving credit facilities, which includes features like timeliness in repayments, utilisation, and historical arrears. Behavioural data comprises dynamic variables that track financial signals over time, which is not possible for application scoring, where the data is largely static (Bellotti and Crook, 2013). Behavioural scoring of existing clients can, for example, be used to make decisions on setting credit limits as a firm's situation changes or to put firms into different risk grades for capital calculations (Sohn et al., 2014). We further combined this data with traditional accounting information and the profile of the firms, such as business sector and location, to form a rich tabular set of data that can be used to estimate the one-year ahead default probability.

Apart from such tabular data, alternative data sets are also increasingly available. These, too, may provide useful indicators of a firm's risk, especially when traditional data sources are not available (Owens, 2017). Financial institutions have large swaths of internal data on customers such as SMEs, which are either collected directly, sourced externally, or generated internally due to customer interactions. SMEs, in particular, are benefiting from increased access to credit provided by fintech firms who are tapping into alternative data sets (Lu, 2018). Networks are a prominent type of data source, and relationships like common ownership, supply chain partnerships, or transactions can be used to connect a group of firms. Using networks as a data source gives additional insights into how risks are propagated and can capture inter-firm dynamics that cannot be explained with just firm-level data (Cainelli et al., 2012). For a lending institution, this kind of network view could provide early warning of how one firm's vulnerability could impact other firms in its portfolio, and also provide insights on correlation risks that exist in the portfolio (Giesecke and Weber, 2004).

Network data have also been used in the context of SMEs and were found to improve the models' performance when network metrics are added to the input features (Song et al., 2016; Vinciotti et al., 2019; Óskarsdóttir and Bravo, 2021). However, most works have focused on implied networks, or networks derived from tenuous relationships, generally within a small network or a single type of network. This is due to the limited availability of data, since SME datasets are private to an institution, and where public data is available, it might be outdated. More timely data with extensive coverage is needed to study the importance of networks for SMEs more fully.

For this, we partnered with one of the largest banks globally, who granted us access to both behavioural and network data. We focused on default prediction for SMEs that were previously granted credit, all within one major European economy. The data is extensive, with more than 150 thousand SME firms and 3.8 million connections. It contains monthly snapshots of contractual data, which can be used to generate behavioural features, and network information based on three different connection sources: firm ownership, supply chains and transaction information. Among these, ownership links remain mostly static, whilst transaction networks are highly dynamic. The complexity of the explicit inter-firm links revealed by these three connection types, their dynamic nature, and the scale involved, as well as their intended use alongside the available behavioural data, calls for advanced computational analyses, such as those introduced in the realm of Artificial Intelligence (AI).

AI methods, and especially deep learning, have produced state-of-the-art results in many problem settings. They have been developed to use various data types as inputs, including images, video, audio, text, and networks. Through multimodal learning, these models combine these different data sources to generate new integrated features, further improving the benchmark results in various domains (Ngiam et al., 2011). Pre-existing methods were not ideally suited to using such data, as they required making modelling decisions as to how to transform the inputs into quantitative signals, without any assurances such inputs would prove useful. Similarly to earlier studies of SMEs that also used network data (Lazo et al., 2021), we could choose to define a few such explicit features ourselves, in an attempt to represent the network, but deep learning models can automatically generate large embeddings that could be better tuned to the problem and help improve the model performance. To extract the higher-order relationships in the network data, we propose building upon graph attention networks (Veličković et al., 2018), a deep learning architecture that is designed specifically for network data. These models are scalable to large networks but we need to make them suitable to handle the large, dynamic networks we have, and integrate them into a multimodal architecture that is able to use the behavioural data as well. Note that, unlike some previous studies on consumer credit risk that also used deep learning models (Mercep et al., 2021) or Graph Attention networks (GAT) models (Wu et al., 2023), this paper focuses on SME credit risk, which traditionally needed more qualitative inputs than consumer models as relevant risk drivers proved more challenging to quantify.

Deep learning models, however, tend to be more difficult to interpret, which may be an issue if the model will be used to screen applications and/or is subject to particular regulatory scrutiny. With behavioural scoring, however, the model focus is on internal monitoring and capital allocation after the borrower firm has already been granted a credit line. In this study, we look for a middle ground whereby the resulting models are at least partially explainable, whilst the (opaque) network embeddings generated by the deep learning models lead to demonstrably more precise predictions than traditional models.

Summarising how we seek to fill current gaps in the literature, our study looks to predict the one-year ahead probability of default of SME firms that were granted credit lines, by employing behavioural and network data held by financial firms. The proposed approach is particularly salient when traditional data sources, such as timely audited financial statements, are difficult to source or do not carry strong signals. We feed these different data sources to a multimodal GAT, to produce the final outputs, and test our methods on a large real-life dataset. The paper thus aims to answer the following research questions:

1. Can SME default be predicted effectively using behavioural and network data?
2. Are the proposed deep learning methods suitable for drawing insights that could be useful for other, more traditional models, when those are preferred?
3. Are the predictions stable over periods of shock to the system?

In terms of contributions made, this paper is the first to introduce multimodal dynamic deep learning techniques to leverage multiple, explicit SME networks and empirically compare the relative predictive power of behavioural and network data in the context of SME credit lines. It does so with the help of a large, real-world dataset from a major lender, exploring the robustness and predictive capacity of the proposed solutions over time.

The remainder of the paper is organised as follows. Section 4.2 reviews relevant literature on credit lines, networks and deep learning, with a focus on SMEs. Section 4.3 describes the data and the process by which this data is converted into networks and relevant measures used in the study. The proposed models, and the baseline models against which they are compared, are described in Section 4.3.2. Section 4.4 then presents the results of the various models applied to different data types. Finally, Section 4.5 summarises the contributions and suggests future steps.

4.2 Literature review

There is extensive literature on credit risk modelling for established firms and individual consumers, much of which proposed various techniques and benchmarked them across different datasets. We refer the reader to Crook et al. (2007), Lessmann et al. (2015), and Shi et al. (2022b) for detailed surveys of the main work in this area. Instead, we focus mainly on the credit lines literature and the rationale for those studies, whilst highlighting some of the differences with our own. Next, we review existing studies in the credit risk domain that consider network data within the specific context of SMEs or credit lines. Thirdly, we discuss some related deep learning work.

4.2.1 Credit lines

Earlier studies on credit lines, or loan commitments as they are called in relation to large corporates, show the need for a separate stream of study as they behave differently from instalment loan products yet make up an important part of a bank's risk exposure (Shockley and Thakor, 1997; Martin and Santomero, 1997). Part of the appeal of credit lines to banks is that they reduce information asymmetry between lender and borrower, as banks can monitor the risk actively, especially in the case of SMEs. Sufi (2009) found evidence that credit lines provide liquidity support to the firms, and not having a credit line is a powerful measure of financial constraints to credit access compared to other indicators. A recent study by Chodorow-Reich et al. (2022) compared drawdowns by large and small firms during the Covid-19 crisis, identifying several obstacles faced by SMEs. These studies provide a clear rationale for further work on SME credit lines that could alleviate some of these constraints.

From a risk management perspective, rich behavioural data can be derived from credit lines, providing signals that could be applied to other relationships with the borrower firm, such as loans or underwriting services. In the case of SMEs, though, these are challenging to manually monitor, and, instead, automated strategies must be developed to assist the bank (Munro, 2013). This is precisely what this study sets out to do, by extending the current methods for default risk management of a large-scale portfolio of credit lines.

Through the large amount of undrawn credit in the system, credit lines or other forms of revolving credit can transmit certain macroeconomic shocks. Greenwald et al. (2020) studied different types of shocks and how they are transmitted. They found that normal business cycle shocks are absorbed by these products, but extreme shocks such as Covid-19 provided interesting conclusions in that large firms could crowd out smaller firms. Acharya et al. (2021) established the systemic importance of these facilities, as they also function as liquidity insurance for the firms when there are shocks to the system. They find credit lines are more cyclical than loans. The credit amount drawn as a ratio of available credit increases ahead of recessionary periods, and therefore could provide an early warning to the banks.

To fully quantify the credit risk of credit lines (or any other forms of revolving credit), one has to estimate not just default risk but also exposure risk. In addition to requiring banks to model the PD, the Basel II Accord mandates estimating Credit Conversion Factors (CCF) to model the EAD parameter for credit lines. Tong et al. (2016) used retail credit card data and studied models with and without the CCF to understand EAD for revolving facilities. Credit cards are similar to credit lines for firms, where there is an agreed limit, and it is up to the borrower how to utilise the credit limit. Thackham and Ma (2019) provided a similar study on EAD for large corporates and found that both exposure and limit need to be jointly modelled to derive the EAD correctly. They also found evidence of active risk management by lenders decreasing the limits to risky firms, as well as weak evidence of countercyclicality whereby exposure to risky firms was reduced ahead of recessionary periods. Bergeres et al. (2015) studied the dependence between exposure size and default risk of two different retail products, term loans and credit lines, and found that higher drawdowns are indicative of higher loan default risk and vice-versa. A recent study by Wattanawongwan et al. (2023) used a large set of credit card defaults to build a series of regression models for predicting EAD. They noted that the EAD risk drivers differed depending on how close utilisation had come to the credit limit prior to default.

In this paper, we chose not to target exposure risk but default risk, using prior utilisation information as part of the input to a PD model. Although there is a large body of work on default modelling for term loans and consumer loans in particular, there has been a surprising lack of work reported in the academic literature that focuses specifically on corporate credit line defaults. Some empirical work in the setting of consumer credit cards has focused on setting limits based, at least in part, on model-based estimates of default risk. For example, Alfonso-Sánchez et al. (2024) used reinforcement learning to adjust credit limits and found the optimal policies are non-trivial but can be now automated with such models. Sohn et al. (2014) take a multi-step approach by first predicting the risk of default and then grouping similar credit lines for profit maximisation. Stress testing credit card default using survival models is another related area of research. In so doing, Bellotti and Crook (2013) were able to include both behavioural features and macroeconomic conditions into their models, which resulted in better forecasts. As an example of related work on SMEs, Calabrese et al. (2016) used a novel generalised additive model to predict bankruptcy in a large set of Italian SMEs, and found that it was important not to assume linearity of the effects of explanatory variables. Their model, however, solely employed accounting data and did not consider credit line behavioural data, as we do in our work.

A second key focus of our paper is how to incorporate data on inter-firm networks into the default prediction models.

4.2.2 Networks in credit risk

Networks are found across a large variety of settings. The internet, social networks, financial networks or supply chains are some of the important networks we encounter regularly. There is considerable value to be unlocked from these networks, as, for example, the PageRank algorithm developed by Brin and Page (1998) showed. A large body of literature exists on various kinds of networks, as surveyed by Bricco and Xu (2019) and Chunaev (2020). In this paper, we focus on SMEs, which are part of much larger economic networks and, as economic actors, can be co-owned, and frequently transact with other firms, all of which create rich networks.

One of the earlier examples of SME network studies by Naudé et al. (2014) analysed a small survey sample including data on the social networks of SME owners. Using structural equation modelling, they showed that the performance of SMEs is influenced by the network structure and external networking behaviour of their owners. Song et al. (2016), again with the help of survey data, considered the supply chain network attributes of SMEs and analysed the relationships between different types of inter-firm ties, information sharing, and the credit quality of SMEs. Other studies have sought to embed network information into traditional models, thus aiming to improve the performance of those models. Vinciotti et al. (2019) were provided access to transaction data on a sizable set of SMEs in the UK, from which they generated a selection of network features that could be added to their credit risk models. Their analyses showed that these network-augmented models performed significantly better than models that merely used traditional accounting or other structured data. Another study using an agricultural loans dataset developed a multi-layer bipartite network and used that to create a novel personalised PageRank centrality measure (Óskarsdóttir and Bravo, 2021). This measure was then added to the existing feature set, which again was shown to improve the credit default prediction performance.

Similarly to Óskarsdóttir and Bravo (2021), we also use a page-rank centrality measure in our study, as one of the network features. The core difference between Vinciotti et al. (2019) and our paper is that we consider three distinct sources of network data and seek to leverage more complex patterns embedded in them, using state-of-the-art deep learning methods.

4.2.3 Deep learning and graph neural network applications

Deep learning methods have produced state-of-the-art performance in a wide range of domains such as speech recognition, drug discovery, or language translation (LeCun et al., 2015). Applied to large data sets and using large-scale computations, they are able to discover intricate relationships that remained hidden with earlier models. Like for other types of data, such as text or imagery, deep learning has delivered dramatic advances with network data as well. Graph neural networks (GNNs) are a family of deep learning models specialised in using network data. Among them, Graph Attention networks (GATs) have achieved state-of-the-art results on various graph or network data sources. GATs leverage self-attention over the node features, which assigns different importance to the nodes in the local neighbourhood, instead of depending on the global structure, which was typical of earlier GNN methods. This reduces the computational complexity and was also found to improve the performance of the models (Veličković et al., 2018). They are thus suitable to large-scale graphs and can be deployed without excessive computation requirements. We employ GATs to generate embeddings for each firm of interest, from the available network data and behavioural data. These embeddings then serve as inputs to other deep learning models that predict default probability.

Deep learning has recently attracted growing interest in credit scoring research. Using behavioural data and applying deep neural networks improved the performance of the credit ratings of a large loan portfolio (Mercep et al., 2021). Sun and Vasarhelyi (2018) deployed neural networks on credit card behavioural data to predict delinquency or default and suggested they are suitable for automating part of the credit risk assessment. More recently, Ala'raj et al. (2021) used Long Short-Term Memory (LSTM) models, a sequential deep learning model designed to handle temporal data, applying them to behavioural data on retail credit card usage. Their analyses showed that they significantly outperformed traditional methods on this data. In a related retail banking setting, deep learning methods have also been found to improve credit card fraud detection capabilities (Alarfaj et al., 2022).

GNNs, too, have been deployed for credit default prediction, albeit not for SMEs. A first class of GNNs, graph convolutional networks, were developed along with convolutional neural networks used for image processing. Unlike image data, however, graph data are more complex because they may have various features and node connections that vary in importance. Graph convolutional networks, a subclass of GNNs, generate a higher-order representation of input features and neighbours by weighting the features of each neighbour based on its respective degree of centrality. This choice of a single measure rules out more complex weighting mechanisms. GATs, another subclass of GNNs, solve this problem by using the self-attention mechanism. The latter introduces learnable parameters to generate the weights for neighbours, making GATs more flexible in learning from the neighbour's features (Veličković et al., 2018). Zandi et al. (2024) proposed a three-stage model using graph attention network embeddings as inputs to an LSTM model to capture temporal relationships. The temporal outputs of the LSTM are then passed through an attention model to predict default. Their networks are multi-layer networks derived from a well-known US mortgage loans dataset, and formed using geography and the mortgage providers as the relationship variables.

Furthermore, this is particularly attractive in many real-world settings, like ours, where the SME networks are dynamic, showing significant evolution as time passes. Macroeconomic conditions also tend to vary over time. As a result, fixed weighting of neighbours may not perform well, which is what happens with graph convolutions. The attention operations are also more computationally efficient than alternative approaches since they are parallelisable across node neighbour pairs. Different features can be learnt through multiple heads, such as short-term moves in one attention block head and longer-term relationships in another. The GAT model was also employed by Wu et al. (2023) to predict credit card defaulters using a transaction data network. Elsewhere, GCN models were used in combination with the publicly available Lending Club data set of peer-to-peer loans, to create networks using similarity measures for application, history or soft information such as location (Lee et al., 2021), or using a composite similarity measure (Li et al., 2024). Compared with traditional models, GCN models were found to perform better, especially when all the networks are used as inputs.

Unlike the above applications of GATs to consumer credit, the present study turns to SMEs and the credit lines they use. SMEs are more complicated as their risk profile changes with their size, and they are traditionally more complex to service compared to consumer credit. We expect deep learning models to improve model performance in the credit monitoring phase that we are chose to focus on. Next, we outline our approach and how we set out to test it.

4.3 Methods

This section describes our data, the models we built, and how we set up our experiments.

4.3.1 Data

To conduct the study, we were provided access to an extensive SME credit line portfolio of a major European financial institution which is part of the globally systemic important banks (FSB, 2024). From this, we sourced monthly data ranging from February 2018 to March 2021. The aim was to predict the one-year ahead probability of default for the firms that have an active credit line contract. We highlight some data features in Appendix B.1. We also show a summary of the data in the form of charts for defaults and network data in Figure 4.3 and Figure 4.1.

The data contains five kinds of information, which we group into static, contract, financial, location and network categories. The first four fall into the structured information category. The static information is generally collected before the credit line contract is established with the firm and contains information (in anonymised form) such as the location, number of employees, and any external information supplied at the start. The contract information includes the specifics about the credit line, such as the limit (i.e. the maximum drawn amount set by the bank), which, combined with the behavioural variables collected by the bank, produces some extra behavioural features related to utilisation. The third category is the financial information from the balance sheet and income statements that are filed by the firm. These are annual statements and have a one-year lag. As the focal firms are SMEs, some of this information is missing, since reporting requirements could vary depending on the size and age of the SME. For example, small and new firms might not need to have standard accounting statements or could report more infrequently. Next, the location information corresponds to where the SME is located, which is again collected anonymously, meaning the connections are made through meaningless codes. Finally, we have the network information, which provides detailed insights into how individual firms are connected.

4.3.1.1 Network characteristics

Generally, a network consists of a set of nodes and edges that represent relationships between the nodes. We will denote the network as $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$, in which $\mathcal{V}_t$ are the nodes and $\mathcal{E}_t$ is the set of edges. In our study, it is the firms for which we need to produce the probability of default that make up the nodes. The edges are based on three different types of relationships: 1. ownership, 2. financial transactions between firms, and 3. supply chain-related transfers. First, data about firm ownership, i.e. who owns each firm, allows us to link firms that share at least one owner, which could be an individual or another firm. The transfers represent factoring, invoicing or supply chain relationships, all grouped into a single edge type. Factoring involves a relationship between client firms and typically another financial institution, that help the firm realising cash flow earlier than the payment date of the invoice; here, firms with the same financial institution relationship will have a connection between them. Invoicing is done when a firm in the supply chain is contracted with another firm, which links the firms as having an invoicing relationship. Financial transactions are monetary payments that flow from one firm to another. In this, the transaction data can resemble the supply chain-related transfer data, but it can also include other payments, such as transfers to payroll firms, allowing for a richer view of the supply chain. Transfers and transactions are daily data, which we aggregated to monthly information. All this data is also not uniform, with different start periods and a few months where the data is missing. We used a six month lookback period in this study. Using this specific window allows us to assemble the network without having to impute data, capture the ownership information and handle different starting periods effectively. A particular feature of the ownership network is that data is missing in a few months, but when we do have data, it is stable with a similar amount of monthly information. We combine all of the information on ownership, transactions and supply-chain transfers, for each pair of nodes. If the same pair of nodes have multiple transactions or transfer information in the past six months, we aggregate the transfers into two edge-related variables: total amount and total number of transactions. If there are multiple owners between the firms, we model this as just one edge between the firms, but the edge attribute is calculated as the number of shared owners.

