



# Credit Rating Prediction using Multi-modal Temporal Neural Networks

by

Magnús Freyr Morthens

Thesis of 60 ECTS credits submitted to the School of Technology,  
Department of Computer Science  
at Reykjavík University in partial fulfillment  
of the requirements for the degree of  
**Master of Science (M.Sc.) in Artificial Intelligence and  
Language Technologies**

June 2024

Examining Committee:

María Óskarsdóttir, Supervisor  
Associate Professor, Reykjavík University, Iceland

Cristián Bravo, Co-supervisor  
Associate Professor, Western University, Canada

Stephan Schiffel, Examiner  
Assistant Professor, Reykjavík University, Iceland

Stefán Ólafsson, Examiner  
Assistant Professor, Reykjavík University, Iceland

Copyright  
Magnús Freyr Morthens  
June 2024

# Credit Rating Prediction using Multi-modal Temporal Neural Networks

Magnús Freyr Morthens

June 2024

## Abstract

This research explores the integration of Recurrent Neural Networks (RNNs) and Graph Neural Networks (GNNs) to enhance corporate credit rating prediction. Corporate credit risk, characterized by the dynamic financial interactions and dependencies among firms, is effectively modeled using a multi-modal approach that merges graph-based and sequential data processing. This research employs Graph Convolutional Networks (GCNs) and various forms of RNNs, including Long Short-Term Memory (LSTM) units, Gated Recurrent Units (GRUs), and Transformers, to analyze corporate financial health both statically and dynamically. The models are trained using data from 318 US companies, synthesizing numerical data from quarterly financial statements and graph data constructed from stock market interactions to capture intricate patterns of corporate financial activities. The effectiveness of these multi-modal models is measured on both the 318 companies it was trained on, in a transductive way, as well as inductively 112 "unseen" companies that were excluded during model training. The superior performance in both seen and unseen data scenarios demonstrates robust generalization capabilities, essential for practical deployment in financial risk assessment.

**Keywords:** Credit rating, Multi-modal, Corporate

# Lánshæfisspá með Fjölpáttu Tímabundnum Tauganetum

Magnús Freyr Morthens

júní 2024

## Útdráttur

Þessi rannsókn kannar samþættingu endurtekinna tauganeta (RNN) og neta tauganeta (GNN) til að bæta spá um lánshæfismat 318 ameríkska fyrirtækja. Lánshæfisspáslíkan fyrir fyrirtæki er útfært með samþættingu RNN og GNN líkönnum sem sameinar gagnastruma frá bæði netum og tölulegum gögnum. Með því að sameina töluleg gögn úr ársfjórðungsreikningum og netagögn sem eru byggð á hlutabréfamarkaðsviðskiptum, ná líkön okkar að fanga flókin mynstur í fjármálastarfsemi fyrirtækja. Ýmsar samsetningar af RNN og GNN, eins og LSTM, GRU, Transformers og GCN, eru sameinaðar, niðurstöður bornar saman og metnar. Til þess að meta styrkleika og hagnýta notkun þessara fjölpáttalíkana, eru niðurstöður metnar bæði fyrir þessi 318 fyrirtæki sem líkönin eru þjálfuð á og einnig fyrir 112 áður óséð fyrirtæki.

**Efnisorð:** Lánshæfisspá, Fjölpáttu tauganet, Fyrirtæki

# Contents

<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related Work</b>	<b>5</b>
2.1 Graph Neural Networks . . . . .	5
2.2 Recurrent Neural Networks . . . . .	6
2.3 Dynamic Graph Neural Networks . . . . .	6
2.4 Multi-modal Deep Learning for Credit Risk Assessment . . . . .	7
<b>3 Methodology</b>	<b>9</b>
3.1 Data Collection and Processing . . . . .	9
3.1.1 Numerical Data . . . . .	9
3.1.2 Graph Data . . . . .	10
3.2 Model Architecture . . . . .	15
3.2.1 RGNN Architecture . . . . .	15
3.2.2 RNN Architecture . . . . .	16
3.2.3 Multi-modal Architecture . . . . .	17
3.3 Loss Function and Optimization Techniques . . . . .	18
3.3.1 Loss Function . . . . .	18
3.3.2 Optimizer . . . . .	19
3.3.3 Adaptive Learning Rate . . . . .	19
3.3.4 Gradient Clipping . . . . .	19
3.4 Evaluation Metrics and Validation . . . . .	20
3.4.1 AUC . . . . .	20
3.4.2 F1 Score . . . . .	20
3.4.3 AUPRC . . . . .	21
3.5 Baseline Models . . . . .	21
<b>4 Experimental setup</b>	<b>23</b>
4.1 Dataset . . . . .	23
4.2 Training Procedure . . . . .	23
4.3 Hyper-parameter Tuning . . . . .	24
4.4 Experiments . . . . .	24
<b>5 Results and Discussion</b>	<b>27</b>