Figure 4.1 shows the summary statistics for each month. The number of firms is shown as a blue bar chart, with values plotted against the right-hand side axis, and the number of edges as time series lines against the left-hand side axis. Note that the three data sources have different starting periods as the bank initiated collections over time, and some information, such as ownership, has data missing over certain periods. In Figure 4.1, the number of ownership links are shown in purple colour, amounting to approximately 100 thousand edges in each month and there are periods when the data is missing and the number goes to zero. The supply chain transfer relationship data starts much earlier from February 2018, shown as the yellow line in Figure 4.1. Lastly, the transaction information contains actual monetary transfers between the firms. These could be very similar to the transfers network but we do add that information.

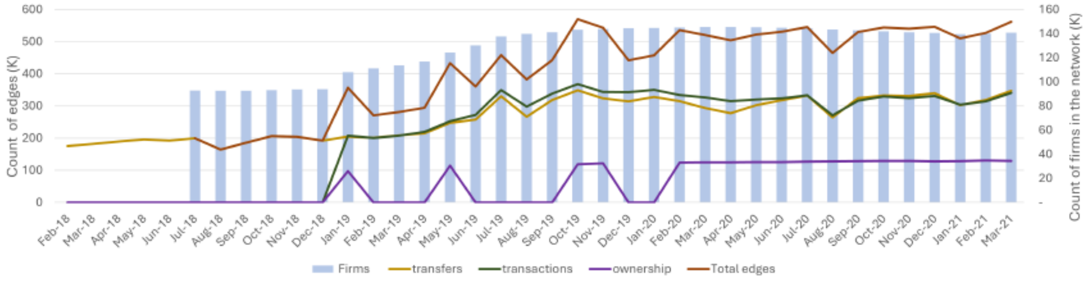


FIGURE 4.1: Network composition over time, showing the node count and the number of edges from different network data sources

As we prepare our data, we do not account for the direction of the relationship so we aggregate the three different relationship information types between two firms across all networks in the last 6 months, creating undirected edges. Each edge contains information about what kind of network connections applies and the frequency of links between a pair of firms. For a pair of firms that have common ownership, had transactions in the last 6 months and supply chain information as well, we will thus have a vector of attributes $[b_o, b_{tr}, b_{sc}, c_o, c_{tr}, c_{sc}]$, where $[b_o, b_{tr}, b_{sc}]$ are binary attributes indicating whether they are from ownership, transactions or supply-chain transfer networks, respectively. $[c_o, c_{tr}, c_{sc}]$ indicates the frequency of this relationship in the past 6 months. If they transacted three times and do not have another relationship, the edge attributes would be $[0, 1, 0, 0, 3, 0]$ for the pair of firms. The networks start from July 2018 for the input to the models, as we need 6 months of history. For example, for August 2019, we have 139,808 firms in the network with 382,694 edges.

In this study, the network at a given time t is, hence, represented as an undirected graph, $\mathcal{G}_t$. For each node, the neighbours of that node are crucial to calculate certain network-related features. Formally, we define the set of neighbours for node i, denoted by $\mathcal{N}_i$, as

$$\mathcal{N}_i = \{j|(i, j) \in \mathcal{E}_t\} \tag{4.1}$$

The other data categories are tabular in nature, and we combine them to create a snapshot for each firm in a given month. There are two ways to handle the data: aggregating to the firm level or treating each contract separately. We observed that each company could have multiple contracts, not all of which might be actively used. We tried the first way, which was to group all lines of a given company into one and create credit behaviour variables that tracked the aggregate utilisation (i.e., total balance over the sum of limits) over the last six months. This, however, made us lose important behavioural information as having multiple contracts would affect the utilisation metrics. Instead, we used a hybrid approach, in that we used the contract with maximum exposure or the highest drawn amount so as to capture all the patterns, while we used the firm-level features for the other static or financial information.

4.3.1.2 Firm characteristics

Table 4.1 summarises some of the observed data according to the size of the firms. There are 156,460 distinct firms in this sample. On average, the largest (medium-sized) firms, categorised as P3, have 10.32 times the exposure (the monetary amount that is drawn) of the smaller firms, which means their default count might be small, but when they do default, there is a larger amount at risk per firm. The overall default rate is 4.95%, which is largely due to the category of smallest firms (P1). As we see, the average number of contracts per firm is much higher for medium-sized (P3) firms than for micro to small (P1) firms.

TABLE 4.1: Population of firms by size over the entire time period with default rate and drawn amount relative to category P1

Firm Type	Total firms	Defaults	Default rate	Relative exposure	Avg no of contracts
Medium (P3)	18,749	486	2.59%	10.32x	5.5
Small (P2)	36,421	1,475	4.05%	3.31x	3.15
Micro to Small (P1)	101,290	5,776	5.70%	1.00x	1.32
Total	**156,460**	**7,737**	**4.95%**		

These networks are also heterogeneous: some of the focal firms whose default risk we seek to predict do not have any network information, whilst a large subset of firms do not have a credit line with the bank but are nonetheless part of the networks. We illustrate this with an example in Figure 4.2. This also shows the dynamic nature of the problem with new firms joining the network.

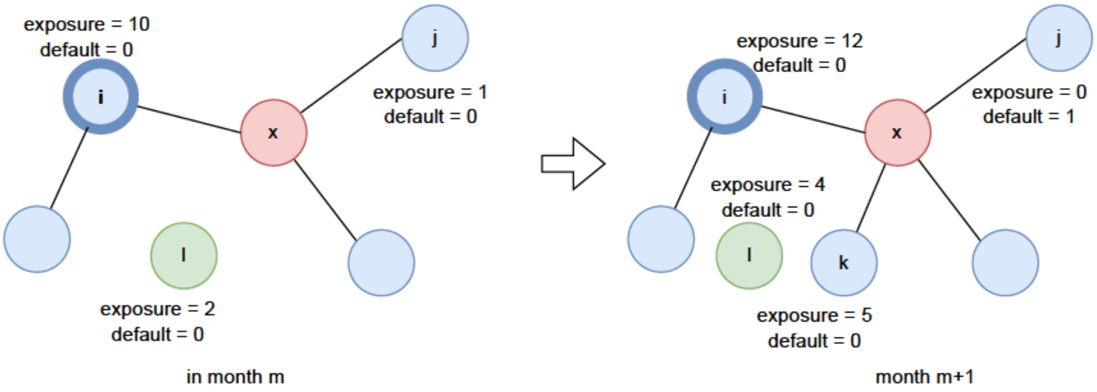


FIGURE 4.2: An example subnetwork depicting the heterogeneity of firms using coloured nodes and the edges representing a relationship from any network source being observed in the past six months; the figure also illustrates the dynamic nature of the network by showing how it might change from one month to the next

In this figure, the blue and green coloured nodes are the ones we are interested in predicting default for; the red-coloured node is part of the network of firms but does not have a revolving credit facility, so there is no exposure information for it. In the next period, a new firm k joins the network. We show one feature, exposure for each firm, and the default is the target variable, which is the status of the firm one year ahead. If we were to remove firm X, then the network would lose information. For firm i, the status of firm j could be important. Also, we show firm l, which does not have network information, meaning the models have to fall back on behavioural data for that firm. The illustration assumes these have not had any transactions or supply-chain transfers in the past 6 months, and we could not find any ownership data for the firm.

Active firms have at least one active credit line contract, which are the blue and green coloured nodes in the example. We plot the number of active firms in each month in Figure 4.3. Compared to the firm count in Figure 4.1, this number is much lower, since a sizable proportion of firms in the network do not have an active credit line contract. Inversely, some firms have an active credit contract but are not part of the network (shown as 'out of network' in Figure 4.3).

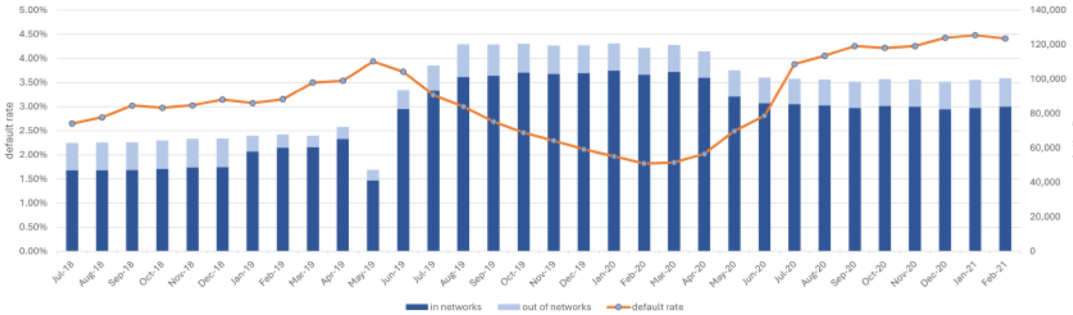


FIGURE 4.3: Active firms and the default rate in a given month aggregated to all firm sizes

4.3.1.3 Population changes over time and default characteristics

Figure 4.3 shows similar information to that in Table 4.1, but now separated by month irrespective of the size of the firm. The default rate is shown as an orange time series against the left-hand side axis, and the number of active firms in a given month is shown as a bar chart against the right-hand side axis. Here, the default rate tells us of the active firms in a given month and how many defaulted in the next year. This differs from the earlier Table 4.1 where we aggregated over the entire time period and differentiated only by firm size. The default rate shows a cyclical trend, with default rates initially rising until May 2019, from 2.73% to 4.2%. They then kept falling until the start of the pandemic, around March 2020, to a low of 2.08%. The rate doubled in the next few months and has been at these elevated levels until the end of the data period, suggesting a clear impact of the pandemic shock.

Every month, we use the network described above, and the tabular behavioural data available at that time, as inputs to our models. After producing a probability of default for the subsequent year, we roll one month forward for the next prediction.

4.3.1.4 Behavioural features

The data already includes some behavioural variables in the contracts listing, the most important of which are listed in Appendix B.1. In selecting which additional behavioural features to create from the available data, we follow Tong et al. (2016), who proposed some similar features to predict EAD. Table 4.2 shows this list of features that we created. For those features that roll up behavioural data over time, we use the same lookback period of 6 months as we did with the networks. We define the exposure as the amount drawn in the relevant contract and utilisation as the ratio of the exposure to the credit limit of the revolving credit facility. In addition to these, the contract data has ready-made behavioural features. We describe those essential features in Appendix B.1.

TABLE 4.2: Customised behavioural variables used in the models

Variable	Description
utilisation	Ratio of exposure to limit on the contract with maximum exposure
max_delinquency	The present worst delinquent state across all contracts
contract_count	Number of contracts the firm has active
total_exposure	Overall exposure across all contracts
total_limit	Overall limit across all contracts
min_utilisation	The lowest utilisation across all contracts in the past 6 months
max_delinquency_time	The worst delinquent state across all contracts in the past 6 months

All the behavioural features, including customised and ready-made features for a firm i, are from here on represented as a vector, $X_{i,beh}$.

4.3.1.5 Derived network features

In addition to behavioural features, we also include a few network features, which are denoted by $X_{i,nw}$, for firm i. We use the well-known measures of degree centrality, betweenness centrality, and personalised PageRank for each node, which can be calculated from the network topology. The degree and betweenness centrality quantify how important that node is in the network, either counting the neighbours or shortest path that pass through it (a measure of how much control the firm has). We selected degree and betweenness centrality based on earlier studies (Naudé et al., 2014; Poenaru-Olaru et al., 2022). The PageRank metric, unlike the other metrics, differentiates the neighbours or the connections based on the importance of nodes (Brin and Page, 1998).

The degree centrality dc_i for firm i is defined as

$$dc_i = |\mathcal{N}_i| / (|\mathcal{V}_t| - 1) \tag{4.2}$$

or the total number of neighbours divided by the total number of nodes minus one. $\mathcal{N}_i$ again denotes the set of neighbours for node i.

The betweenness centrality measure specifies the activity that passes through a node in the network. To measure this activity level, we first compute the shortest paths between all pairs of nodes. The betweenness centrality of a node is the fraction of the shortest paths that pass through the node relative to all shortest paths in the network. For a firm i at time t, this is defined as

$$bc_{i,t} = \sum_{j,k \in \mathcal{V}_t} \frac{s(j,k|i)}{s(j,k)} \tag{4.3}$$

where $s(j,k)$ is the number of shortest paths between (j,k) and $s(j,k|i)$ is the number of shortest paths that pass through i. $s(j,k|i) = 0$ if $i = u$ or $i = v$, and $s(j,k) = 1$ if $j = k$.

Finally, the personalised PageRank allows us to discriminate between the nodes. We use personalised instead of traditional PageRank as, in our network, there are a number of nodes that do not represent firms requiring a default prediction, which means their importance is lower compared to the firms we are interested in. This measure has been extended for credit risk prediction using multi-layer networks by Óskarsdóttir and Bravo (2021). For all nodes in a network, the PageRank is calculated recursively, with a maximum of p iterations. The personalised PageRank vector is the probability, for each node, that a random walker lands on it. A higher value implies higher importance in the network.

$$\vec{PP}_p = \alpha \cdot \vec{PP}_{p-1} + (1-\alpha) \cdot \vec{h} \tag{4.4}$$

$\vec{h}$ are the nodes of interest which have default and contract information, and α is a parameter for the algorithm, which represents the probability that, at each iteration, the walk is over the neighbours of the current node and, with a probability $1-\alpha$, we start the walk again from any of the nodes of interest. This α is also called the damping factor and is set to 0.85 as per most studies.

In addition to these traditional metrics of centrality, we also introduce a few network features based on the utilisation and current delinquencies of a firm's local neighbourhood. For example, for each firm, we add the average utilisation observed over the contracts of all neighbours. We know from earlier studies that high credit utilisation by a given firm is associated with increased default risk for that firm (Tong et al., 2016). If utilisation by any firm linked to that firm is also found to be important, this would be a novel insight gained from this study, giving us an early warning signal of how risk might propagate over a network of firms.

All aforementioned network features can be easily incorporated into any traditional model, as they can be merged with any other tabular data.

4.3.2 Models

The problem at hand is a case of binary classification, in which we look to predict the risk of a default event one year ahead, using behavioural and network features. The first subsection below covers the GAT model. Next, we discuss the multimodal architecture in detail, the reasons for choosing it over other possible multimodal architectures, and how we trained and fine-tuned all models. We also discuss the benchmark models and the training settings that are specific to all these models.

4.3.2.1 Graph Attention Networks

Here, we formally define the GAT specific to our problem. Given a graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ as defined in section 4.3.1.5, and the input features defined earlier, the GAT transforms the input features X_i, of dimension T, for firm i, into a higher-order representation H_i of dimension T' given by

$$H_i = F(X_i, \Sigma_{j \in \mathcal{N}_i} a(j,k)(WX_j)) \tag{4.5}$$

Here, $\mathcal{N}_i$, defined in equation (4.1), are the neighbours of firm i. $W \in \mathbb{R}^{(T' \times T)}$ is a weight parameter matrix of the model that is learnt during training. $a(i,j)$ is the weighted importance score for the pair of firms j and i. This is the self-attention mechanism, which consists of a feed-forward layer, non-linear ReLU function and finally, Softmax layer. A feed-forward network over each neighbour firm j with the present firm i first generates the embeddings. The ReLU is a non-linear activation applied on these embeddings, and finally, the Softmax layer generates the attention weight for each pair of firms. The function $a(i,j)$ is where each type of GNN differs; whereas it was a convolutional function for GCNs, it is attention for GATs,

$$a(i,j) = softmax_j \left(\sigma(\mathbf{a} \left[Wx_i || Wx_j \right]) \right) \tag{4.6}$$

where σ is a non-linear function, LeakyReLU in this case, $\mathbf{a} \in \mathbb{R}^{2T'}$ and $||$ is the concatenation operator.

$F(\cdot)$ applies the non-linearity function (usually a Regularized Linear Unit, ReLU) after aggregating all weighted outputs. GAT also allows for multi-head attention, where each head learns a different input aspect. For a K-multi-head attention, we concatenate each head's outputs to construct the final representation. In this study, based on where the GAT is in the multimodal architecture, we used $K = 2, 4$. The GAT model is a significant component of the multimodal model used for this study.

4.3.2.2 Multimodal architecture

Figure 4.4 depicts the full multimodal architecture used in our study. The input is a graph where all the nodes contain their features, and these same node features are the inputs to a feed forward linear block (shown on the left). These features consist of the behavioural features and network features discussed above. For all models to receive the same input, we use only the firms that are part of the network here for comparison, hence leaving out firms that have a revolving facility but did not generate any network information in the past 6 months. For firm i in the graph, its node attributes are $X_{i,beh} \parallel X_{i,nw}$ (where $\parallel$ is again the concatenation operator). We see the neighbours of this node are j, k, z, each with their own attributes. The active firms are denoted by a boolean flag, $mask = 1$ (we see the features are from the data), but for the firm z whose $mask = 0$, these features are just filled as ones, as this is a firm that is part of the network but does not have any revolving contract. The firm z will, hence, not be part of the training or test firms set. These features are the inputs to the GAT model (labelled as GATConv in the figure), which we discussed in Section 4.3.2.1. The graph embeddings at these layers are H_i for a firm i. Note that the H_i can also replace derived network features when used in a traditional model. We test these embeddings with the benchmark random forest model and report them in the results section (section 4.4).

A non-linear transformation is applied to the GAT model output with the ReLU layer. We pass these outputs two more times, as each successive block of layers can learn higher-order representations, commonly done in deep learning models. The blocks closest to the output should have different embeddings that are much more suitable to what we want to predict. In the figure, we write $2x$ to denote that the previous section of blocks is thus repeated two times. For the structured data, the embeddings pass through a pair of Linear and ReLU blocks, after which they are all combined with the Concat block.

A second part in the architecture (shown in the left section of Figure 4.4) simply passes the firm-level tabular data, $X_{i,beh} \parallel X_{i,nw}$, to a series of deep learning layers. The first linear block is represented as

$$M_i = F(W \cdot (X_{i,beh}||X_{i,nw}) + b) \tag{4.7}$$

where $F(\cdot)$ represent the non-linear ReLU activation function, W is the learnt parameter matrix and b is the bias term. The GAT blocks generate embeddings from the firm's features and also use the neighbours' features. This linear block generates embeddings of a firm's own input features.

Further down, the final output from this tabular data component M_i (left) is concatenated with the network embedding H_i, produced by the graph learning

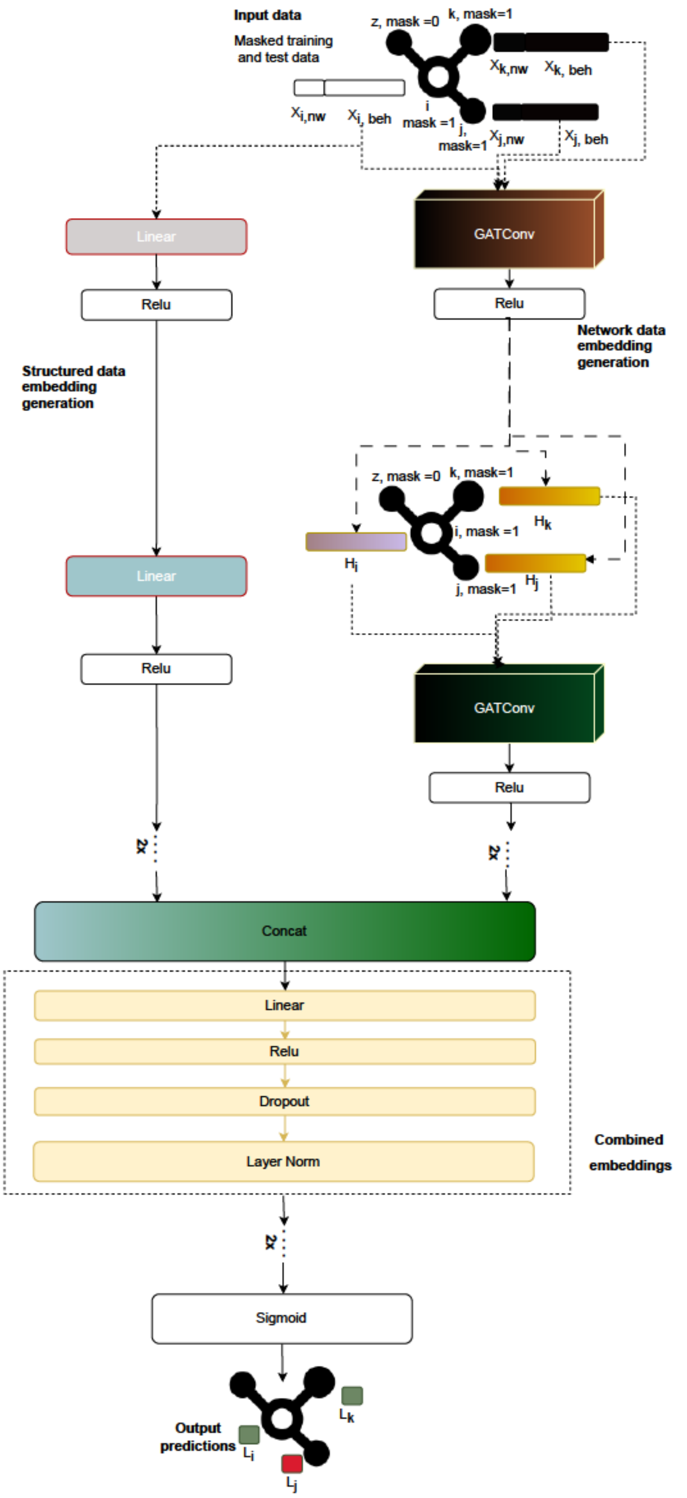


FIGURE 4.4: Multimodal architecture overview with behavioural structured data and network embeddings

component (right). To learn any relationships between these two representations and default, we train the model further with another set of linear and ReLU blocks which generates the combined embeddings of tabular and network embeddings. We deploy dropout (Srivastava et al., 2014) to avoid overfitting and layer normalisation (Ba et al., 2016), which normalises the parameters and reduces training time. Lastly, we use the sigmoid layer to derive the final probability of default, as this is a binary classification problem. The sigmoid function provides a suitable output between 0 and 1.

This architecture allows us to quantify model performance differences on various data specifications and models. For example, it helps us answer whether the tabular data or network is more important, and also, among the different model components, whether the GAT adds value or whether using the linear blocks, which are feed-forward networks, is sufficient. This can be achieved by masking, with the help of a binary flag indicating whether to include some element. For example, when discussing the network heterogeneity earlier on, we have already seen that, at a firm level, each firm had a binary mask (which can also be seen in Figure 4.4) indicating whether it is included in the training. Likewise, each block in the architecture can be configured similarly. We can thus test this full multi-modal architecture under different combinations. Appendix B.2 presents more details on these extensions. The various configurations are used to report the results.

4.3.2.3 Benchmark models

We use two benchmark models to compare our own model: logistic regression and random forests.

Logistic regression is the most widely used model for credit scoring or default prediction. In this paper, the model is used in a similar context, to predict a one-year probability of default for each firm in any given month. The logistic regression gives the probability of default as

$$\frac{e^{\alpha+\beta * X_{i,beh}}}{1+e^{\alpha+\beta * X_{i,beh}}}$$

where α and β are the parameters estimated using the training data. The logistic model is retrained every month to facilitate a like-for-like comparison with our proposed model.

Random Forests are ensemble models that use a collection of decision trees to arrive at an overall model output. Each decision tree is trained on a different sample of the data, choosing splits from a random subset of the variables (Breiman, 2001). They have been used as benchmark models in fraud detection (Alarfaj et al., 2022) and credit scoring (Lessmann et al., 2015; Ala'raj et al., 2021).

4.3.3 Experimental settings

Part of the experimental setup involves comparing the predictive performance of different groups of variables. Initially, we start with our benchmark models and add the different groups of variables. We group them by the type of information, which we discussed in Section 4.3.1. From this, we use the two most informative groups, and we use feature selection to use the most informative set of features. Using this set of features, we test with our proposed model and architecture and again with the benchmark models.

4.3.3.1 Data preprocessing and training split

We replace all the missing data with their median values in the original data and, each time, create another binary variable indicating whether that value was missing or not. These newly created missing variable flags are added to the data. Missing data could indicate some behaviour that is not yet expressed but is not necessarily random. The study by Korangi et al. (2022) on mid-caps showed that missing data are a good indicator of future deterioration of the health of a firm.

As explained earlier, during the training and evaluation phase, we represent the heterogeneity in the firms by using a binary mask flag. If the firm is an active firm, i.e., one which has an active credit line, it is included in the training and testing set of firms and masked as 1. The other firms in the network are not used to measure training or test performance. For training purposes, we split the data in different ways, initially out-of-time, whereby the last 30% of the data following December 2020 was used as a test set, and the data from June 2018 to December 2020 constituted the training set. We also used out-of-universe validation where we removed the firms that are part of the test set from the training data such that the firms in the test set do not feature at all in the training set. We use these splits for our initial results, to understand the data sources and to prepare the features for the GAT model and the other benchmark models.

The deep learning models had a different split, as we can only predict for the firms in the network. This same split is used for the benchmark models such that all models process same data. The networks also vary with time, so we ran a similar out-of-universe validation but within the same time. So for every graph, we used one set of firms as testing firms and the remaining as training firms. The GAT model trains on the same graph but the test firms are not visible. We report the AUC for these models using this in-time but out-of-universe testing procedure.

4.3.3.2 Model assessment and loss function

For performance measurement, we used the Area Under the Receiver Operating Characteristic Curve (AUC) measure. As we have a classification problem showing pronounced class imbalance in every period (i.e. only a small set of firms default), using an accuracy metric is unsuitable. A higher AUC value for the model suggests a better ability to discriminate between defaults and non-defaults. We look for a value greater than 0.5 as the latter value would signify random predictions. The higher the AUC, the better the model performance; perfect classification would give a value of 1. This metric has been widely used in default prediction literature (Calabrese et al., 2016; Óskarsdóttir and Bravo, 2021; Korangi et al., 2022)

Training the deep learning models also require a loss function to minimise for the gradient descent. We use the binary cross entropy function,

$$entropy(y_i, L_i) = -(y_i * \log(L_i) + (1 - y_i) * \log(1 - L_i)) \tag{4.8}$$

where L_i is the probability of default produced by the model, and y_i denotes the true default outcome (0 or 1) for a firm i.

4.3.3.3 Parameter selection and model tuning

Each model in our study has a set of hyperparameters that must be set or fine-tuned, and every run could be different. In this section, we describe the hyperparameters for each model and the range of values we tried.

The deep learning models have a large number of different parameters, as we used four blocks of GAT and ReLU for the networks. The first and the final GAT block had multiple head attention with an output dimensionality of $[32, 128, 64, 64]$ for each block, respectively. The Linear block for the neural network and the multimodal architecture were the same with dimensionality of $[4096, 512, 80, 80]$. We used the default dropout rate of 0.5 for all models. We arrived at these from a hyperparameter search across these outputs and found these to be the best fit for the first graph.

For the logistic regression, we use L1 regularisation to shrink some of the coefficients to zero and perform variable selection. In the random forest, we set a maximum depth of 10 and use the default settings elsewhere.

In the next section, we report the results and raise some points in the discussion.

4.4 Results and discussion

4.4.1 Performance across benchmark models

As we handled large data sets, we tested out different combinations of the data to prepare for behavioural credit default prediction. We needed to understand which data was giving the highest predictive power and focus on that data to create behavioural variables. We looked at the four different kinds of information and gradually increased the data input to our model. For this purpose, we used only logistic regression (LR) and random forests (RF), since they are easier to train and produce benchmark models. The results are reported in Table 4.3. We started with static information such as client type and number of employees, which was not informative of default at all, as both models had AUC close to 0.5. Once we add contract information, the AUC increases considerably, especially for the Random Forests (RF in the table). Next, we added financial data to the static and contract information, which did not improve the performance by much, but the (anonymised) location information did improve the model, albeit only slightly. Based on this finding, we decided to utilise the contract information and location information for the rest of the study.

TABLE 4.3: All data groups, benchmark models performance, to select the most relevant features

Out-of-time AUC	**Model**	
Input	**Logistic**	**RF**
Static information	0.508	0.506
+ Contract information	0.660	**0.761**
+ Financials	0.666	0.764
+ Static location information	0.671	0.777
Contract information + Network features	**0.601**	**0.833**
Out-of-universe AUC	**Model**	
Input	**Logistic**	**RF**
Contract information + Network features	0.598	0.830

We added network features to this data source and performed the two validation tests outlined in Section 4.3.3. The results are shown in the lower part of Table 4.3. In both test setups, we employed only contract information and network features. The validation shows that the logistic regression model continues to perform fairly poorly. In contrast, the random forest model performed even more after augmenting with the network features listed in section 4.3.1.5.

A key finding from the table is that the Random Forest model performs substantially better than logistic regression, for example, giving an AUC of 0.830 for the network-augmented out-of-universe and out-of-time test set (see bottom row). The AUC on the out-of-time test set (0.833) is rather similar, giving us some confidence that there is no overfitting. Encouragingly, these are much better than the out-of-time AUC results without the network information, shown in the higher half of the table. This demonstrates that there is already a lift in predictive power by simply adding network features to the firm's own features.

For the next analysis, we dropped the logistic regression, as its results were not favourable, and instead focused on the random forest as the benchmark model, adding a neural network model as that is more relevant to the next analysis.

4.4.1.1 Multimodal performance

In this section, we add the deep learning models to our analysis. The results are shown in Table 4.4 for the test set and average over all periods. For the contract information we identified earlier as our preferred behavioural data source, we performed variable selection to include the most relevant behavioural variables, calculating them as we discussed previously. The left column shows the input data sources for each model on the right. Where the results are not applicable we leave them blank.

TABLE 4.4: AUC performance of each model over different sets of inputs

Metric: AUC	**Model**		
Input data	**RF**	**NN**	**GAT**
Behavioural data +			
Network features	**0.8780**	0.7350	0.7822
GAT model embeddings	**0.8964**	-	-
Network data			
Network features	0.7390	0.6825	**0.8241**
GAT model embeddings	0.7947	-	-

The first two result rows shows the AUC when behavioural data are combined with either the derived network features from section 4.3.1.5 or deep learning model generated embeddings. We see that, with an AUC score of 0.878, the RF model outperforms the other two models when we just take the behavioural data and add the calculated network features. However, the next row shows that the model can be improved further by using the deep learning embeddings instead of these simpler network features. This percentage point improvement of 1.8%, however, needs to be evaluated against the added cost and complexity of deploying these deep learning models. As we compare against some of our earlier results that omitted network data, we can see that adding network data again improves the models significantly.

To further test, we then trained the models using only the network data as input, the results of which are shown in the second half of Table 4.4. Here, we see that the GAT models are able to extract some higher relationships and hence have better predictive power than the other two models, giving an AUC of 0.8241. Moreover, the embeddings generated by GAT are powerful, as the NN model, which is similar to the GAT but without the network structure as an input, performs worse with an AUC of 0.6825. This shows that the GAT embeddings are better value compared to the neural network model embeddings. Note that we discussed how to extract these embeddings in Section 4.3.2.1. We did not test with the neural network embeddings and used the GAT model embeddings for RF input. They improved the random forest performance from 0.878 to 0.8964 higher up the table and from 0.7390 to 0.7947 if we do not use behavioural data.

These results shed light on the relative importance of behavioural variables and the added network data in improving the models. The deep learning model embeddings add value to the performance when they are combined with the random forest model, whereas the GAT model outperforms without the behavioural data, but underperforms when it is present. We believe this may be due to the nature of the data, as a more complex model such as the GAT might not be required when there is already a strong direct relationship between our target feature and some of the independent variables, which is the case with the behavioural data. This confirms earlier studies that suggested that, on tabular data, tree ensemble models such as random forests tend to perform better than deep learning models, although this area is still evolving.

4.4.2 Evolution over time

Over all the data, we have seen the behavioural data, which are constructed features, have far more predictive power. The network can be represented in the form of traditional network variables, and that further improved the model's performance. Here, for the best model, that is, the random forest, we conduct an out-of-universe validation. We sample 30% of firms into a test set and remove them from the training set. The model is trained every month with only the training set of firms, and we test the model's performance on the active firms in the test set. In practice, we expect the model to be retrained and to work with firms that have not been part of the model. The novel set of firms can be challenging. This is why we chose the random forest model for this validation. Figure 4.5 shows the AUC score over time, with the solid line representing the AUC score when all the data is given as input, and the two dashed lines are the AUC scores when either only the behavioural or the network data are inputs to the model.

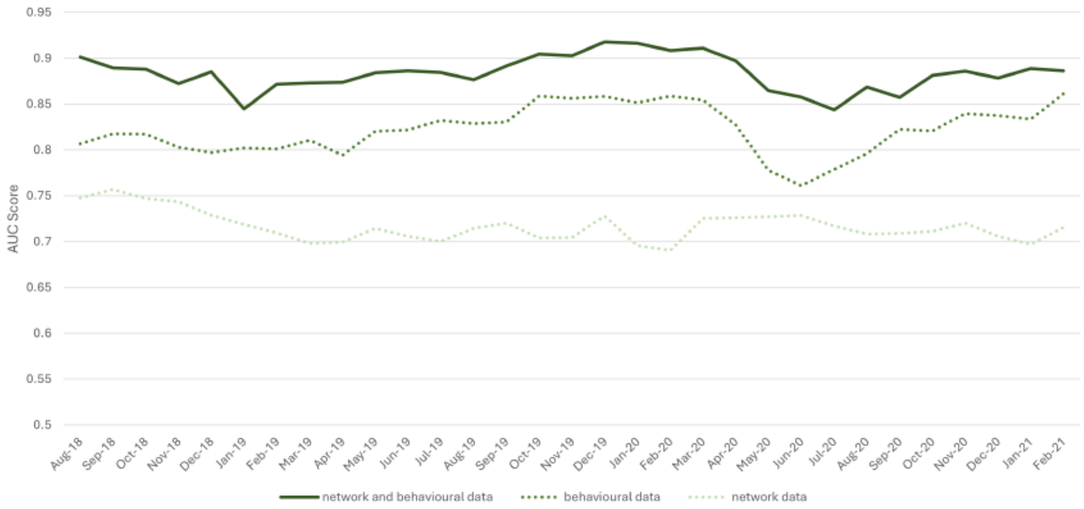


FIGURE 4.5: Model performance over time

We see that behavioural data does well compared to the network data. Interestingly, the curve also resembles the default rate in Figure 4.3. We clearly see that when there was an external shock to the system in March 2020 (COVID-19 pandemic), the behavioural model performance deteriorated, roughly with a two-month lag. As the economy was in shock, there were a lot of irregular payments that caused the model to deteriorate. The AUC picked up again as the economy was normalised (reopening after lockdowns) to the new behaviour. The network performance, on the other hand, was more resilient, showing that valuable performance could be extracted from it, especially when the portfolio experiences an external shock. In other words, adding network features to the model can bring not just improved but also more stable performance. Taken on its own, in the early part of the available timespan, the network data started quite well. Still, performance deteriorated, which we believe is due to the availability of the data in the network and mixed signals generated with the churn in the data. As time evolved, the network data performance stabilised.

The overall model's AUC is also quite high, which will be linked to us testing the model during the same period as the training data now. This shows, though, that the model is resilient to being presented with new firms and can predict their default risk even when it has not encountered the firm in the training data. The model has thus learned some generalisable features from the training set that is being used.