5.1	Baseline Models . . . . .	27
5.1.1	RNN Models, Transformers . . . . .	27
5.1.2	RGNN Models . . . . .	28
5.1.3	XGBoost . . . . .	28
5.2	RNN-RGNN Models . . . . .	28
5.3	Window Size Comparison . . . . .	29
5.4	Performance and Generalization Capabilities . . . . .	30
5.5	Error Analysis . . . . .	31
5.5.1	Accuracy and Misclassifications . . . . .	31
5.5.1.1	Baseline Models . . . . .	43
5.5.1.2	Multi-modal Models . . . . .	43
5.5.2	Performance Over Time . . . . .	44
<b>6</b>	<b>Conclusion</b>	<b>47</b>
<b>Bibliography</b>		<b>49</b>
<b>A</b>	<b>Numerical Features</b>	<b>55</b>

# List of Figures

3.1	Class distribution of the financial statement data. . . . .	10
3.2	Distance correlation matrix for the first 10 companies in the dataset. . . . .	12
3.3	Complete graph showing the connections between the companies in the first quarter of 2010 pre-TMFG application. . . . .	13
3.4	Image showing the first two iterations of TMFG triangulation process. . .	14
3.5	Graph showing the connections between the companies in the first quarter of 2010 post-TMFG application. . . . .	15
3.6	An overview of the RNN-RGNN multi-modal architecture. . . . .	18
4.1	An example of one observation window in the rolling window approach. .	24
5.1	Comparison of model performances across different observation window sizes.	30
5.2	a) Train/Validation loss. b) AUC convergence plot. c) AUPRC convergence plot. . . . .	30
5.3	Confusion Matrix for the GCNGRU model. . . . .	32
5.4	Confusion Matrix for the GCNLSTM model. . . . .	33
5.5	Confusion Matrix for the LSTM model. . . . .	34
5.6	Confusion Matrix for the GRU model. . . . .	35
5.7	Confusion Matrix for the Transformers model. . . . .	36
5.8	Confusion Matrix for the XGBoost model. . . . .	37
5.9	Confusion Matrix for the GRU-GCNLSTM model. . . . .	38
5.10	Confusion Matrix for the TRA-GCNGRU model. . . . .	39
5.11	Confusion Matrix for the LSTM-GCNGRU model. . . . .	40
5.12	Confusion Matrix for the GRU-GCNGRU model. . . . .	41
5.13	Confusion Matrix for the LSTM-GCNLSTM model. . . . .	42
5.14	Accuracy and misclassifications over time for the GRU model. . . . .	44
5.15	Accuracy and misclassifications over time for the GRU-GCNGRU model. .	44

# List of Tables

4.1	Hyper-parameter space . . . . .	24
5.1	Performance of Baseline Models on Seen Companies . . . . .	27
5.2	Performance of Baseline Models on Unseen Companies . . . . .	27
5.3	Performance of Top 5 RNN-RGNN Models on Seen Companies . . . . .	29
5.4	Performance of Top 5 RNN-RGNN Models on Unseen Companies . . . . .	29
5.5	Model Prediction Summary . . . . .	43

# Chapter 1

## Introduction

Credit risk, defined as the potential for a borrower to default on their financial obligations, plays a pivotal role in the financial system [1]. It signifies the uncertainty associated with the repayment of both the principal and accrued interest by borrowers. This inherent risk underscores the importance of credit risk ratings, evaluations that gauge the creditworthiness of a variety of borrowers, ranging from individuals and corporations to nations [2]. These evaluations, primarily overseen by credit rating agencies, provide indispensable insight into the borrower's fiscal responsibility. An accurate estimation of credit risk is foundational for maintaining the stability of the financial ecosystem, as evidenced by the catastrophic implications of misjudgments, such as the 2008 sub-prime crisis [3]. To preemptively mitigate such outcomes, lending institutions engage in meticulous scrutiny of borrowers, particularly corporations, given the intricate nature of their operations.

Corporate credit risk evaluations encompass a wide range of critical determinants. Foremost among these are the financial statements, like income sheets and balance statements, which offer a snapshot of the corporation's fiscal health. Coupled with these are financial ratios that illuminate aspects like liquidity and operational efficacy [2]. Beyond these numerical indicators, lenders take stock of qualitative attributes, such as a company's strategic positioning, growth trajectories, and managerial competence. External elements, ranging from prevailing economic conditions to the company's alignment with regulators and its rapport with lenders, also significantly influence lending decisions. As a part of ongoing risk management, future financial forecasts and loan covenant adherence are continually under scrutiny [1].

The aftermath of the 2008 crisis accentuated the urgency for refined credit risk predictive models [4]. Overestimating credit ratings poses several risks for lenders, including increased default risk, financial losses, underpricing of risk, inefficient capital allocation, and regulatory and compliance issues. Conversely, underestimating credit ratings can lead to lost business opportunities, higher interest rates for low-risk borrowers, reputational damage, inefficient use of capital, and an imbalanced loan portfolio. Accurate credit rating assessments are crucial for lenders to manage these risks effectively, ensuring reliable credit decisions that contribute to financial stability and profitability. In response, researchers and fiscal specialists embarked on extensive explorations to diversify methodologies and augment model features, striving for optimal accuracy [5]–[7]. While traditional approaches relied on statistical analyses and manual oversight, the dawning of financial artificial intelligence (AI) heralded a shift towards machine learning (ML) paradigms in credit risk modeling. A recent article [8], presents a critical analysis of the current regulatory landscape for credit risk

management, particularly in the context of the increasing use of ML and AI models. The paper highlights the growing divergence in regulatory approaches across different regions and the urgency for a unified regulatory framework. It discusses the Basel Regulations as a comprehensive global framework that has historically guided credit risk management and how the introduction of AI and ML models is reshaping this landscape. ML models have exhibited superior efficacy compared to their traditional counterparts, credited to their adaptive weight tuning and nuanced data classification capabilities [9].