Overall, these out-of-universe results show that the behavioural features are again the most important, but network features also add a lot of value in further improving the model and can be relied upon in macroeconomic shocks.

4.4.3 Explainability

Finally, we conducted SHAP analysis to explain the model results better and quantify the importance of the groups of variables (Nandlall and Millard, 2019). We did this test on the best-performing random forest model with all the features available. This analysis has been extensively used in the deep learning literature to identify the most important input features. Figure 4.6 shows a beeswarm plot of the SHAP analysis. We also map the variable names to their description in Appendix B.2. This should be read for each variable. The SHAP value is plotted against the x-axis, with the impact on the model estimates of high (low) values of the feature shown in red (blue), respectively. For example, high values of *irregular_count* thus have a strong positive impact on the model estimates for default risk (i.e. they are deemed to increase the risk).

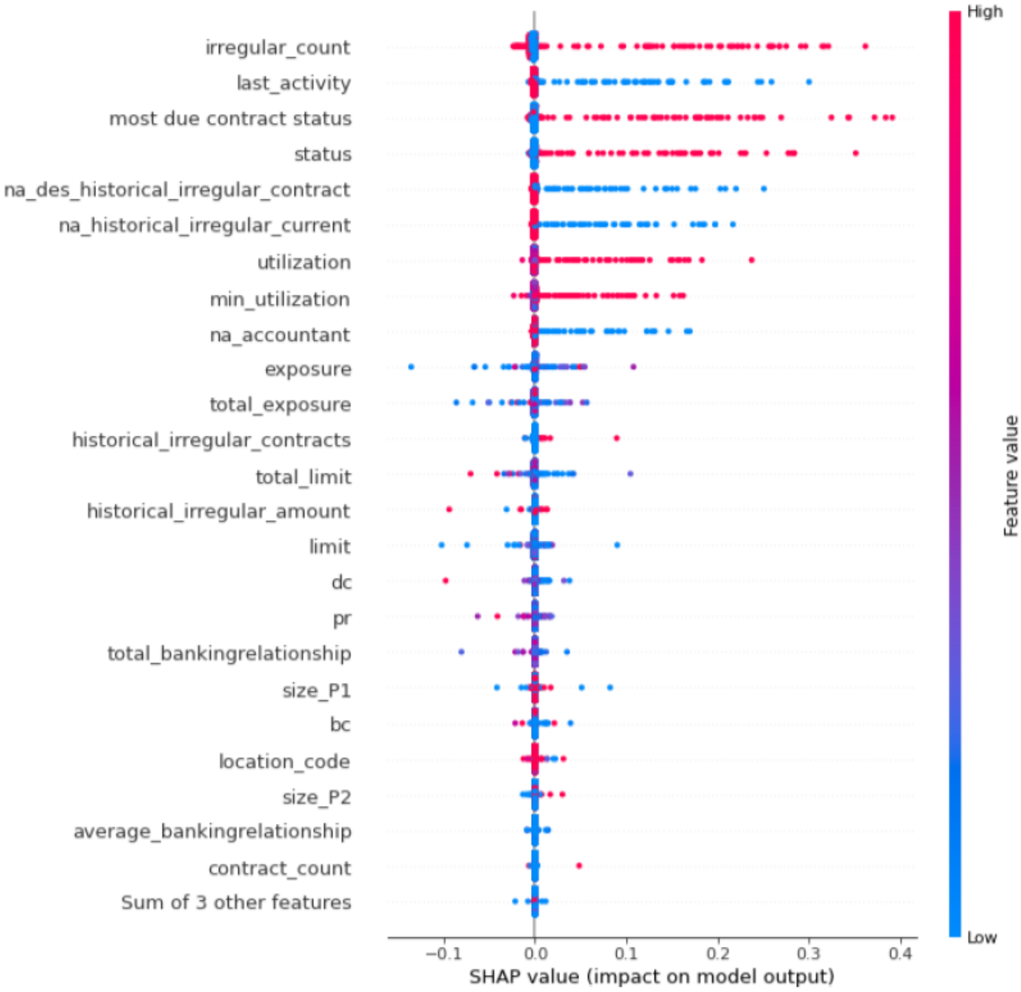


FIGURE 4.6: Beeswarm plot of the SHAP value explanations

The SHAP analysis shows that the count of contracts exhibiting some irregularity has the highest relative importance, and from the variables we created, the utilisation ones appear to be the most important features. This confirms results mentioned in earlier studies where more risky firms could draw down on their limits and thus increase the exposure and utilisation (Agarwal et al., 2006; Bellotti and Crook, 2013; Wattanawongwan et al., 2023). The number of days some payments are due is also important. Here, we note that we categorised default as any contract that is over 90 days due, thereby using the definition that is widely used by banks. We believe the models could be further tested when these signs of irregularity are unavailable. Lastly, the network features also come into play, with degree centrality, PageRank, and the average status of a firm's neighbours' contracts being important variables.

4.5 Conclusions and further research

In this study, we worked with a unique credit lines dataset, to predict SMEs' default probability from behavioural data relating to revolving lines of credit, and explicit networks comprised in the data. We initially used a deep learning architecture, along with random forests and logistic regression as benchmark models. We found the behavioural data to have good predictive power in itself, and adding the network-related variables improved the model performance even further.

The deep learning models did not perform well with behavioural data as there are easier relationships in behavioural variables that more traditional models can exploit. However, with no behavioural data, just using the networks as the inputs to these models performs fairly well. Here, the GAT model performs particularly well compared to a similar deep neural network model, which shows some network structure in the data being exploited by the GAT, which other models cannot. They generate embeddings which, when fed together with behavioural data, produce better results for the traditional models as well. However, we feel more research is needed here, in light of the mere 1% improvement we observed by using graph embeddings as opposed to network features that were calculated in the traditional manner. The costs of maintaining these models and developing them over time initially might look too high compared to the improvement, but further analysis suggested these models may be more suitable to changing data distributions or economic environments, so the embeddings could be more robust.

We conclude that the behavioural data, in particular, does not need higher-order relationships to predict default better, as these variables themselves track the firm's behaviour closely. Some changes in these behaviours are predictive of default. After conducting an explanatory analysis of the features, we find higher utilisation and the number of times the account was irregular in the past to be some of the most important features, confirming earlier studies. We believe the deep learning models will be better equipped to detect the very first signs of irregularity. We would need to define our research problem again, which could be subject for further research.

With an out-of-universe sample validation over the time period for which we have the data, we found the models to track the observed default rates as long as they are stable. When there was an extreme shock in the environment in the form of a pandemic causing the behaviour of the portfolio to be impacted, we found the behavioural data lost some of its importance while the network data performance was largely stable. As the markets returned to normality, albeit with higher default rates, the behavioural models picked up again and improved their performance. We believe network models could, hence, be deployed in stress testing the portfolio to different economic conditions. Over time, as network data collection improved, we found the models initially had lower performance as newer relations were established, but once they were established, they were quite robust over time.

Future work could look at using deep learning models directly to generate behavioural features without having to manually calculate them, which could generate novel features for behavioural credit scoring. For the networks, an interesting area would be to use the embeddings to model contagion and create systemic early warning signals for portfolio deterioration. We have not included much external data about the firms, and this could potentially be another area to explore to improve the model's performance further.

We conclude that behavioural variables are significant predictors of the probability of default, and traditional models can be suitable if one has access to this kind of close monitoring data. When such data is unavailable, we have seen deep learning models perform quite well, extracting higher-order relationships and embeddings that are more powerful than calculated network features.

Acknowledgements

This work was supported by the Economic and Social Research Council [grant number ES/P000673/1]. The last author acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [Discovery Grant RGPIN-2020-07114]. This research was undertaken, in part, thanks to funding from the Canada Research Chairs program.

The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

Chapter 5

Conclusions

5.1 General conclusions and summary of the thesis outcomes

This thesis proposed novel approaches to address credit risk modelling problems in high-dimensional and sparse data settings. These approaches aimed to alleviate the high cost of credit for SMEs or to better diversify into illiquid firms such as mid-caps. An interdisciplinary strategy was adopted to conduct the research, which involved bringing in new techniques from other areas to improve credit risk models. In doing so, I have made methodological contributions in risk management, adaptation of computational methods, and using novel data sets. In risk management, I studied the term structure of default prediction instead of single point, large-scale portfolio optimisation, and using behavioural features for default prediction in revolving credit facilities. In the methodological contributions using computational methods, I was the first to adapt the now widely popular transformer encoder designed for textual data for panel data methods, used different loss functions to suit the modelling objectives, developed suitable training methods, quantified the importance of various input data sources, developed custom deep learning layers to make them ideal for portfolio optimisation and developed multimodal architectures to use various kinds of data as inputs effectively. Finally, I used novel data sets focusing on mid-caps and SMEs throughout this thesis. I concentrated on mid-cap firms for initial studies as they are less well-studied. I hypothesised that deep learning models are more suitable when the data is sparse and wide (several features with lots of missing data) or there is illiquid pricing of market data (no continuous trading data). I focused on long time scales of 30 years to test the robustness of these models. With the SME dataset, this is one of the few studies that has worked on such a large-scale dataset from a major financial institution. I studied revolving credit facilities, as the literature is sparse compared to loan data, and worked with behavioural features. Network data also featured prominently as another data source that I used, by either creating networks from tabular data using existing filtering techniques for the mid-cap data, or using existing network data inherent in some of the transactions and ownership information from the SME dataset. In general, in all of these areas, the results are promising. In this section, I go into further detail about the results and identify limitations and the future scope of such studies.

In this study, we tried to replicate the business environment while being aware of the challenges this brings out. On the data, while data selection choices can make them suitable for academic research, they can result in survivorship bias, which makes it challenging to deploy them. Our data is population-based studies, where I consider all the data. This helps to bridge the gap between academic research and its application in industry. This ambition is also apparent in the final chapter, where I worked with a large financial institution and sought to provide insights from their data.

The limitations to applying them in the real world pose a different challenge now. The areas of enquiry studied in the thesis are promising to explore further but simultaneously prove complex to deploy. Business needs dictate timely, fast and interpretable analysis. At the same time, complicated, robust modelling with improved accuracy and performance could be beneficial over a longer term and might need significant investments. I identified several specific modelling improvements such as a differential training approach, initialisation of graphs through a graph filtering approach, and use of deep learning embeddings as inputs to traditional models. From these specific contributions and improvements, in this section, I also expand this work's results into a much broader scope for our work and show the complex needs of future research. As more alternative datasets emerge, the complexity of handling them requires institutions to invest in advanced analytics and deep learning capabilities and adapt these to their needs. I do not believe all alternative datasets would be valuable for improving the model's predictive power. However, with increasingly available alternative data sets of all types, it is essential to develop methods to evaluate such datasets without much manual time and effort. To even come to that decision quickly, we need more computational methods, such as deep learning methods, as they work on various datasets that require minimal data processing and less domain expertise.

The practical application of these methods to real-world data brings clear computational challenges. None of the research presented in this thesis would have been possible without using state-of-the-art GPUs and multiple high-memory computing nodes. However, once these challenges are tackled, the questions that can be asked become much broader, and the corresponding analysis can be applied to large-scale data settings. An interesting observation made over the course of my research is that, whereas at the outset, there were hardly any computing libraries specific to deep learning models, by the end of my research, we are now inundated with a wide variety of available libraries. This shows the rapid evolution and adaptation of deep learning and its interest worldwide. As these advances are likely to continue, I expect the challenges of adapting them to credit risk research questions will remain an interesting research focus over the coming years, especially as this brings about meaningful performance uplifts.

Specifically, I investigated three problems related to small or medium-sized businesses. I started by improving the firm-level credit default predictions for publicly listed mid-cap firms (often illiquid to trade). Next, I approached this same set of firms collectively and optimised the risk-adjusted returns of a mid-cap investment portfolio. Third and finally, I worked on an extensive private data set where I leveraged behavioural and network data to improve the credit default prediction of SME firms. A common thread among these three studies is that I worked on large datasets and employed similar computational techniques and deep learning methods. Each application required a degree of adaptation from the standard procedures in terms of the training and grouping of the data sets, the collective network and behavioural feature generation, and the design of specific architectures required to solve each problem.

For the first study on mid-cap firms, I focused on publicly listed mid-cap companies, as, similarly to unlisted companies, these often have limited liquidity. This differs from most deep learning studies in finance, which tend to gravitate to other large public data sets, such as those relating to large companies (for which there is continuous trading data), or on market microstructures such as limit order books or high-frequency trading. I correctly hypothesised that deep learning models could also be effective when the data is sparse, but a large breadth of data is available. Using different data of varying frequency of availability obtained from accounting information, market prices and general market economic data, I developed a transformer-based model. I needed to adapt it, as it was initially designed for natural language processing, and it was better to equip it to a panel-data setting like ours. The transformer-based model was powerful, offering better default prediction than a series of benchmark models, provided that a differential training regime was used. Specifically, I learnt that iteratively freezing previously trained models whilst adding further models in our multichannel architecture allowed it to use all input data effectively. Also, instead of producing a single one-year probability of default estimate, I modelled the default probability term structure over a short to medium-term horizon. This required introducing a different loss metric more suitable to the problem. This resulted in capturing some intuitive time dependencies in the forecasts. I also asked which periods and data sources are more critical. For this, I employed a Shapley approach to derive the importance of groups of features.

The second part of the research continued on the subject of mid-cap firms but looked at them collectively as a portfolio of investable assets rather than on a firm-by-firm basis. Firstly, I had not yet considered the interactions between the firms, which could bring additional insights. Looking at much of the existing research, I also found that they focused on very different types of firms, e.g. constituents of highly liquid indices, and routinely eliminated firms that defaulted (which could result in survivorship bias). To address this, I created a network topology of the firms based on historical market pricing information. This information was often sparse as these firms traded infrequently, so I used a procedure that had been shown effective in other application areas for similar data types but not used until now in portfolio optimisation. Once I had created the networks, the next question was about extracting value from them, for which I deployed deep learning models for network data (deep graph neural networks). Finally, I examined whether these are robust enough and how they work in different market conditions. This study also had to adapt the deep learning models to function in problems where reinforcement learning is generally used. I created new deep learning layers that satisfy portfolio constraints and produce portfolio allocations. I had limits on the number of firms that could be invested using regularisation parameters. I also had to modify the loss function like in our earlier study, and now creating one suitable for this problem.

For the final research output of this thesis, I combined some of the knowledge and skills gained from the previous two studies in that I again combined different data sources and applied network learning. This time, though, I focused on credit lines for small businesses, thus aiming to add to the sparse literature on this type of credit product within the SME context. Also, rather than build an application scoring model, I set out to build a behavioural scoring model, which, in the credit scoring literature, has been an under-researched topic for all business lines. Although, like in consumer lending, a firm's prior credit utilisation should provide the lender with a rich data source for creating behavioural features, it appeared no other studies had previously reported work on this within the SME setting, possibly as such data can be challenging to source. The network data considered in this study arose from relationships between SMEs regarding ownership, financial transactions between firms, and supply chain relationships. As default risk might potentially propagate over the ensuing firm networks, I investigated how to extract value from this data. I examined the suitability of multimodal deep learning models combining tabular and network data. I looked at additional insights deep learning models can generate, especially from the network data. One of the critical aspects here was the presence of a macroeconomic shock (the pandemic) in the data, which gave us interesting insights and allowed us to compare the stability of the predictive power of the models. I concluded the research by studying the importance of networks as features or embeddings, showing that behavioural data in itself is powerful.

After establishing the main reasoning behind the three research areas I looked at, I now take a critical approach to our results, discuss our contributions and identify details of our experiments that could be improved and broader risk management techniques that could be worked on.

5.2 Discussion

5.2.1 Illiquid and sparse data application of deep learning

Deep learning models often get deployed when large amounts of data with several features are widely available. In this thesis, we kept the high dimensionality of the data but looked at areas where there might not be complete data available. Using that gap, we saw the insights we could get from such data together with deep learning models; we improved the credit risk default prediction modelling, deployed a single model for several predictions, adapted deep learning models for panel data structure and also quantified the importance of these different data sources. In what follows, I develop these insights further and identify areas of improvement in each area.

In Chapter 2, I developed deep learning techniques to predict a default term structure of mid-cap companies. Within a single multimodal model, I provided the probability of default over the short term from 3 months to up to 3 years. This extends the traditional focus on one year ahead default and more closely aligns with banking practice that must measure corporate risks across a term rather than at a specific time. This was made possible by the deep learning models I relied upon. I found that these advanced computational deep learning methods can be deployed as long as the training program and architecture work well with the data, a challenge modellers must overcome. The models are initially complex to design, and the training periods to optimise the architecture take a long time. However, training the model with newer data is less time-intensive once the model architecture is decided. I developed new differential training strategies, which are now commonplace in fine-tuning large language models. Also, the models allowed a natural combination of the data without dropping much of the initial data. A new multimodal architecture had to be designed to use different data sources. Their performance was better, showing the efficacy of using all the data available; unlike the traditional models for which to work, I would have to remove a lot of input data as the data sources cannot be merged easily.

A few specific research questions remain that could be answered further within this context. I would have liked to introduce constraints on the outputs of the model, where I establish the default curve to be upward-sloping over time. This could be done by weighing the loss function differently instead of averaging, as done in our work. The model does predict with more uncertainty over longer periods, which is expected as we predict more into the future, and the data becomes less useful, which reduces performance. This expectation in the form of constraints might have further improved the model by reducing the search space for the models and bringing in faster optimisation. In our next paper on portfolio optimisation (Chapter 3), I developed custom layers that could apply constraints; I believe a similar setup might work even better. Another area of improvement on the credit side is to incorporate more data types. Textual reports, company filings and news reports have improved the predictive power individually, but explicitly looking at how they impact mid-caps will make different challenges arise, as there might not be much, or maybe much more sparse, public news or reports coverage on a large set of companies. Deep learning models are better suited for prediction in these sparse settings. The data collection process will become challenging. Existing research uses such data types, but not for mid-caps, where the default rates are higher and individual firm risk matters more than large-caps, where defaults are largely due to idiosyncratic issues such as governance. This is an exciting area as I bring more relevant data sources to the problem, which do not naturally lend themselves to such analysis with earlier traditional models.

Another aspect is the challenge of deploying end-to-end deep learning models, as we have seen here. Without customised learning methods that show an understanding of the problem and the data, it might be challenging to extract higher performance, which could lead to wrong conclusions that these models do not perform. I first designed the problem as a multilabel classification one and created a training loss metric suitable to that kind of classification problem. The experimental setup with the computing power available to analyse should also be considered, as model training takes a long time, especially in the hyperparameter search space phase of the experiment. Different combinations of these parameters must be trained on large datasets; therefore, the experimental setup is critical. The choice of hyperparameters also needs to be examined, as the default values in the literature have been designed for different problems. Here, we are adapting for credit risk settings with a general data imbalance (far fewer defaults) as a characteristic of these datasets.

Designing a new deep-learning model is another challenge. I developed a transformer-based model, but I adapted it for panel data. I used only the encoder part, which is similar to the BERT architecture. These architectures have been shown to be effective in classification problems like the ones I tackled; other types of problems, such as time-series to time-series prediction, would require full encoder-decoder transformers. The panel data structure is very similar to the input transformer models, except that the input data has a sequential nature. The data is not the same type at each sequence but has different meanings. In our inputs to the model, I have balance sheet information, cash flow information, and pricing information where each data at a specific time for a firm is related but not the same type. I have shown how the model performs with lower frequency data and incorporated higher frequency data like pricing into the same model. Competing models cannot handle this pricing source, and had I considered the traditional approach of having one data set as input after combining all of them, I would have concluded that pricing information is not relevant to default prediction, but instead, I showed that pricing information was relevant for short-term default prediction. Still, the lower frequency balance sheet and fundamentals drive longer-term probabilities. This confirms the market's experience and how the model correctly captures and models this insight. I also used strong benchmark deep learning models such as TCN and LSTM. I have observed that the architecture of the models matters and that attention, part of the transformer model, was learning a different higher-order relationship to default, improving its performance. From a perspective of credit risk literature, I also saw the importance of missing data. This is generally irrelevant when we look at large-cap companies, but where default risk is relatively high, I found that missing data generated some importance. This happens because mid-cap companies could be changing their accounting methods or delaying the results during stressful times for the company. These stressed situations could lead to default in the future. A company might overcome some of these challenges, but this is a good behavioural data point for how it changes. Later, I saw how important behavioural data is for credit default prediction in a different setting. Some missing data situations might be good when there is a merger and acquisition situation, which again leads to such changes in the accounting data. I did not investigate the causes of this missing information and whether there is any structural pattern to it, but that is an interesting area of inquiry.

Combining some risk management and deep learning areas, another aspect of the paper was the ability to train on multiple data sources within the same model and for different prediction horizons. Traditionally, I would have to develop an ensemble of models to work with multiple prediction horizons and other data types. This approach also allows for more learning of relationships that can be learnt from one data source to another, where together, they can deliver better results than individually. Moreover, the model can be easily extended to different data types, allowing further refinement of classification performance.

Interpretability of the predictions was another area I looked at in this work, as deep learning models are challenging to interpret. I looked at visually understanding the attention matrix of different transformer heads, following some, at the time, nascent experiments in explaining text models. When I averaged these matrices over default and surviving firms, we saw a clear difference in the model parameters thresholds, which gives different colours in our heat maps. It shows the higher-order learning and differentiation between these firms. Another area I looked at was developing a custom Shapley-based method. The existing literature was on individual feature interpretability. However, with the large number of available features, it was difficult to provide such an interpretation, as the existing Shapley approaches will again be complicated to interpret and consume significant computing resources. Instead, I grouped the variables and ran the Shapley-based method by adding and removing groups of variables in all permutations, taking advantage of the additive property of Shapley values. It showed us how the results vary for the importance of data sets over time and which data source is more important. With this approach, I believe it is easier to interpret the results and improve the understanding of data sources. A different grouping of the data could be used regarding the cost of acquiring those features or the complexity of the features, which can recommend what data collection process needs to improve for the credit risk management of such risky companies.

For future work, this multimodal learning process could be extended in terms of the data sources, as I discussed earlier, and by using the same model for different credit risk areas, such as provisioning and pricing of the credit risk. The model could become more robust when trained on different outputs, as with other popular deep learning models in multitask learning. This work successfully adapted the advanced computational methods in the form of deep learning models and used them in a large data setting of mid-cap default prediction. I looked at different aspects of this problem, both from deep learning model design and the multi-horizon default prediction.

5.2.2 Deep learning models as an optimisation structure

In this section, I focus on using deep learning models as an optimisation structure rather than a prediction model. We tested this with portfolio optimisation for mid-cap firms. We provided a different way to generate relationships between firms, modified a deep learning model with custom allocation layers, and tested the robustness over a long period. In this subsection, I develop insights from the networks and portfolio models and identify areas of improvement.

I was the first to examine this problem by combining different research areas. First, I used the volatility series to capture the relationship between firms instead of the return series of firms. Volatility is studied to understand market regimes, and high volatility times lead to more intercorrelated risk than less volatile times. I expected such a series would capture the market and interrelationship dynamics more than return series data. This is particularly significant when relationships fluctuate over time; otherwise, there might be little to learn from that data. The distance correlation measure was more apt, with firms appearing and disappearing over time to establish the relationships.

Next, I borrowed from graph theory methods the TMFG graphs to filter the dense covariance matrix I generated, as this reduces the training complexity of the model, which I encountered in the earlier study on mid-cap firms. This also allowed us to extract more structural information with established methods and has garnered significant attention from the community in conferences and symposiums where we have discussed this work. Finally, the GAT model was trained on these graphs to develop the embeddings with a customised loss function, a modified form of the Sharpe ratio. I used the same performance metric but in its traditional form to measure the model's performance. This is similar to the mean-variance model, where the returns are maximised, keeping the variance to a minimum. I saw that with the same data and similar objectives, the outcomes were very different between a mean-variance model and a deep learning model. Without the deep learning model, the conclusion would be that mean-variance models cannot work on large-scale portfolio optimisation. However, it is the form or the search space that is deployed. Compared to the optimisation procedures developed in deep learning, traditional methods are optimised to lower their computing requirements and make assumptions. For the same problem expressed in the form of a deep learning architecture, the outcome is that they are quite suitable. These architectures can better approximate any complex relationship or outcome due to the hierarchic representation of relationships and the ability to find optimum in high dimensional settings. The GAT model found the optimum embeddings in the high dimensional space. These embeddings needed to be converted to the weights or the available allocation, which led us to develop a custom layer. Such layers are more straightforward to represent using deep learning technology as various constraints or maximising layers are readily available, analogous to objectives and constraints in quadratic optimisation procedures.

Next, I discuss some areas for specific improvement regarding portfolio optimisation based on the data, models and experiment settings. I did not look into using different network creation strategies. Some of the other network creations from returns data could be different, and using another metric, such as the interconnectedness metric used in financial networks, could have generated different networks or used traditional Pearson correlation. There is a caveat, though, with the interconnectedness metric, as this requires high-frequency data, which is challenging to obtain for long periods. Next, about the modelling, there are two ways to make the models end-to-end with deep learning models for portfolio optimisation. They could be deployed to detect the relationships between the firms analogous to a covariance matrix (one of the inputs to the portfolio optimisation models) using an adjusted attention mechanism without the distance correlation and TMFG approach we used. Feng et al. (2022) has used that with GCN and attention mechanism but not at the scale we faced in this thesis. Still, I believe that in areas with plenty of data, whole market covariance matrices could be developed, and these covariance metrics could be tracked over time to detect changes in asset class (stocks, bonds or derivatives) relationships. Another aspect of the modelling is using edge attributes to generate better embeddings. In this study, the GAT model was trained without using the edge features available with the distance correlation metric between any pair of firms in the network; this could be another area to explore further. Finally, end-to-end deep learning models need different training strategies in the experiment settings. This could be achieved with different objective and loss functions, which can be developed from existing traditional portfolio metrics such as max drawdown, information coefficient and others. I also found merging this study with previous portfolio studies challenging, as no studies considered the possibility of default risk within the same portfolio allocation approach. For portfolio optimisation, reinforcement learning is often used in studies as no labelled data is available to make it a supervised learning problem. I took a different approach, using supervised learning with a custom layer that satisfies the portfolio constraints, but much more can be done in this space.

To bring the models to more practical implementations, especially for portfolio managers who manage a multitude of funds, I could adopt more constraints to the portfolio. Practical ones could be such that no position can get more than a certain amount of capital or that positions need to be churned over time and cannot be constant, all of which translates to developing a more customised approach. This constraint-aware approach may help the model train better for specific purposes. Also, different constraints for the portfolio, like sectoral constraints, climate or social-related constraints during the weight allocation stage, might optimise the portfolio differently and satisfy specific business purposes. This helps the investment decision-making process to be more robust and reduces the ex-post risk of finding out the model-generated weights are not applicable; hence, another manual overlay is needed.

We have shown the shortcomings of traditional models when applied to a large-scale portfolio optimisation, especially compared to the GAT model, and the robustness of the model. The model can build upon the initial filtering to develop better embeddings suitable to the optimisation problem. There was noticeable performance improvement in a setting where trading prices are not readily liquid. The model also continues to work over an extensive period of 30 years, encompassing several market environments, such as bullish and bearish environments and a few recessionary periods.

Studying such firms is critical to unlocking excess returns without taking a commensurate risk. This will reward the firms in lowering the cost of capital and increasing the investor base. I expect such models to be widely adopted by institutional investors rather than retail investors, as the computing requirements and the data processing are quite exhaustive. One of the key takeaways is the different results that I get compared to portfolio studies using network topology. Choosing peripheral firms, which have the least connections in the network, might not work well in mid-cap firms with a higher chance of default compared to alternative portfolios. The GAT models chose neither the most central nodes nor the peripheral nodes, which gives a balance of risk and performance. From a portfolio optimisation view, these become relevant when looking at large-scale private firm networks.

Finally, the model turnover is a metric that helps understand how practical it is to implement the portfolio. In large-scale portfolios, using an equal-weighted approach, shown in earlier studies, performs much better than any sophisticated portfolio studies, especially as the number of assets increases. In this study, equal-weighted portfolios do not work, though, as it is costly to trade on every change in the portfolio, and more importantly, it will contain all the risky companies in the portfolio. A single default or bankruptcy will impact the portfolio much more than collective underperformance in a large set of firms. Also, the churn in the equal-weighted portfolio is significant as the firms' universe changes quickly over time. The GAT model was more restrictive than others, with the lowest turnover in the firms and avoiding peripheral or firms that could default or go bankrupt.

As a future study, this could be adapted to discover market regimes. These occur when markets trade with similar regimes, especially the regimes of high and low volatility, which are studied in the literature. Here, one could use the graph embeddings to understand different market regimes and see if these embedding generations have similarities in similar market environments. Networks have been used to study contagion and systematic risk, and I believe the GAT embeddings could also serve a similar purpose. I applied large-scale portfolio optimisation for risky mid-cap firms and found the GAT-generated allocations are robust over time and cost less in portfolio turnover. I also used a deep learning model as a generic optimisation structure rather than a prediction model, which is how often they are used.

5.2.3 Multimodal deep learning models for private large-scale datasets

In this section, we revisit the application of multimodal deep learning models, working with internal data of large financial institutions and focusing on behavioural data on SMEs' credit lines. We found network data to be relevant, but behavioural data is quite powerful. We also found network data to be robust to market changes. I will go into more detail in this section.

Chapter 4 studies these SMEs with multimodal deep learning models. As initially discussed in the Introduction, SMEs are an essential source of job generation in any economy but suffer due to data limitations when accessing credit. I looked to improve the default prediction models of SMEs using behavioural data and internally generated network data. The behavioural data was quite predictive of the default, and simpler models sufficed to explain them. However, network-related variables were also important, and I found these features to be predictive in themselves but not as powerful as behavioural variables. This confirmed earlier studies that showed the power of behavioural variables in default prediction and how simple models can extract the relationship necessary to predict default. The behavioural data, however, may not always be available as it requires contracts that financial institutions can monitor over time. I focused on credit lines because they generate that behavioural data. Once established, the network data could be used when behavioural data is absent, as the relationships do not change drastically over time.

Another contribution from the study was that behavioural data could be susceptible to macroeconomic shocks, as I have observed that firms' behaviour changes drastically over time. In my sample, the default rates for the population go up during a big shock to the market. Using behavioural models in those environments would not produce reliable results as the previous behaviours are no longer relevant. In those periods, the networks were robust, which could offer learning of the economic shocks transmitted. The network models rely on the neighbours of the firms, so their behaviour could be localised but still have transmission shocks spread, which improves the model's predictions during volatile times.

There are further areas of development. The network data are heterogeneous, i.e. the nodes are of different types, such as firms that only own other firms but are not part of the population for which I want to predict default. Various graph neural network methods for heterogeneous graphs were developed during the last year, and using them could further improve the models. I used heterogeneity by masking the data, which is a basic method in our view when looking at deep learning graph neural network developments. On the credit side, one of the improvements could be in using more extended data periods. We were limited to using specific data periods from 2018 to 2021. The data before 2018 were different, with changes in internal procedure indicating that the data definitions before the period were different, and restrictions on data extraction meant that we could not use the data after 2021. A more varied period could have tested the model more robustly than the out-of-universe selection and masking procedures. In addition, having relevant macroeconomic data for the firms could further improve the models, as these firms are subject to systemic risk. Still, different sectors react differently to it, and the network data could have picked up on such differences as it has an integrated view of the market. Another area of study can be to look into the default shock transmission in the network, i.e., how many years after neighbouring firms show similar signs after default or how the dynamics of the risk propagation across the network.

The deep learning models did not perform better than standard alternatives with behavioural data, as there are straightforward relationships in behavioural variables that traditional models can use to predict default. However, with no behavioural data, just using networks as the inputs to these models, they perform sufficiently accurately to be used on their own. The GAT model performs particularly well compared to a similar deep neural network model, which shows some network structure in the data being exploited by the GAT, which other models cannot. They generate embeddings which, when run together with behavioural data, produce better results for the traditional models. However, I feel more research is needed in this area with the 1% improvement I got in the results using graph embeddings compared to conventional network features. The costs of initially maintaining and developing these models over time might look too high compared to the improvement. Still, I believe these models are more suitable for changing data distributions or economic environments, so the embeddings could be more robust as we have seen they had a stable performance during a pandemic shock while behavioural models' performance deteriorated.

We also conducted an explanatory analysis of the behavioural and network variable data to gauge the relative importance of these data sources. Confirming our earlier studies, I found that missing data in the behavioural data is also important in predictions. Missing data may not be completely random, and churn may signify some change in events within the firm. In behavioural data, I found that higher utilisation of the contract limit is another marker for likely stress in the future. This confirms earlier studies, which showed similar behaviour by firms. One of the reasons for the underperformance of the deep learning model is that they are good at finding the turning points or breaks in the data, in our case, periods of high default transitioning towards a lower default period. There could be scope for future studies on this topic that deploy models for different behavioural markers. In our research, I classified someone as default that is overdue on their payments for over 90 days, but I could use an earlier period to detect stress the first time it happens, like trying to understand which accounts go into arrears the first time; deep learning models might better answer these questions. As I saw with just network data, they could be trained to find some higher-order relationships to detect default prediction better than traditional models. Data collection methods also play a role in the performance of complex deep learning models. I have seen the network data improve over time in the data we used in our study, with ownership data coming later, so we had periods when there was only one dataset. With richer representation, network-based models improve their performance. Established network metrics could easily represent the network data, which increases performance over just the behavioural data model by adding to the models. The deep learning embeddings still improved the model, but not as much as I hoped they would. However, the model embeddings can be even more potent for prediction with better data representation. The behavioural data could also be further improved by adding even more features that could be generated, such as intra-firm credit line contract changes; I looked at only the maximum value contract in this study.

Credit lines are major predictors of changing behaviour and could serve as an input to a firm's other credit contract models. This will enable closer tracking and better pricing of risk, both for the firm and the financial institution that provides the credit. I have seen that the internal network data generated by the firm is quite helpful because it is explicit, and advanced analytical methods need to be deployed to extract value from it. The final predictive model was traditional, but inputs needed to be sourced by generating new behavioural and network characteristics.

5.3 Future work

I identify a few areas of work with more general themes throughout this work. This work combines credit risk studies, advanced computational methods using deep learning and novel datasets. I looked at often illiquid mid-caps and understood various market risks, and I also integrated the studies for private large-scale datasets of SME firms. On the credit risk side, I expect to have models that can be developed together with publicly available data models and private datasets. Here, I worked with publicly available data or private datasets. Still, some of the risks inform each other, and using macroeconomic embeddings generated from public data models in private data models could be informative of the current macroeconomic environments. These could further improve the model's predictive powers. I would also like to see the work extended into the pricing of the credit instruments, such that they are no longer classification problems but regression problems with precise objective functions. Deep learning methods can find the optimum values in a higher-dimensional space to work with complex objective functions.

Another area of further study is automating the architectures of the deep learning models. I chose specific deep-learning models based on the data and the nature of the problem. For example, with the credit default prediction problem for mid-caps, I used transformer models because the underlying input for these models has a panel data-like structure. For the subsequent two studies, networks were important, which meant using the GAT models. Deep learning architecture's initial and final layers play important roles, such as how we combine the embeddings and the need for a feed-forward network in an intermediate layer within the architectures. Sometimes, it might not be clear which model is to be used among so many available models. I have identified three broad areas in the literature: pruning, quantisation, and automated machine learning (AutoML). Pruning removes deep learning layers, which reduces the number of parameters in the model to be learnt. The impact on model performance and the amount of pruning must be tracked to identify the optimal number of layers. The second measure is quantisation, which decreases the parameter's precision from 64 bits to 16 or 8 bits for the learnable parameters. This reduces computational complexity and has shown that it does not impact model performance metrics as much. Finally, the AutoML approach could be tried, where the architecture is learnt based on the data and the nature of prediction. Given the available alternative data types and how different models perform against different data types, this approach would become needed in the future. Much of this project's literature and research was in the development phases. While our data was quite complex, much research time was spent on the data collection and making them fit the input models. I would leave these changes as the significant next steps in advancing research, especially when using scalable computational methods that work on population-level datasets. Together with such advanced analytical methods, we could further improve the models, as the results for improved performance are still there.