In the modern era of data-driven decision-making, the adoption of multi-modal models are gaining momentum, particularly due to their ability to integrate diverse data streams for comprehensive analysis [10]. The intricate nature of corporate operations strongly justifies the integration of multiple data sources for predicting corporate credit scores. By merging data across diverse domains, multi-modal models offer a wide-ranging perspective [11]. This all-encompassing approach unravels complex patterns and interdependencies that single-source models might miss, leading to a more thorough assessment of credit risk [12]. Such a holistic strategy not only improves prediction accuracy but also strengthens the resilience of these models, thereby enhancing the quality of decision-making in credit risk management.

Recurrent Neural Networks (RNNs) are foundational to the analysis of sequential data [13] and have demonstrated effective results in capturing the temporal dynamics inherent in financial time series [14]. Their architecture is designed to handle sequences by maintaining a hidden state that effectively captures information from previous time steps. This characteristic makes RNNs particularly valuable for tasks such as predicting corporate credit scores, where past financial performance can provide crucial insights into future fiscal health. By processing data through time, RNNs allow for the integration of historical trends, seasonal variations, and sudden shifts in economic conditions, enabling a nuanced and predictive analysis of credit risk. Models like Long Short-Term Memory (LSTM) [15] units and Gated Recurrent Units (GRUs) [16] enhance this capability further by addressing the challenge of long-term dependencies, where significant events from the distant past influence current predictions. These models are adept at discerning which historical data points are relevant for making accurate predictions, thus improving the reliability and robustness of credit risk assessment.

Graph Neural Networks (GNNs) have been increasingly applied to a variety of tasks and problems, particularly those involving data with inherent graph structures or relational contexts [17]. The application of GNNs is especially pertinent in corporate credit risk assessment, where credit risk is not only correlated but also evolves over time. Given that corporations exist within interwoven networks comprising stakeholders like suppliers, clients, and competitors, their creditworthiness is inherently intertwined and dynamically influenced by these relationships. For instance, financial instability of a vital supplier can have cascading effects on dependent entities. In such a scenario, the use of Dynamic Graph Neural Networks (DGNNS) becomes crucial as they are equipped to handle the time-evolving nature of credit risk, providing a means to track and predict changes in creditworthiness over time. DGNNS adeptly navigate these convoluted interrelations, offering enriched credit risk analyses that account for both the static and dynamic aspects of corporate financial networks.

Using RNNs and GNNs, we delve into the realm of multi-modal deep learning models, focusing specifically on their application in the prediction of corporate credit risk. We integrate numerical data such as market indicators, financial ratios, and covariate

information, which provide traditional quantitative measures of a company’s performance and stability. Alongside these tabular data sources, we also incorporate graphs derived from stock data. These graphs are constructed using a novel method based on similarity of market price volatility and thus captures the complex relationships and dependencies between different corporate entities, which are often missed in traditional numeric analyses. Moreover, the temporal aspect of the data streams is crucial, as both the graph representations and the tabular data for the companies dynamically evolve, reflecting the changing conditions and interactions within the market. By merging these diverse data streams, our multi-modal framework aims to provide a more nuanced and accurate prediction of corporate credit risk, harnessing the strengths of both numeric and graph-based data and their dynamics. We evaluate our models in both an inductive way on companies seen during model training, and a transductive manner on companies that are not. Our models show superior performance across a range of both graph based, recurrent and baseline models, especially for the unseen nodes. This indicates that a holistic multi-modal architecture capturing the dynamics of both numeric and financial graph data is necessary to accurately predict the credit rating of new companies in the portfolio.

In this research, we explore the potential of multi-modal temporal neural networks to enhance credit rating prediction. Specifically, we aim to address two primary research questions: First, can multi-modal temporal neural networks, using both tabular numerical data and graph data, enhance credit rating prediction compared to unimodal models? Second, what contributions does each modality provide in the context of credit rating prediction? By investigating these questions, we seek to demonstrate the value of integrating diverse data types and uncover the unique advantages offered by each modality in improving predictive accuracy and reliability.



# Chapter 2

## Related Work

### 2.1 Graph Neural Networks

GNNs have gained substantial traction in recent years, owing to their ability to efficiently process and analyze graph-structured data [17]. This data structure, comprised of nodes and edges, effectively captures the rich relational information among elements, making GNNs particularly adept at a wide range of applications. These applications span various domains, including social networks [18], physical systems [19], [20], protein networks [21], and knowledge graphs [22].

A pivotal aspect of GNNs is their ability to capture dependencies within graphs through message passing between nodes. This feature has led to groundbreaking performances in many deep learning tasks [17]. Variants of GNNs, such as Graph Convolutional Networks (GCNs) [23] and Graph Attention Networks (GATs) [24], have demonstrated remarkable effectiveness in handling graph data, which is inherently non-Euclidean. This non-Euclidean nature poses unique challenges and opportunities, leading to the emergence of geometric deep learning as a significant research area.

Graph representation learning [25]–[28], another cornerstone of GNNs, focuses on encoding graph nodes, edges, or subgraphs into low-dimensional vectors. This approach, evolving from the traditional machine learning methods which often relied on hand-engineered features, has shown breakthroughs in graph analysis [17]. Following the success of graph representation learning and word embedding [29], techniques like DeepWalk [30], node2vec [31], LINE [32], and TADW [33], have been pivotal in advancing graph-based machine learning by introducing graph embedding. However, these models have shown limitations in terms of flexibility and computational efficiency [34].

Moreover, GNNs have been applied in specialized graph learning fields, including adversarial learning methods on graphs, graph attention models, and heterogeneous graph representation learning. Studies have also focused on dynamic graphs and graph embeddings methods for combinatorial optimization, demonstrating the versatility of GNNs across various graph types and applications [35].

GNNs have been categorized into recurrent, convolutional, autoencoder, and spatial-temporal networks, each addressing specific aspects of graph data processing and analysis [36]. Comprehensive reviews on GNNs highlight the diversity and evolution of these models. For instance, Bronstein et al. [37] offer an extensive overview of geometric deep learning, encompassing its challenges and applications. Similarly, Zhang et al. [38] and others provide detailed surveys of GCNs, focusing on different computational modules such as skip connections and pooling operators [39], [40].

## 2.2 Recurrent Neural Networks

RNNs have become instrumental in handling sequential data, making them a fundamental tool in various domains, particularly in the analysis of time-series information. Their ability to maintain a memory of previous inputs using their internal states allows RNNs to exhibit temporal dynamic behavior. This capability is crucial for applications where the sequence of data points is essential, such as in speech recognition [41], language modeling [42], and notably, in financial time-series analysis for credit risk assessment [43].

In the field of credit risk assessment, RNNs have shown promising results by effectively capturing the temporal dependencies of financial indicators [12], [43]–[48]. Financial data, inherently sequential and time-dependent, often consists of patterns that unfold over time, dictating the financial health and stability of an entity. RNNs are adept at learning these patterns, providing a dynamic perspective that traditional static models may overlook. For instance, research has demonstrated that RNNs, particularly those using LSTM units or GRUs, significantly improve the prediction accuracy of credit default [43], [46]. These models are capable of learning from extensive historical financial data, recognizing long-term dependencies that are critical for anticipating future credit events. Such capabilities are essential for developing robust predictive models that financial institutions can rely on for making informed lending decisions.

A study by Ala’raj et al. [47] employed LSTM networks to predict the probability of single and consecutive missed payments of credit card users by analyzing their transaction and payment patterns over time. The LSTM model outperformed traditional statistical models by adapting to the nuances of individual financial behavior. Another significant contribution by Chen et al. [48] developed the SMAGRU model, an innovative end-to-end architecture combining self multi-head attention with GRUs. This model excels in corporate credit rating by effectively harnessing the time-series features of financial data and assigning appropriate weights to market benchmarks. Their results revealed that the SMAGRU model, when applied to 515 American enterprises across 10 credit rating class categories, achieved an AUC score of 0.75.

## 2.3 Dynamic Graph Neural Networks

Dynamic Graph Neural Networks (DGNNs) have emerged as a crucial innovation in encoding dynamic networks, integrating both structural and temporal patterns [49]. Utilizing the foundations of GNNs, DGNNs extend these capabilities by incorporating time-series modules, such as RNNs or positional attention, to effectively process dynamic networks. This advancement allows DGNNs to encode not only the structural attributes of the graph but also the temporal evolution, providing a richer representation of dynamic data.

DGNNs demonstrate versatility in their application, being adaptable to a range of tasks similar to those addressed by GNNs [49]. Common applications include node classification and link prediction, both of which have wide-ranging implications across multiple disciplines, such as knowledge graph completion and recommender systems. Furthermore, DGNNs have been employed in novel tasks like predicting path-failure in dynamic graphs [50], traffic flow forecasting [51], identifying influencers [52] and

detecting social behavioral and characteristic patterns such as dominance, deception, and nervousness [53].

Previous surveys, such as those by Kazemi et al., Xie et al., and Barros et al. [54]–[56], have generally addressed dynamic network representation learning, highlighting the broader scope of the field. However, the specific focus on DGNNs in this context emphasizes their unique contribution to encoding dynamic aspects of networks.

A recent article [57], illustrates the application of DGNNs in the field of financial portfolio optimization. By employing GATs, the study adeptly harnesses the dynamic nature of financial markets, capturing the evolving relationships and dependencies between different assets. This approach allows for a more nuanced and adaptive portfolio optimization strategy, reflective of the real-time changes in market conditions. The utilization of dynamic GNNs in this context not only enhances the accuracy of financial predictions but also demonstrates the versatility and efficacy of these networks in managing complex, time-sensitive financial data.

## 2.4 Multi-modal Deep Learning for Credit Risk Assessment

By leveraging a multi-modal deep learning framework for corporate credit rating prediction, researchers have effectively combined quantitative financial information with qualitative aspects like textual data from financial reports and news articles [12]. The research highlights the effectiveness of combining various data sources, surpassing the limitations of traditional models that rely solely on numerical data, and paves the way for more sophisticated and accurate financial risk evaluation methods.