Among the three significant areas of risk management problems that I studied, I had to develop customisable loss functions in two of them; I believe it is an understudied area to determine which loss function is best suitable to a given problem and the reasons for this. The model uses the loss function as a training objective to minimise, so choosing this needs to be well thought out. Several loss functions are developed, but which ones to use depends on the data and the expected output.

Combining different data sources and how we choose to combine them is another area of study that will come from computing research. This work showed methods for using various data types and developing multimodal models. Much of the research assumes that some of the costs are worth the complexity, which is generally the case as the amount of capital invested in solving such problems is immense, but it needs to be carefully considered. Contrary to expectations, the models' computational or training costs are trivial in terms of the amount of work needed to maintain these models. Monitoring of the model's performance needs to be developed. As we have seen in Chapter 4, we could see models where the traditional, more explainable models are at the front, but deep learning models could be used to develop some intricate features as input to these models. Overall, we should be able to work on a model that incorporates varied data sets and applies them to different prediction and optimisation problems, making the model robust and improving over time. I believe that seeking to achieve this goal, in a variety of other applications, still offers a promising area of research. It could also impact the industry by advancing best practices for using different data sources.

Appendix A

Mid-cap default prediction

A.1 Fundamental channel data variables

Accounting and market-based ratios calculated from raw data are shown in Table A.1. This data is used as part of fundamental channel data in the models. Ratios that need price or market performance is used in the market channel. Ratios used are derived from Mai et al. (2019)

TABLE A.1: Ratios and Variable description

Ratios Derived	**Description**
ACTLCT	Current Assets/Current Liabilities
APSales	Accounts Payable/Sales
CASHAT	Cash and Short-term assets/Total Assets
CHLCT	Cash/Current Liabilities
EBITDA/AT	EBITDA/Total Assets
EBITAT	EBIT/Total Assets
EBITSALE	EBIT/Sales
FAT	Total Debts/Total Assets
INVCHINVT	Growth of Inventories /Inventories
REAT	Retained Earnings/Total Asset
INVTSALE	Inventories/Sales
RELCT	Retained Earnings/Current Liabilities
LTAT	Total Liabilities/Total Assets
SALEAT	Sales/Assets
LOG(AT)	Log(Total Assets)
LOG(Sales)	Log(Sale)

textbfRatios Derived	multicolumn1l**Description**
CASHMTA	Cash and Short-term assets/(Market Equity + Liabilities)
LTMTA	Total Liabilities/(Market Equity + Liabilities)
MB	Market-to-Book Ratio
NIAT	Net Income/Total Asset
NIMTA	Net Income/(Market Equity + Total Liabilities)
NISALE	Net Income/Sales
PRICE	Log(Price)
SEQAT	Equity/Total Asset
WCAPAT	Working Capital/Total Assets
LCTCHAT	(Current Liabilities – Cash)/Total Asset
LCTAT	Current Liabilities/Total Asset
LCTSALE	Current Liabilities/Sales
LCTLT	Current Liabilities/Total Liabilities
RSIZE	Log(Market Capitalization)
SIGMA	Stock Volatility
EXCESSRETURN	Excess return over S&P 500

A.2 Market channel data variables

The various market variables are used to place the performance of the company. 1 month and 3-month returns of each of these variables are included

TABLE A.2: Market channel

Market variable & Data Stream Variable	
S&P 500 Return	S&P 500 COMPOSITE - PRICE INDEX
Corporate Index	ICE BofA US Corporate Index - Yld to Mat convent
High Yield Index	ICE BofA US High Yield Index - Yld to Mat convent
Treasury Index	ICE BofA US Treasury Index - Yld to Mat convent

Appendix B

Network-Enhanced Credit Risk models

B.1 Model input variables by type

The Shap feature variables are listed in two tables, Table B.1 and Table B.2 are given in this appendix. We describe the variables and also group them into a relevant category to understand them better. The bank category contains the variables that are defined by our financial institution; while they are not static, this category is not completely behavioural either. Static features are data that we do not expect to change over time. The other variables are self-explanatory.

B.2 Multimodal specification

The multimodal architecture can be modified to create consistent inputs for the model without further transformations. We mask the data or the layers that we do not need while keeping the training procedure the same, with similar learning rate, optimisers and loss function. This actually allows us to see the impact of only the changes we make. Figure B.1 shows the masking with a cross mark. For example, Figure B.1(a) shows when the GATConv blocks are masked it makes a neural network (NN) model. The inputs only go through the Linear model and we can understand the impact of the GAT blocks in our model. Similarly, in Figure B.1(a), we mask the behavioural data so that we use only the minor network features as inputs; the models need some input and we chose to use network features as these come from the same network again. We call this model the GAT NW model as the input is only network information. This represents the model used for the second half of results presented in Table 4.4 where

TABLE B.1: Important features from the SHAP analysis

Variable	**Type of variable**	**Description**
status	Behavioural	Status of the contracts <15, <30days due
exposure	Behavioural	Exposure in the current contract
limit	Bank	Bank defined limit on the contract
bankingrelationship	Network	Degree centrality of the firm in the network
irregular_count	Behavioural	Number of current contracts that are irregular
last_activity	Behavioural	Last activity in the current contract
historical_irregular_contracts	Behavioural	Number of historical irregular contracts
historical_irregular_amount	Behavioural	Historical irregular amount
utilization	Behavioural	Ratio of exposure to limit
contract_count	Behavioural	Total number of contracts
total_exposure	Behavioural	Total exposure in all contracts
total_limit	Bank	Total limit over all contracts
min_utilization	Behavioural	Min utilisation across all contracts
most due contract status	Behavioural	Of all current currents the contract that is most due
location_code	Static	Most frequent location of the firm
max_current_delinquency_state	Behavioural	Maximum delinquency state of all contracts
na_historical_irregular_current	Missing indicator	The worst state the contract historically has been in

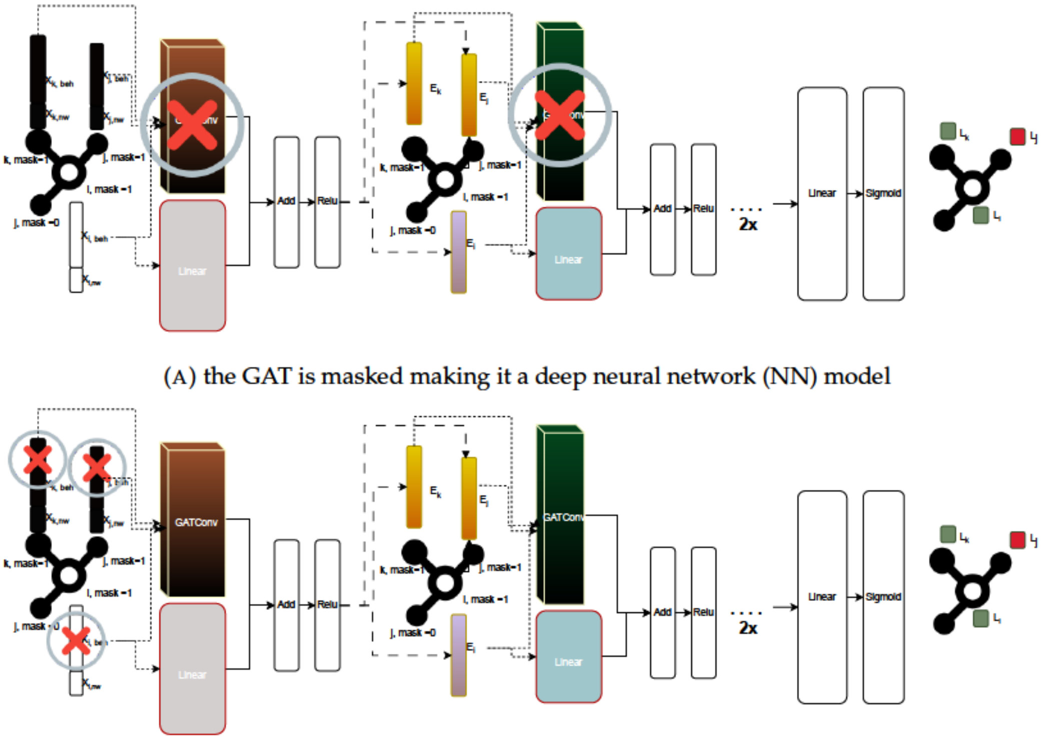
TABLE B.2: Continuation of important features from the SHAP analysis

Variable	Type of variable	Description
na_des_historical_irregular_contract	Missing indicator	Description of the worst state the contract has been in
na_timestamp_historical	Missing indicator	Tmestamp of entry of the worst state
na_accountant	Missing indicator	Kind of accountant for the contract firm
size_P1	Static	Size of the firm, one hot encoded variable P1,P2, P3
size_P2	Static	Size of the firm, one hot encoded variable P1,P2, P3
pr	Network	Page Rank
dc	Network	Degree centrality of the firm in the network
bc	Network	Betweenness centrality
average_bankingrelationship	Network	Average degree of linkage of all neighbours
total_bankingrelationship	Network	Total degree of linkage of all neighbours

we used only network data as input. This helps us to clearly understand the power of the network data in the model and how it impacts performance.

B.3 Training evolution

Here, we present the evolution of training over a single graph. We used three learning rate regimes in the model, initially with a higher learning rate and gradually reducing it. The chart in Figure B.2 shows the training over 15000 epochs. We test the efficacy of the fit on the same training data but using the AUC performance metric. The model is trained for accuracy but the validation is done based on the AUC. This gives a reasonable check on overfitting in the model.



(A) the GAT is masked making it a deep neural network (NN) model

(B) Model setup with no behavioural data (GAT NW); each node's features are excluded from training

FIGURE B.1: Multimodal training setup

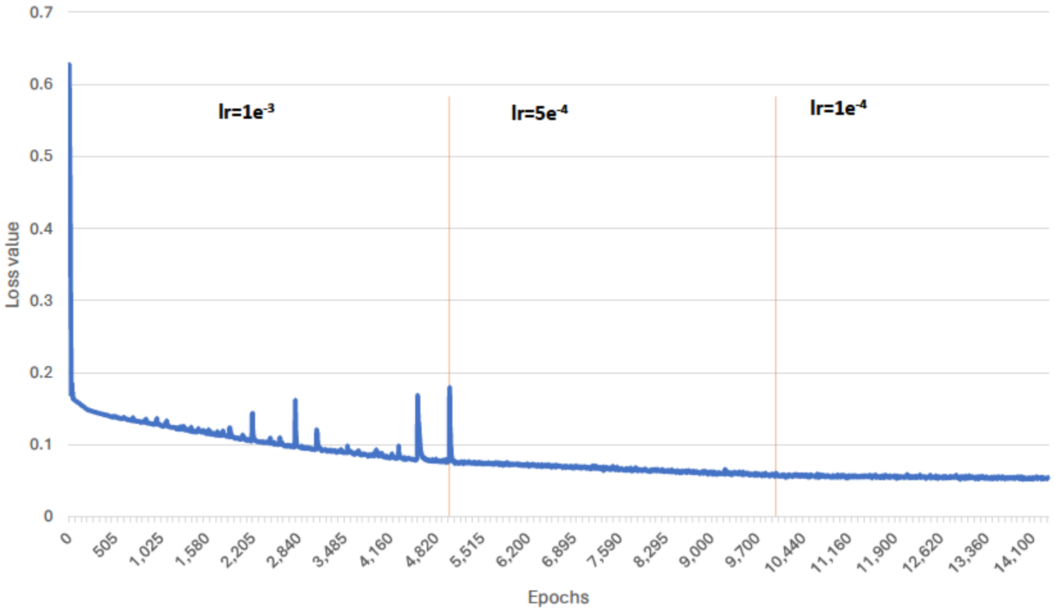


FIGURE B.2: Training of the multimodal model; decreasing loss with stability

References

V. V. Acharya, Y. Amihud, and S. T. Bharath. Liquidity risk of corporate bond returns: Conditional approach. *Journal of Financial Economics*, 110(2):358–386, November 2013.

Viral V. Acharya, Heitor Almeida, Filippo Ippolito, and Ander Perez-Orive. Credit lines and the liquidity insurance channel. *Journal of Money Credit and Banking*, 53(5): 901–938, August 2021. ISSN 0022-2879.

Ikmal Adian, Djeneba Doumbia, Neil Gregory, Alexandros Ragoussis, Aarti Reddy, and Jonathan Timmis. *Small and Medium Enterprises in the Pandemic : Impact, Responses and the Role of Development Finance*. World Bank, Washington, DC, September 2020.

S. Agarwal, B. W. Ambrose, and C. L. Liu. Credit lines and credit utilization. *Journal of Money Credit and Banking*, 38(1):1–22, February 2006. ISSN 0022-2879.

Sumit Agarwal, Vincent Y. S. Chen, and Weina Zhang. The Information Value of Credit Rating Action Reports: A Textual Analysis. *Management Science*, 62(8): 2218–2240, August 2016. ISSN 0025-1909.

Hafiz A. Alaka, Lukumon O. Oyedele, Hakeem A. Owolabi, Vikas Kumar, Saheed O. Ajayi, Olugbenga O. Akinade, and Muhammad Bilal. Systematic review of bankruptcy prediction models: Towards a framework for tool selection. *Expert Systems with Applications*, 94:164–184, March 2018. ISSN 0957-4174.

Maher Ala'raj, Maysam F. Abbod, and Munir Majdalawieh. Modelling customers credit card behaviour using bidirectional LSTM neural networks. *Journal of Big Data*, 8(1):69, May 2021. ISSN 2196-1115.

Fawaz Khaled Alarfaj, Iqra Malik, Hikmat Ullah Khan, Naif Almusallam, Muhammad Ramzan, and Muzamil Ahmed. Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access*, 10:39700–39715, 2022. ISSN 2169-3536.

Sherly Alfonso-Sánchez, Jesús Solano, Alejandro Correa-Bahnsen, Kristina P. Sendova, and Cristián Bravo. Optimizing credit limit adjustments under adversarial goals using reinforcement learning. *European Journal of Operational Research*, 315(2): 802–817, June 2024. ISSN 0377-2217.

Edward I Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23(4):589–609, 1968. ISSN 0022-1082.

Edward I. Altman, Robert G. Haldeman, and P. Narayanan. ZETATM analysis A new model to identify bankruptcy risk of corporations. *Journal of Banking & Finance*, 1(1): 29–54, June 1977. ISSN 0378-4266.

Edward I. Altman, Małgorzata Iwanicz-Drozdowska, Erkki K. Laitinen, and Arto Suvas. A race for long horizon bankruptcy prediction. *Applied Economics*, 52(37): 4092–4111, 2020.

Edward I. Altman and Gabriele Sabato. Modeling Credit Risk for Smes: Evidence from the Us Market. In *Managing and Measuring Risk*, volume Volume 5 of *World Scientific Series in Finance*, pages 251–279. WORLD SCIENTIFIC, June 2012. ISBN 978-981-4417-49-5.

Edward I. Altman, Gabriele Sabato, and Nicholas Wilson. The value of non-financial information in small and medium-sized enterprise risk management. *The Journal of Credit Risk*, 6(2):95–0_7, 2010. ISSN 17446619.

Jeffery D Amato and Eli M Remolona. The credit spread puzzle. *BIS Quarterly Review, December*, 2003.

Mengmeng Ao, Li Yingying, and Xinghua Zheng. Approaching Mean-Variance Efficiency for Large Portfolios. *The Review of Financial Studies*, 32(7):2890–2919, July 2019. ISSN 0893-9454.

Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. A survey on modern trainable activation functions. *Neural Networks*, 138:14–32, June 2021. ISSN 0893-6080.

Fernando M. Aragon, Alexander Karaivanov, and Karuna Krishnaswamy. Credit lines in microcredit: Short-term evidence from a randomized controlled trial in India. *Journal of Development Economics*, 146:102497, September 2020. ISSN 0304-3878.

M. Arenas Parra, A. Bilbao Terol, and M. V. Rodríguez Uría. A fuzzy goal programming approach to portfolio selection. *European Journal of Operational Research*, 133(2):287–297, January 2001. ISSN 0377-2217.

Doron Avramov, Si Cheng, and Lior Metzker. Machine Learning Vs. Economic Restrictions: Evidence from stock return predictability. *Management Science*, 69(5): 2587–2619, May 2023. ISSN 0025-1909.

Meghana Ayyagari, Thorsten Beck, and Asli Demirguc-Kunt. Small and medium enterprises across the globe. *Small Business Economics*, 29(4):415–434, December 2007. ISSN 1573-0913.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv e-prints*, page arXiv.1607.06450, July 2016.

Jennie Bai, Robert S. Goldstein, and Fan Yang. Is the credit spread puzzle a myth? *Journal of Financial Economics*, 137(2):297–319, 2020. ISSN 0304-405X.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Gah-Yi Ban, Noureddine El Karoui, and Andrew E. B. Lim. Machine Learning and Portfolio Optimization. *Management Science*, 64(3):1136–1154, March 2018. ISSN 0025-1909.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, June 2020. ISSN 1566-2535.

Basel Committee on Banking Supervision. Basel II: International convergence of capital measurement and capital standards - A revised framework. Technical report, Bank for International Settlements, 2003.

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction Networks for Learning about Objects, Relations and Physics. In *NIPS*, January 2016.

W. H. Beaver, M. Correia, and M. F. McNichols. Do differences in financial reporting attributes impair the predictive ability of financial ratios for bankruptcy? *Review of Accounting Studies*, 17(4):969–1010, 2012.

William H Beaver. Financial ratios as predictors of failure. *Journal of Accounting Research*, pages 71–111, 1966. ISSN 0021-8456.

Thorsten Beck and Asli Demirguc-Kunt. Small and medium-size enterprises: Access to finance as a growth constraint. *Journal of Banking & finance*, 30(11):2931–2943, 2006.

David Beckworth, Kenneth P. Moon, and J. Holland Toles. Monetary policy and corporate bond yield spreads. *Applied Economics Letters*, 17(12):1139–1144, 2010.

Tony Bellotti and Jonathan Crook. Forecasting and stress testing credit card default using dynamic models. *International Journal of Forecasting*, 29(4):563–574, October 2013. ISSN 01692070.