GNNs combined with multi-modality have been applied on financial time series forecasting [58]. The researchers explored the synergy between GNNs and multi-modal data, leveraging the graph-based representation to efficiently integrate and process information from multiple modalities. The study showed that by using GNNs, their model effectively captured the complex interdependencies and relationships between different data types, whether it be textual, numerical, or visual.

DeepRisk [59], a multi-modal credit risk prediction model for small and medium-sized enterprises (SMEs), innovatively combines static enterprise demographic data with dynamic financing behavioral data, addressing the limitations of traditional models that primarily rely on static data. The research demonstrates that DeepRisk surpasses baseline methods in various metrics, including F1-score and Area Under Curve (AUC), highlighting the effectiveness of integrating multiple data modalities in credit risk prediction.

However, there remains a significant gap in the literature concerning the use of GNNs specifically for corporate credit rating within a multi-modal framework. While GNNs have shown promise in capturing relational and structural data in various applications, their potential in credit rating, especially when combined with other data modalities, has not been extensively explored. The intricate dependencies in financial networks and the dynamic nature of corporate interactions are areas where GNNs could provide substantial insights, yet empirical studies demonstrating their effectiveness in a multi-modal context are sparse. Addressing this gap could lead to more robust and accurate models for credit risk assessment, leveraging the strengths of GNNs in under-

standing complex relationships and temporal dynamics in conjunction with traditional financial indicators and/or textual data.

# Chapter 3

## Methodology

This section explains the methods used in this study to assess and predict corporate credit ratings with advanced machine learning techniques. We used recurrent and graph deep learning models in a multi-modal learning strategy. We describe how we collected and processed data and the design of the models. Our approach combines different data types and utilizes both graph-based and sequential data processing, helping to address the complex task of credit risk assessment. By combining GNNs and RNNs, we aim to capture both static and dynamic features of financial data, offering a solid framework for understanding and predicting corporate creditworthiness.

### 3.1 Data Collection and Processing

#### 3.1.1 Numerical Data

The dataset comprises numerical and categorical attributes extracted from the quarterly financial statements of 318 US companies from 2010 to 2020, encompassing 44 quarters per company. The target label within this dataset is derived from the credit ratings assigned by credit rating agencies, which use a letter-based system ranging from AAA (highest credit quality) to lower qualities and ultimately default. By employing a conversion table, these letter-based ratings are translated into a numerical scale for quantitative analysis, forming 22 distinct classes of rating. Given the disparity in data volume across these classes, those with a frequency below 5% were merged with adjacent classes, resulting in a condensed grouping of 8 classes for classification purposes as in [12]. Figure 3.1 displays the final class distribution of this condensed grouping, providing a visual representation of the data used in our models. Class 0 represents the highest credit rating classes and 7 the lowest.

Initially, the quarterly financial statements dataset comprised 681 features. After preprocessing, we refined the dataset to 201 features by dropping irrelevant features such as city names, addresses, phone numbers, email addresses, URLs, dates, ID columns, features with near-zero variance, and columns with over 50% missing values. To address the challenge of missing values within these selected features, we employed iterative imputing using Bayesian Ridge as the estimator. Iterative imputing is a multivariate imputing method that models each feature with missing values as a function of other features in a round-robin fashion. Studies have shown that using multivariate imputing methods can significantly improve the accuracy of imputation compared to simpler univariate imputation methods [60]–[62]. The Bayesian Ridge estimator, cho-