Anne-Sophie Bergeres, Philippe d'Astous, and Georges Dionne. Is there any dependence between consumer credit line utilization and default probability on a term loan? Evidence from bank-customer data. *Journal of Empirical Finance*, 33: 276–286, September 2015. ISSN 0927-5398.

Joost Berkhout and Bernd F. Heidergott. Analysis of Markov Influence Graphs. *Operations Research*, 67(3):892–904, May 2019. ISSN 0030-364X.

Jose M. Berrospide and Ralf R. Meisenzahl. The real effects of credit line drawdowns. *International Journal of Central Banking*, 18(3):321–397, September 2022. ISSN 1815-4654.

Dimitris Bertsimas and Ryan Cory-Wright. A Scalable Algorithm for Sparse Portfolio Selection. *INFORMS Journal on Computing*, 34(3):1305–1840, January 2022.

Zhicun Bian, Yin Liao, Michael O'Neill, Jing Shi, and Xueyong Zhang. Large-scale minimum variance portfolio allocation using double regularization. *Journal of Economic Dynamics & Control*, 116:103939, July 2020. ISSN 0165-1889.

Fischer Black and John C. Cox. Valuing Corporate Securities: Some Effects of Bond Indenture Provisions. *The Journal of Finance*, 31(2):351–367, 1976. ISSN 1540-6261.

P. Bonami and M. A. Lejeune. An exact solution approach for portfolio optimization problems under stochastic and integer constraints. *Operations Research*, 57(3): 650–670, 2009-05/2009-06. ISSN 0030-364X.

Michael Branch, Lisa R. Goldberg, and Pete Hand. A guide to ESG portfolio construction. *The Journal of Portfolio Management*, 45(4):61–66, March 2019. ISSN 0095-4918, 2168-8656.

Aida Brankovic, Marjan Hosseini, and Luigi Piroddi. A distributed feature selection algorithm based on distance correlation with an application to microarrays. *Ieee-Acm Transactions on Computational Biology and Bioinformatics*, 16(6):1802–1815, 2019. ISSN 1545-5963.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. ISSN 1573-0565.

Jana Bricco and TengTeng Xu. Interconnectedness and contagion analysis: A practical framework. *International Monetary Fund*, October 2019.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117, April 1998. ISSN 0169-7552.

Miriam Bruhn, Martin Hommes, Mahima Khanna, Sandeep Singh, Aksinya Sorokina, and Joshua Wimpey. MSME finance gap: Assessment of the shortfalls and opportunities in financing micro, small, and medium enterprises in emerging markets. Technical report, World Bank, 2017.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs, May 2014.

H. Buehler, L. Gonon, J. Teichmann, and B. Wood. Deep hedging. *Quantitative Finance*, 19(8):1271–1291, August 2019. ISSN 1469-7688.

Jie Cai and CFA Houge, Todd. Long-term impact of russell 2000 index rebalancing. *Financial Analysts Journal*, 64(4):76–91, July 2008. ISSN 0015-198X.

Giulio Cainelli, Sandro Montresor, and Giuseppe Vittucci Marzetti. Production and financial linkages in inter-firm networks: Structural variety, risk-sharing and resilience. *Journal of Evolutionary Economics*, 22(4):711–734, September 2012. ISSN 1432-1386.

Raffaella Calabrese, Giampiero Marra, and Silvia Angela Osmetti. Bankruptcy prediction of small and medium enterprises using a flexible binary generalized extreme value model. *Journal of the Operational Research Society*, 67(4):604–615, April 2016. ISSN 1476-9360.

Nelson Camanho, Pragyan Deb, and Zijun Liu. Credit rating and competition. *International Journal of Finance & Economics*, 27(3):2873–2897, 2022. ISSN 1099-1158.

Rosella Castellano and Roy Cerqueti. Mean–Variance portfolio selection in presence of infrequently traded stocks. *European Journal of Operational Research*, 234(2):442–449, April 2014. ISSN 0377-2217.

Fei Mei Chan and Craig Lazzara. Mid-cap: A sweet spot for performance. https://www.spglobal.com/spdji/en/documents/education/practice-essentials-mid-cap-a-sweet-spot-for-performance.pdf, 2015.

Abel Chandra, Laura Tünnermann, Tommy Löfstedt, and Regina Gratz. Transformer-based deep learning for predicting protein properties in the life sciences. *eLife*, 12:e82819, January 2023. ISSN 2050-084X.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, August 2016. ISBN 978-1-4503-4232-2.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep*

Learning for Recommender Systems, DLRS 2016, pages 7–10, New York, NY, USA, September 2016. Association for Computing Machinery. ISBN 978-1-4503-4795-2.

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks, June 2020.

Younghwan Cho and Jae Wook Song. Hierarchical risk parity using security selection based on peripheral assets of correlation-based minimum spanning trees. *Finance Research Letters*, 53:103608, May 2023. ISSN 1544-6123.

Gabriel Chodorow-Reich, Olivier Darmouni, Stephan Luck, and Matthew Plosser. Bank liquidity provision across the firm size distribution. *Journal of Financial Economics*, 144(3):908–932, June 2022. ISSN 0304-405X.

Petr Chunaev. Community detection in node-attributed social networks: A survey. *Computer Science Review*, 37:100286, August 2020. ISSN 1574-0137.

European Commission. Fitness Check on the EU framework for public reporting by companies. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021SC0081, April 2021.

Martijn Cremers, Ankur Pareek, and Zacharias Sautner. Short-Term Investors, Long-Term Investments, and Firm Value: Evidence from Russell 2000 Index Inclusions. *Management Science*, 66(10):4535–4551, October 2020. ISSN 0025-1909.

Jonathan N. Crook, David B. Edelman, and Lyn C. Thomas. Recent developments in consumer credit risk assessment. *European Journal of Operational Research*, 183(3): 1447–1465, December 2007. ISSN 0377-2217.

Peter Crosbie and Jeff Bohn. Modeling default risk: Modeling methodology. *KMV corporation*, 2003.

Xolani Dastile, Turgay Celik, and Moshe Potsane. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, 91: 106–263, 2020. ISSN 1568-4946.

Frank De Jong and Joost Driessen. Liquidity risk premia in corporate bond markets. *The Quarterly Journal of Finance*, 02(02):1250006, June 2012. ISSN 2010-1392.

Victor DeMiguel, Lorenzo Garlappi, Francisco J. Nogales, and Raman Uppal. A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science*, 55(5):798–812, May 2009a. ISSN 0025-1909.

Victor DeMiguel, Lorenzo Garlappi, and Raman Uppal. Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The Review of Financial Studies*, 22(5):1915–1953, May 2009b. ISSN 0893-9454.

Asli Demirgüç-Kunt, Maria Soledad Martinez Peria, and Thierry Tressel. The global financial crisis and the capital structure of firms: Was the impact more severe among SMEs and non-listed firms? *Journal of Corporate Finance*, 60:101514, February 2020. ISSN 0929-1199.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805 [cs]*, October 2018.

L Di Bella, A Katsinis, and J Lagüera-González. SME performance review 2022/2023. *Publications Office of the European Union*, Annual Report on European SMEs, 2023.

Francis X. Diebold and Kamil Yilmaz. Better to give than to receive: Predictive directional measurement of volatility spillovers. *International Journal of Forecasting*, 28(1):57–66, January 2012. ISSN 0169-2070.

Francis X. Diebold and Kamil Yilmaz. On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics*, 182(1):119–134, September 2014. ISSN 0304-4076.

Zhi-Long Dong, Fengmin Xu, and Yu-Hong Dai. Fast algorithms for sparse portfolio selection considering industries and investment styles. *Journal of Global Optimization*, 78(4):763–789, December 2020. ISSN 1573-2916.

D. Du, R. Elkamhi, and J. Ericsson. Time-varying asset volatility and the credit spread puzzle. *Journal of Finance*, 74(4):1841–1885, 2019. ISSN 0022-1082.

P. du Jardin. Bankruptcy prediction using terminal failure processes. *European Journal of Operational Research*, 242(1):286–303, 2015.

Philippe du Jardin and Eric Séverin. Forecasting financial failure using a Kohonen map: A comparative study to improve model stability over time. *European Journal of Operational Research*, 221(2):378–396, September 2012. ISSN 0377-2217.

Darrell Duffie, Leandro Saita, and Ke Wang. Multi-period corporate default prediction with stochastic covariates. *Journal of Financial Economics*, 83(3):635–665, March 2007. ISSN 0304-405X.

Douglas Dwyer, Ahmet Kocagil, and Roger Stein. The Moody's KMV RiskCalc v3. 1 Model: Next-generation technology for predicting private firm credit risk. *Moody's KMV*, 2004.

Jeffrey L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 1551-6709.

Eric G. Falkenstein, Andrew Boral, and Lea V. Carty. Riskcalc for Private Companies: Moody's Default Model, November 2000.

Jianqing Fan, Yuan Liao, and Martina Mincheva. Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 75(4):603–680, September 2013. ISSN 1369-7412.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference*, WWW '19, pages 417–426, New York, NY, USA, May 2019. Association for Computing Machinery. ISBN 978-1-4503-6674-8.

Simone Farinelli, Manuel Ferreira, Damiano Rossello, Markus Thoeny, and Luisa Tibiletti. Beyond Sharpe ratio: Optimal asset allocation using different performance ratios. *Journal of Banking & Finance*, 32(10):2057–2063, October 2008. ISSN 0378-4266.

P. Feldhutter and S. M. Schaefer. The myth of the credit spread puzzle. *Review of Financial Studies*, 31(8):2897–2942, 2018.

Shibo Feng, Chen Xu, Yu Zuo, Guo Chen, Fan Lin, and Jianbing XiaHou. Relation-aware dynamic attributed graph attention network for stocks recommendation. *Pattern Recognition*, 121:108119, January 2022. ISSN 0031-3203.

Linton C. Freeman. A Set of Measures of Centrality Based on Betweenness. *Sociometry*, 40(1):35–41, 1977. ISSN 0038-0431.

Carol Ann Frost. Credit rating agencies in capital markets: A review of research evidence on selected criticisms of the agencies. *Journal of Accounting, Auditing & Finance*, 22(3):469–492, July 2007. ISSN 0148-558X.

FSB. 2024 List of Global Systemically Important Banks (G-SIBs). https://www.fsb.org/2024/11/2024-list-of-global-systemically-important-banks-g-sibs/, November 2024.

Wei Ge. The curious case of the mid-cap premium. *The Journal of Index Investing*, 8(4): 22–30, February 2018. ISSN 2154-7238, 2374-135X.

Robert Geske. The valuation of corporate liabilities as compound options. *The Journal of Financial and Quantitative Analysis*, 12(4):541–552, 1977. ISSN 0022-1090.

Hossein Gholamalinezhad and Hossein Khosravi. Pooling methods in deep neural networks, a review, September 2020.

Gotz Giese. Enhancing creditrisk+. *Risk*, 16(4):73–77, 2003.

Kay Giesecke and Stefan Weber. Cyclical correlations, credit contagion, and portfolio losses. *Journal of Banking & Finance*, 28(12):3009–3036, December 2004. ISSN 0378-4266.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 513–520, June 2011. ISBN 978-1-4503-0619-5.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting Deep Learning Models for Tabular Data. In *Advances in Neural Information Processing Systems*, volume 34, pages 18932–18943. Curran Associates, Inc., 2021.

Daniele Grattarola and Cesare Alippi. Graph Neural Networks in TensorFlow and Keras with Spektral [Application Notes]. *IEEE Computational Intelligence Magazine*, 16(1):99–106, February 2021. ISSN 1556-6048.

Daniel L. Greenwald, John Krainer, Board of Governors of the Federal Reserve, Pascal Paul, and Federal Reserve Bank of San Francisco. The credit line channel. *Federal Reserve Bank of San Francisco, Working Paper Series*, pages 1.000–96.000, July 2020.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35:507–520, December 2022.

Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864, New York, NY, USA, August 2016. Association for Computing Machinery. ISBN 978-1-4503-4232-2.

Massimo Guidolin and Federica Ria. Regime shifts in mean-variance efficient frontiers: Some international evidence. *Journal of Asset Management*, 12(5):322–349, November 2011. ISSN 1479-179X.

G. Gupton, C. Finger, and M. Bhatia. Creditmetrics-technical document. *JP Morgan, New York*, 1997.

Aric Hagberg, Pieter J. Swart, and Daniel A. Schult. Exploring network structure, dynamics, and function using NetworkX. Technical Report LA-UR-08-05495; LA-UR-08-5495, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), January 2008.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):87–110, January 2023. ISSN 1939-3539.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, December 2015.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, July 2006. ISSN 0899-7667.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, January 2025. ISSN 1476-4687.

Yuh-Jong Hu and Shang-Jen Lin. Deep reinforcement learning for optimizing finance portfolio management. In *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pages 14–20, Dubai, United Arab Emirates, February 2019.

Darien Huang, Christian Schlag, Ivan Shaliastovich, and Julian Thimme. Volatility-of-Volatility Risk. *Journal of Financial and Quantitative Analysis*, 54(6): 2423–2452, December 2019. ISSN 0022-1090, 1756-6916.

John C. Hull and Alan D. White. Valuing Credit Default Swaps: No counterparty default risk. *The Journal of Derivatives*, 8(1):29–40, August 2000. ISSN 1074-1240, 2168-8524.

Xiaoming Huo and Gábor J. Székely. Fast computing for distance covariance. *Technometrics*, 58(4):435–447, October 2016. ISSN 0040-1706.

Summaira Jabeen, Xi Li, Muhammad Shoib Amin, Omar Bourahla, Songyuan Li, and Abdul Jabbar. A review on methods and applications in multimodal deep learning. *ACM Trans. Multimedia Comput. Commun. Appl.*, 19(2s):76:1–76:41, February 2023. ISSN 1551-6857.

R. A. Jarrow and S. M. Turnbull. Pricing derivatives on financial securities subject to credit risk. *Journal of Finance*, 50(1):53–85, 1995.

Robert A. Jarrow and Dilip B. Madan. A characterization of complete security markets on a brownian filtration. *Mathematical Finance*, 1(3):31–43, 1991. ISSN 1467-9965.

Robert A. Jarrow and Stuart M. Turnbull. The intersection of market and credit risk. *Journal of Banking & Finance*, 24(1):271–299, January 2000. ISSN 0378-4266.

Tyler Jayroe. A big role for small and middle-market private equity investments. https://am.jpmorgan.com/us/en/asset-management/adv/insights/portfolio-insights/alternatives/a-big-role-for-small-and-middle-market-private-equity-investments/, July 2024.

Yifu Jiang, Jose Olmo, and Majed Atwi. Deep reinforcement learning for portfolio selection. *Global Finance Journal*, 62:101016, September 2024. ISSN 1044-0283.

S. Jones, D. Johnstone, and R. Wilson. Predicting corporate bankruptcy: An evaluation of alternative statistical frameworks. *Journal of Business Finance & Accounting*, 44 (1-2):3–34, 2017. ISSN 0306-686x.

S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang. A comparative study on Transformer vs RNN in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 449–456, December 2019.

Hong Sik Kim and So Young Sohn. Support vector machines for default prediction of SMEs based on technology credit. *European Journal of Operational Research*, 201(3): 838–846, March 2010. ISSN 0377-2217.

Hyeongjun Kim, Hoon Cho, and Doojin Ryu. Corporate bankruptcy prediction using machine learning methodologies with a focus on sequential data. *Computational Economics*, 59:1–19, 2021.

In Joon Kim, Krishna Ramaswamy, and Suresh Sundaresan. Does default risk in coupons affect the valuation of corporate bonds?: A contingent claims model. *Financial Management*, 22(3):117–131, 1993. ISSN 0046-3892.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, February 2017.

Kamesh Korangi, Christophe Mues, and Cristián Bravo. A transformer-based model for default prediction in mid-cap corporate markets. *European Journal of Operational Research*, 308(1):306–320, October 2022. ISSN 0377-2217.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

S. M. Lakew, M. Cettolo, and M. Federico. A comparison of Transformer and Recurrent Neural Networks on multilingual neural machine translation. In *27th International Conference on Computational Linguistics (COLING)*, pages 641–652, 2018.

William R. Lane, Stephen W. Looney, and James W. Wansley. An application of the cox proportional hazards model to bank failure. *Journal of Banking & Finance*, 10(4): 511–531, December 1986. ISSN 0378-4266.

Herwig Langohr and Patricia Langohr. *The Rating Agencies and Their Credit Ratings: What They Are, How They Work, and Why They Are Relevant*. John Wiley & Sons, April 2010. ISBN 978-0-470-71435-5.

Daniela Lazo, Raffaella Calabrese, and Cristian Bravo Roman. The Effects of Customer Segmentation, Borrowers' Behaviours and Analytical Methods on the Performance of Credit Scoring Models in the Agribusiness Sector. *The Journal of Credit Risk*, 16(4): 119–156, January 2021.

Y. Le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard. Handwritten digit recognition: Applications of neural net chips and automatic learning. In Françoise Fogelman Soulié and Jeanny Hérault, editors, *Neurocomputing*, pages 303–318, Berlin, Heidelberg, 1990. Springer. ISBN 978-3-642-76153-9.

Colin Lea, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal Convolutional Networks for action segmentation and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 1558-2256.

Jong Wook Lee, Won Kyung Lee, and So Young Sohn. Graph convolutional network-based credit default prediction utilizing three types of virtual distances among borrowers. *Expert Systems with Applications*, 168:114411, April 2021. ISSN 0957-4174.

Neil Lee, Hiba Sameen, and Marc Cowling. Access to finance for innovative SMEs since the financial crisis. *Research Policy*, 44(2):370–380, March 2015. ISSN 0048-7333.

Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, and Lyn C. Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136, November 2015. ISSN 0377-2217.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a.

Yan Li, Xiong-Fei Jiang, Yue Tian, Sai-Ping Li, and Bo Zheng. Portfolio optimization based on network topology. *Physica A: Statistical Mechanics and its Applications*, 515: 671–681, February 2019b. ISSN 0378-4371.

Zihao Li, Yakun Chen, Xianzhi Wang, Lina Yao, and Guandong Xu. Multi-view GCN for loan default risk prediction. *Neural Computing and Applications*, 36(20): 12149–12162, July 2024. ISSN 1433-3058.

Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461: 370–403, October 2021. ISSN 0925-2312.