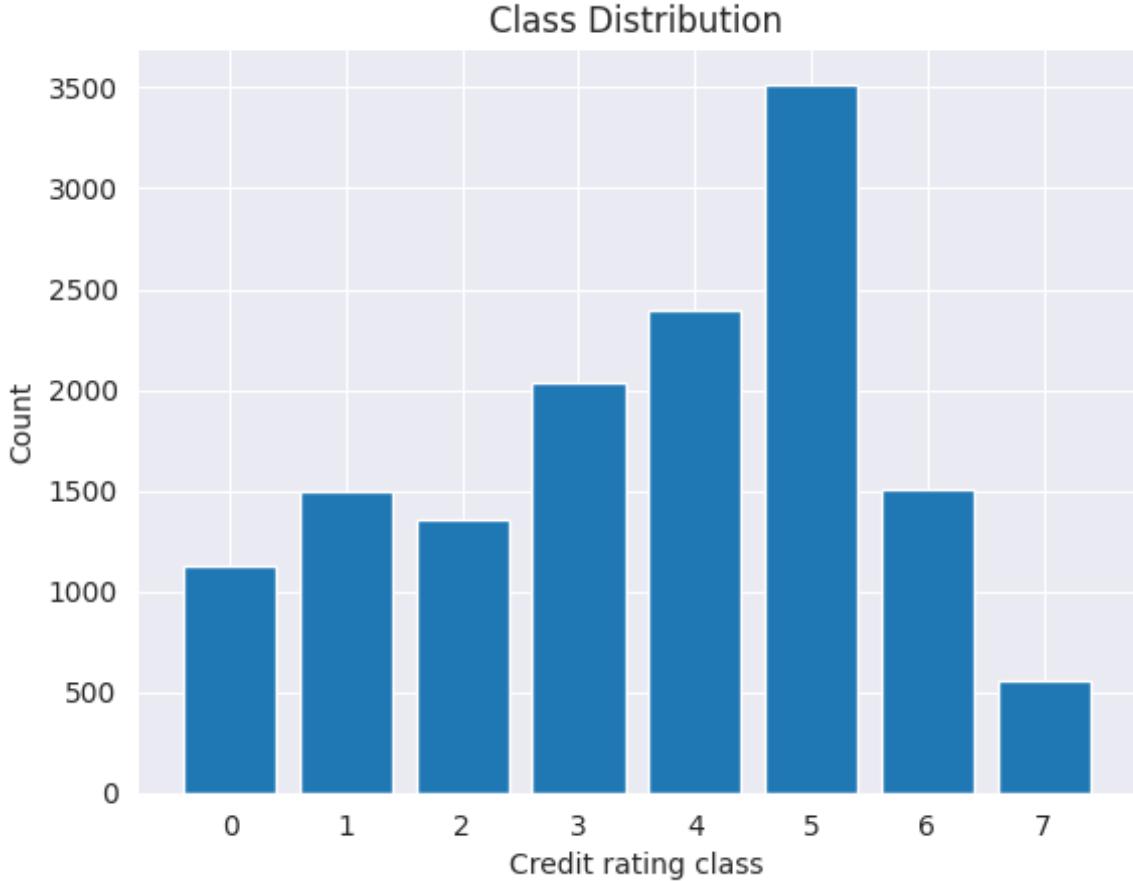


Figure 3.1: Class distribution of the financial statement data.

sen for its efficiency and reliability, combines ridge regression with Bayesian inference to provide a probabilistic approach to estimation. This method imputes missing values effectively and incorporates the uncertainty of the imputations, thereby yielding a more robust and informative dataset for subsequent analysis.

### 3.1.2 Graph Data

We collected the daily closing stock prices of all the 318 companies present in the numerical dataset. To construct graphs with edges representing similarities between the companies, we calculate a correlation between their daily return-volatility series, following Korangi et al. [57]. First, we convert the prices to daily returns by determining the standard deviation of the daily closing stock prices from 2010 to 2020, utilizing a 30-day look-back period. For a company  $u$  at time  $t$ , it is expressed as:

$$c_{ut} = \sigma(r_{ut}, r_{u(t-1)}, \dots, r_{u(t-30)}) \quad (3.1)$$

where  $r_{ut}$  denotes daily returns of firm  $u$  at time  $t$ . Then we define the daily return-volatility series for company  $u$  as:

$$\vec{l}_u = [c_{u1}, c_{u2}, \dots, c_{uT}] \quad (3.2)$$

where  $T$  denotes the length of the look-back period. At any given time,  $T$  is the number of days between 2010 quarter 1 and year  $y$  quarter  $q$  in our numerical dataset.

This daily return-volatility series is key as it helps calculate the covariance matrix between companies. Using the return-volatility series is important as it shows how companies' stocks change in value over time. These changes are important indicators of market perception and financial stability, providing insights into the credit risk of each company.

To quantitatively score the strength of how strongly the volatility of a pair of firms is related in a specific period, we employ the distance correlation measure. Unlike traditional correlation measures that can only capture linear relationships, distance correlation identifies both linear and nonlinear associations between two variables. This feature makes it suitable for financial data, which often exhibit complex, nonlinear interactions influenced by a multitude of factors.

The distance correlation  $dCor(u, v)$  between company  $u$  and  $v$  is calculated as follows:

1. Compute the pairwise Euclidean distances between each point in  $\vec{l}_u$  and  $\vec{l}_v$ , forming distance matrices  $A$  and  $B$ , where  $A_{ij} = \|l_{u,i} - l_{u,j}\|$  and  $B_{ij} = \|l_{v,i} - l_{v,j}\|$ .
2. Center these distance matrices by adjusting each entry as follows:

$$A_{ij}^* = A_{ij} - \bar{A}_{i\cdot} - \bar{A}_{\cdot j} + \bar{A}_{\cdot\cdot}, \quad B_{ij}^* = B_{ij} - \bar{B}_{i\cdot} - \bar{B}_{\cdot j} + \bar{B}_{\cdot\cdot}$$

where  $\bar{A}_{i\cdot}$  and  $\bar{A}_{\cdot j}$  are the means of the  $i$ -th row and  $j$ -th column of  $A$  respectively, and  $\bar{A}_{\cdot\cdot}$  is the overall mean.

3. Calculate the distance covariance ( $dCov$ ) as:

$$dCov(\vec{l}_u, \vec{l}_v) = \sqrt{\frac{1}{n^2} \sum_{i,j} A_{ij}^* B_{ij}^*}$$

where  $n$  is the number of data points in  $\vec{l}_u$  and  $\vec{l}_v$ .