H. Lin, J. B. Wang, and C. C. Wu. Liquidity risk and expected corporate bond returns. *Journal of Financial Economics*, 99(3):628–650, 2011.

Hao Liu, Wilson Yan, and Pieter Abbeel. Language Quantized AutoEncoders: Towards Unsupervised Text-Image Alignment. *Advances in Neural Information Processing Systems*, 36:4382–4395, December 2023.

Liang-Chih Liu, Tian-Shyr Dai, and Chuan-Ju Wang. Evaluating corporate bonds and analyzing claim holders' decisions with complex debt structure. *Journal of Banking & Finance*, 72:151–174, November 2016. ISSN 0378-4266.

Andrew W. Lo. The Statistics of Sharpe Ratios. *Financial Analysts Journal*, 58(4):36–52, July 2002. ISSN 0015-198X.

Francis A. Longstaff and Eduardo S. Schwartz. A Simple Approach to Valuing Risky Fixed and Floating Rate Debt. *The Journal of Finance*, 50(3):789–819, 1995. ISSN 0022-1082.

Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, and Guangquan Zhang. Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems*, 80:14–23, 2015.

Lerong Lu. Promoting SME finance in the context of the fintech revolution: A case study of the UK's practice and regulation, March 2018.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

F. Mai, S. N. Tian, C. Lee, and L. Ma. Deep learning models for bankruptcy prediction using textual disclosures. *European Journal of Operational Research*, 274(2):743–758, 2019.

R. N. Mantegna. Hierarchical structure in financial markets. *The European Physical Journal B - Condensed Matter and Complex Systems*, 11(1):193–197, September 1999. ISSN 1434-6036.

Harry Markowitz. Portfolio Selection. *The Journal of Finance*, 7(1):77–91, 1952. ISSN 0022-1082.

Harry M. Markowitz. *Portfolio Selection: Efficient Diversification of Investments*. Yale University Press, 1959. ISBN 978-0-300-01372-6.

Gautier Marti, Frank Nielsen, Mikołaj Bińkowski, and Philippe Donnat. A Review of Two Decades of Correlations, Hierarchies, Networks and Clustering in Financial Markets. In Frank Nielsen, editor, *Progress in Information Geometry: Theory and Applications*, pages 245–274. Springer International Publishing, Cham, 2021. ISBN 978-3-030-65459-7.

J. Spencer Martin and Anthony M. Santomero. Investment opportunities and corporate demand for lines of credit. *Journal of Banking & Finance*, 21(10):1331–1350, October 1997. ISSN 0378-4266.

Robert Andrew Martin. PyPortfolioOpt: Portfolio optimization in Python. *Journal of Open Source Software*, 6(61):3066, May 2021. ISSN 2475-9066.

Andre Martins and Ramon Astudillo. From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1614–1623. PMLR, June 2016.

Guido Previde Massara, T. Di Matteo, and Tomaso Aste. Network Filtering for Big Data: Triangulated Maximally Filtered Graph. *Journal of Complex Networks*, 5(2): 161–178, June 2017. ISSN 2051-1310.

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, December 1943. ISSN 1522-9602.

Carlos F. O. Mendes and Marcus W. Beims. Distance correlation detecting Lyapunov instabilities, noise-induced escape times and mixing. *Physica a-Statistical Mechanics and Its Applications*, 512:721–730, December 2018. ISSN 0378-4371.

Andro Mercep, Lovre Mrcela, Matija Birov, and Zvonko Kostanjcar. Deep neural networks for behavioral credit rating. *Entropy*, 23(1):27, January 2021.

R. C. Merton. Pricing of Corporate Debt - Risk Structure of Interest Rates. *Journal of Finance*, 29(2):449–470, 1974.

Maricica Moscalu, Claudia Girardone, and Raffaella Calabrese. SMEs' growth under financing constraints and banking markets integration in the euro area. *Journal of Small Business Management*, 58(4):707–746, July 2020. ISSN 0047-2778.

S Mostafa Mousavi, William L Ellsworth, Weiqiang Zhu, Lindsay Y Chuang, and Gregory C Beroza. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature communications*, 11(1): 1–12, 2020.

David Munro. *A Guide to SME Financing*. Palgrave Macmillan US, New York, 2013. ISBN 978-1-349-47742-5 978-1-137-37378-6.

Sacha D Nandlall and Koreen Millard. Quantifying the relative importance of variables and groups of variables in remote sensing classifiers using shapley values and game theory. *IEEE Geoscience and Remote Sensing Letters*, 17(1):42–46, 2019.

Peter Naudé, Ghasem Zaefarian, Zhaleh Najafi Tavani, Saeed Neghabi, and Reze Zaefarian. The influence of network effects on SME performance. *Industrial Marketing Management*, 43(4):630–641, May 2014. ISSN 0019-8501.

Noella Nazareth and Yeruva Venkata Ramana Reddy. Financial applications of machine learning: A literature review. *Expert Systems with Applications*, 219:119640, June 2023. ISSN 0957-4174.

David M. Q. Nelson, Adriano C. M. Pereira, and Renato A. de Oliveira. Stock market's price movement prediction with LSTM neural networks. In *International Joint Conference on Neural Networks*, pages 1419–1426. IEEE, 2017.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Ng. Multimodal deep learning. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 689–696, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

Didrik Nielsen. *Tree Boosting with Xgboost-Why Does Xgboost Win" Every" Machine Learning Competition?* PhD thesis, NTNU, 2016.

Mark Nixon and Alberto Aguado. *Feature Extraction and Image Processing for Computer Vision*. Academic Press, November 2019. ISBN 978-0-12-814977-5.

OECD. Financing SMEs and Entrepreneurs 2024. Technical report, OECD Publishing, Paris, March 2024.

James A. Ohlson. Financial Ratios and the Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research*, 18(1):109–131, 1980. ISSN 0021-8456.

J.-P. Onnela, A. Chakraborti, K. Kaski, J. Kertész, and A. Kanto. Dynamics of market correlations: Taxonomy and portfolio analysis. *Physical Review E*, 68(5):056110, November 2003.

María Óskarsdóttir and Cristián Bravo. Multilayer network analysis for improved credit risk prediction. *Omega*, 105:102520, December 2021. ISSN 0305-0483.

John Vincent Owens. Alternative data transforming SME finance. *World Bank Group*, June 2017.

Maciej Pawłowski, Anna Wróblewska, and Sylwia Sysko-Romańczuk. Effective Techniques for Multimodal Data Fusion: A Comparative Analysis. *Sensors (Basel, Switzerland)*, 23(5):2381, February 2023. ISSN 1424-8220.

Charlotte Pelletier, Geoffrey I. Webb, and François Petitjean. Temporal Convolutional Neural Network for the classification of satellite image time series. *Remote Sensing*, 11(5):523, January 2019.

Hao Peng, Hongfei Wang, Bowen Du, Md Zakirul Alam Bhuiyan, Hongyuan Ma, Jianwei Liu, Lihong Wang, Zeyu Yang, Linfeng Du, Senzhang Wang, and Philip S. Yu. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences*, 521:277–290, June 2020. ISSN 0020-0255.

Andre F. Perold. Large-scale portfolio optimization. *Management Science*, 30(10): 1143–1160, October 1984. ISSN 0025-1909.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710, New York, NY, USA, August 2014. Association for Computing Machinery. ISBN 978-1-4503-2956-9.

Giovanni Petrella. Are Euro Area Small Cap Stocks an Asset Class? Evidence from Mean-Variance Spanning Tests. *European Financial Management*, 11(2):229–253, 2005. ISSN 1468-036X.

Lorena Poenaru-Olaru, Judith Redi, Arthur Hovanesyan, and Huijuan Wang. Default prediction using network based features. In Rosa Maria Benito, Chantal Cherifi, Hocine Cherifi, Esteban Moro, Luis M. Rocha, and Marta Sales-Pardo, editors, *Complex Networks & Their Applications X*, pages 732–743, Cham, 2022. Springer International Publishing. ISBN 978-3-030-93409-5.

F. Pozzi, T. Di Matteo, and T. Aste. Spread of risk across financial markets: Better to invest in the peripheries. *Scientific Reports*, 3(1):1665, April 2013. ISSN 2045-2322.

Justo Puerto, Moisés Rodríguez-Madrena, and Andrea Scozzari. Clustering and portfolio selection problems: A unified framework. *Computers and Operations Research*, May 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, October 2019.

Purnima Rao, Satish Kumar, Meena Chavan, and Weng Marc Lim. A systematic literature review on SME financing: Trends and future directions. *Journal of Small Business Management*, 0(0):1–31, August 2021. ISSN 0047-2778.

J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Akos Rona-Tas and Stefanie Hiss. *The Role of Ratings in the Subprime Mortgage Crisis: The Art of Corporate and the Science of Consumer Credit Rating*. Emerald Group Publishing, 2010.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. ISSN 1939-1471.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986. ISSN 1476-4687.

Emrullah Sahin, Naciye Nur Arslan, and Durmus Ozdemir. Unlocking the black box: An in-depth review on interpretability, explainability, and reliability in deep learning. *Neural Computing and Applications*, November 2024. ISSN 1433-3058.

Gaurav Sahu and Olga Vechtomova. Adaptive Fusion Techniques for Multimodal Data, January 2021.

Bahati Sanga and Meshach Aziakpono. FinTech and SMEs financing: A systematic literature review and bibliometric analysis. *Digital Business*, 3(2):100067, December 2023. ISSN 2666-9544.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, January 2009. ISSN 1941-0093.

Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2 (28):307–317, 1953.

William F. Sharpe. Mutual Fund Performance. *The Journal of Business*, 39(1):119–138, 1966. ISSN 0021-9398.

Si Shi, Jianjun Li, Guohui Li, Peng Pan, Qi Chen, and Qing Sun. GPM: A graph convolutional network based reinforcement learning framework for portfolio management. *Neurocomputing*, 498:14–27, August 2022a. ISSN 0925-2312.

Si Shi, Rita Tse, Wuman Luo, Stefano D'Addona, and Giovanni Pau. Machine learning-driven credit risk: A systemic review. *Neural Computing and Applications*, 34 (17):14327–14339, September 2022b. ISSN 1433-3058.

Richard L. Shockley and Anjan V. Thakor. Bank loan commitment contracts: Data, theory, and tests. *Journal of Money, Credit and Banking*, 29(4):517–534, 1997. ISSN 0022-2879.

Tyler Shumway. Forecasting bankruptcy more accurately: A simple hazard model. *The Journal of Business*, 1:101–124, 2001.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, May 2022. ISSN 1566-2535.

Andrew F. Siegel and Artemiza Woodgate. Performance of Portfolios Optimized with Estimation Error. *Management Science*, 53(6):1005–1015, June 2007. ISSN 0025-1909.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Enhancing AI Systems with Agentic Workflows Patterns in Large Language Model. In *2024 IEEE World AI IoT Congress (AIIoT)*, pages 527–532, May 2024.

So Young Sohn, Kyong Taek Lim, and Yonghan Ju. Optimization strategy of credit line management for credit card business. *Computers & Operations Research*, 48:81–88, August 2014. ISSN 0305-0548.

Hua Song, Kangkang Yu, Anirban Ganguly, and Rabia Turson. Supply chain network, information sharing and SME credit quality. *Industrial Management & Data Systems*, 116(4):740–758, January 2016. ISSN 0263-5577.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014. ISSN 1532-4435.

Nitish Srivastava and Russ R Salakhutdinov. Multimodal Learning with Deep Boltzmann Machines. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Matthew Stevenson, Christophe Mues, and Cristián Bravo. The value of text for small business default prediction: A Deep Learning approach. *European Journal of Operational Research*, 295(2):758–771, 2021.

Amir Sufi. Bank lines of credit in corporate finance: An empirical analysis. *Review of Financial Studies*, 22(3):1057–1088, March 2009. ISSN 0893-9454.

Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4597–4605, 2015.

Ruili Sun, Tiefeng Ma, and Shuangzhe Liu. Portfolio selection based on semivariance and distance correlation under minimum variance framework. *Statistica Neerlandica*, 73(3):373–394, 2019. ISSN 1467-9574.

Ting Sun and Miklos A. Vasarhelyi. Predicting credit card delinquencies: An application of deep neural networks. *Intelligent Systems in Accounting, Finance and Management*, 25(4):174–189, 2018. ISSN 1099-1174.

Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, December 2007. ISSN 0090-5364, 2168-8966.

Kar Yan Tam and Melody Y. Kiang. Managerial applications of neural networks: The case of bank failure rredictions. *Management Science*, 38(7):926–947, July 1992. ISSN 0025-1909.

Mark Thackham and Jun Ma. Exposure at default without conversion factors-evidence from Global Credit Data for large corporate revolving facilities. *Journal of the Royal Statistical Society Series a-Statistics in Society*, 182(4):1267–1286, October 2019. ISSN 0964-1998.

Edward N. C. Tong, Christophe Mues, Iain Brown, and Lyn C. Thomas. Exposure at default models with and without the credit conversion factor. *European Journal of Operational Research*, 252(3):910–920, August 2016. ISSN 0377-2217.

M. Tumminello, T. Aste, T. Di Matteo, and R. N. Mantegna. A tool for filtering information in complex systems. *Proceedings of the National Academy of Sciences*, 102 (30):10421–10426, July 2005.

Daniel Vale, Ali El-Sharif, and Muhammed Ali. Explainable artificial intelligence (XAI) post-hoc explainability methods: Risks and limitations in non-discrimination law. *AI and Ethics*, 2(4):815–826, November 2022. ISSN 2730-5961.

Oldrich Alfons Vasicek. Credit Valuation. Technical report, KMV Corporation, 1984.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, February 2018.

Davide Venturelli and Alexei Kondratyev. Reverse quantum annealing approach to portfolio optimization problems. *Quantum Machine Intelligence*, 1(1):17–30, May 2019. ISSN 2524-4914.

Veronica Vinciotti, Elisa Tosetti, Francesco Moscone, and Mark Lycett. The effect of interfirm financial transactions on the credit risk of small and medium-sized enterprises. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182(4): 1205–1226, 2019. ISSN 1467-985X.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, 2016.

William Watson, Nicole Cho, Nishan Srishankar, Zhen Zeng, Lucas Cecchi, Daniel Scott, Suchetha Siddagangappa, Rachneet Kaur, Tucker Balch, and Manuela Veloso. LAW: Legal Agentic Workflows for Custody and Fund Services Contracts, December 2024.

Suttisak Wattanawongwan, Christophe Mues, Ramin Okhrati, Taufiq Choudhry, and Mee Chi So. A mixture model for credit card exposure at default using the GAMLSS framework. *International Journal of Forecasting*, 39(1):503–518, January 2023. ISSN 0169-2070.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in Time Series: A Survey, February 2022.

Lawrence J. White. Credit Rating Agencies: An Overview. *Annual Review of Financial Economics*, 5(Volume 5, 2013):93–122, November 2013. ISSN 1941-1367, 1941-1375.

Sarah Wiegreffe and Yuval Pinter. Attention is not not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 11–20, November 2019.

Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. Quant GANs: Deep generation of financial time series. *Quantitative Finance*, 20(9):1419–1440, September 2020. ISSN 1469-7688.

Kieran Wood, Stephen Roberts, and Stefan Zohren. Slow Momentum with Fast Reversion: A Trading Strategy Using Deep Learning and Changepoint Detection. *The Journal of Financial Data Science*, 4(1):111–129, January 2022. ISSN 2640-3943.

Jun Wu, XiongFei Zhao, Hang Yuan, and Yain-Whar Si. CDGAT: A graph attention network method for credit card defaulters prediction. *Applied Intelligence*, 53(10): 11538–11552, May 2023. ISSN 1573-7497.

Sifan Wu, Xi Xiao, Qianggang Ding, Peilin Zhao, Ying Wei, and Junzhou Huang. Adversarial sparse transformer for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 33, pages 17105–17115. Curran Associates, Inc., 2020.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, January 2021. ISSN 2162-2388.

Yu Xie, Maoguo Gong, Shanfeng Wang, and Bin Yu. Community discovery in networks with deep sparse filtering. *Pattern Recognition*, 81:50–59, September 2018. ISSN 0031-3203.

Bingbing Xu, Huawei Shen, Bingjie Sun, Rong An, Qi Cao, and Xueqi Cheng. Towards Consumer Loan Fraud Detection: Graph Neural Networks with Role-Constrained Conditional Random Field. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4537–4545, May 2021. ISSN 2374-3468.

Peng Xu, Xiatian Zhu, and David A. Clifton. Multimodal Learning With Transformers: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10): 12113–12132, October 2023. ISSN 1939-3539.

Xin-Guo Yan, Chi Xie, and Gang-Jin Wang. Stock market network's topological stability: Evidence from planar maximally filtered graph and minimal spanning tree. *International Journal of Modern Physics B*, 29(22):1550161, September 2015. ISSN 0217-9792.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *KDD'18: PROCEEDINGS OF THE 24TH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY & DATA MINING*, pages 974–983, New York, 2018. Assoc Computing Machinery. ISBN 978-1-4503-5552-0.

Sahab Zandi, Kamesh Korangi, María Óskarsdóttir, Christophe Mues, and Cristián Bravo. Attention-based dynamic multilayer graph neural networks for loan default prediction, June 2024.

Dengjun Zhang and Geir Sogn-Grundvåg. Credit constraints and the severity of COVID-19 impact: Empirical evidence from enterprise surveys. *Economic Analysis and Policy*, 74:337–349, June 2022. ISSN 0313-5926.

Guoqiang Zhang, Michael Y. Hu, B. Eddy Patuwo, and Daniel C. Indro. Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. *European Journal of Operational Research*, 116(1):16–32, 1999. ISSN 0377-2217.

Muhan Zhang and Yixin Chen. Link Prediction Based on Graph Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep Learning for Portfolio Optimization. *The Journal of Financial Data Science*, 2(4):8–20, October 2020. ISSN 2640-3943, 2640-3951.

Yaochen Zhu, Zhenzhong Chen, and Feng Wu. Multimodal Deep Denoise Framework for Affective Video Content Analysis. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, pages 130–138, New York, NY, USA, October 2019. Association for Computing Machinery. ISBN 978-1-4503-6889-6.

Maciej Zięba, Sebastian K. Tomczak, and Jakub M. Tomczak. Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems with Applications*, 58:93–101, 2016.