4. Compute the distance variances ( $dVar$ ) for  $\vec{l}_u$  and  $\vec{l}_v$  using:

$$dVar(\vec{l}_u) = \sqrt{\frac{1}{n^2} \sum_{i,j} (A_{ij}^*)^2}, \quad dVar(\vec{l}_v) = \sqrt{\frac{1}{n^2} \sum_{i,j} (B_{ij}^*)^2}$$

5. Finally, derive the distance correlation:

$$dCor(\vec{l}_u, \vec{l}_v) = \frac{dCov(\vec{l}_u, \vec{l}_v)}{\sqrt{dVar(\vec{l}_u) \times dVar(\vec{l}_v)}}$$

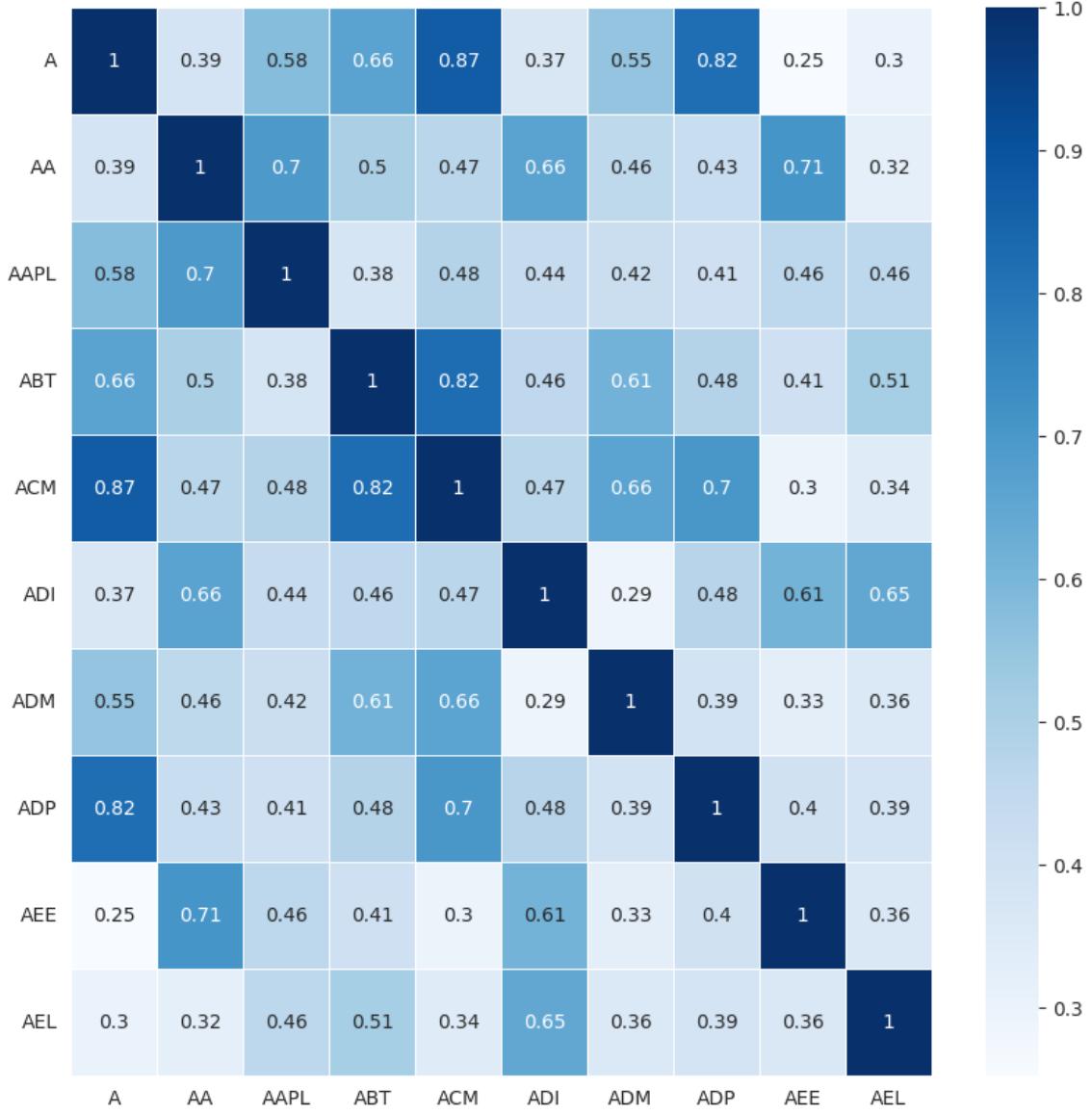


Figure 3.2: Distance correlation matrix for the first 10 companies in the dataset.

This metric ranges from 0, indicating no correlation, to 1, denoting perfect correlation, thus providing a comprehensive measure of both linear and nonlinear associations. Figure 3.2 shows the distance correlation between the first ten companies in the dataset.

The calculation of distance correlation among all firms yields a densely populated dependence matrix, sized [318, 318]. From this matrix, we construct a complete graph in which each node is connected to every other node, irrespective of the correlation strength. This results in a total of 50,403 edges for each graph, encompassing every possible pair of connections. However, the edge weight indicates the strength of the connection between the two nodes it connects.

Following the establishment of these relationships, the Triangulated Maximally Filtered Graph (TMFG) [63] filtering technique is applied to the graphs. The TMFG is a powerful tool for simplifying complex networks while preserving their topological and informational core.

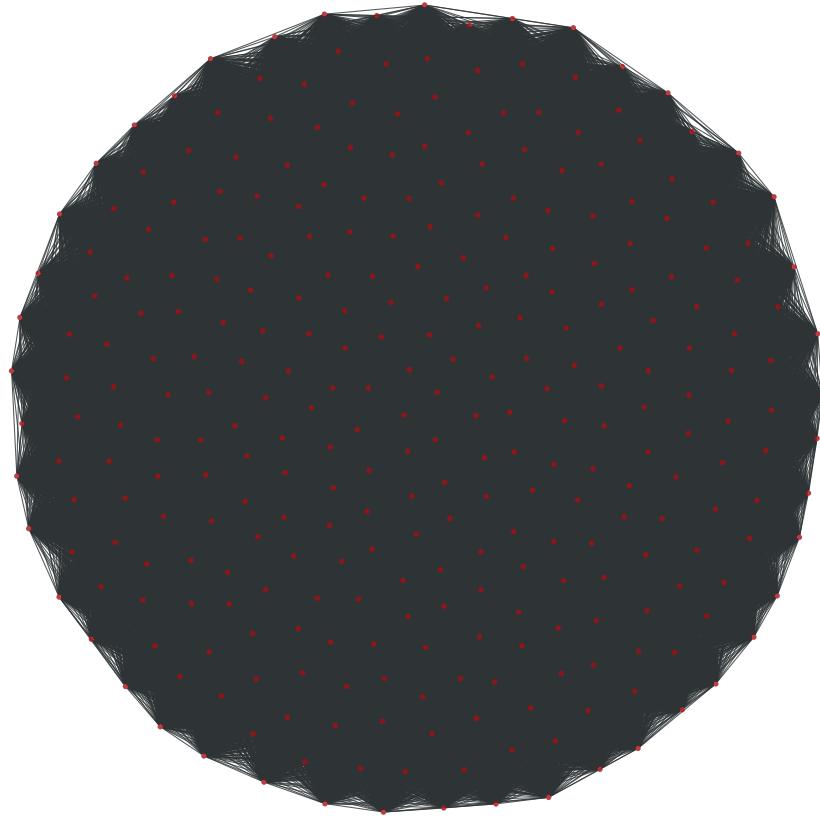


Figure 3.3: Complete graph showing the connections between the companies in the first quarter of 2010 pre-TMFG application.

The TMFG algorithm works by constructing a planar graph that maximizes the sum of the weights of the edges, which represent the strength of the relationships between nodes. This is achieved through a series of steps:

- **Initialization:** Start with a clique of four nodes, which are chosen based on the highest weights of their edges to ensure strong initial relationships.
- **Triangulation Process:**
  - Iteratively add new nodes to the graph.
  - For each new node, identify the triangle (i.e., three connected nodes) that maximizes the weight of the connections when the new node is connected to all three vertices of the triangle.
  - Add the new node to the graph, forming new triangles while maintaining the planarity of the graph.
- **Weight Maximization:**
  - The algorithm seeks to maximize the sum of the weights of the edges in the resulting graph.
  - This is achieved by continuously choosing triangles that result in the highest possible increase in total weight when new nodes are added.

Mathematically, if  $w_{ij}$  represents the weight of the edge between nodes  $i$  and  $j$ , the goal is to maximize the following objective function:

$$\sum_{(i,j) \in E} w_{ij}$$

where  $E$  is the set of edges in the graph. The TMFG ensures that this maximization respects the planarity constraint, meaning no edges cross each other in the two-dimensional representation of the graph.

- **Complexity Management:**

- By focusing on the strongest relationships and maintaining planarity, the TMFG reduces the complexity of the network, making it easier to analyze while retaining its essential structural and informational characteristics.

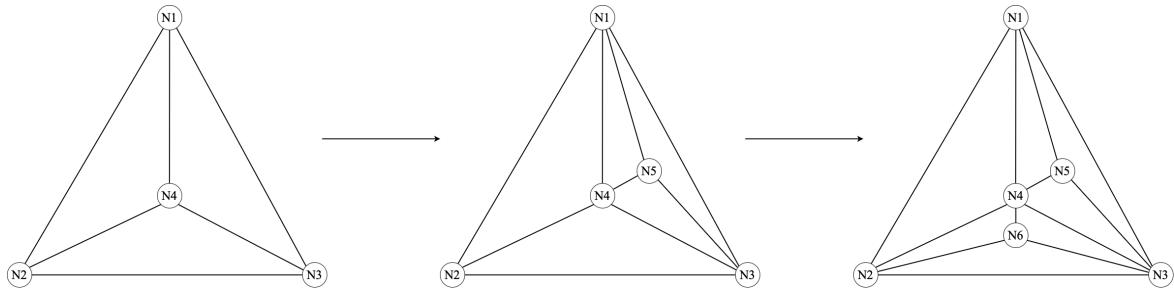


Figure 3.4: Image showing the first two iterations of TMFG triangulation process.

By applying the TMFG algorithm, we significantly reduced the number of edges by more than half in most cases, with an average of 24,692 edges remaining across all 44 graphs. This reduction is particularly beneficial in financial networks where the primary concern is to maintain the most significant relationships. It effectively reduces noise and computational complexity without compromising the structural integrity or the predictive power of the model. This strategic pruning helps in focusing on the most influential and critical connections, ensuring that the essential dynamics within the network are retained and highlighted for analysis. Figure 3.3 and Figure 3.5 illustrate the transformations undergone by the graph for the first quarter of 2010 before and after the application of the TMFG algorithm. These figures visually depict the simplification and structural refinement achieved through the TMFG process.

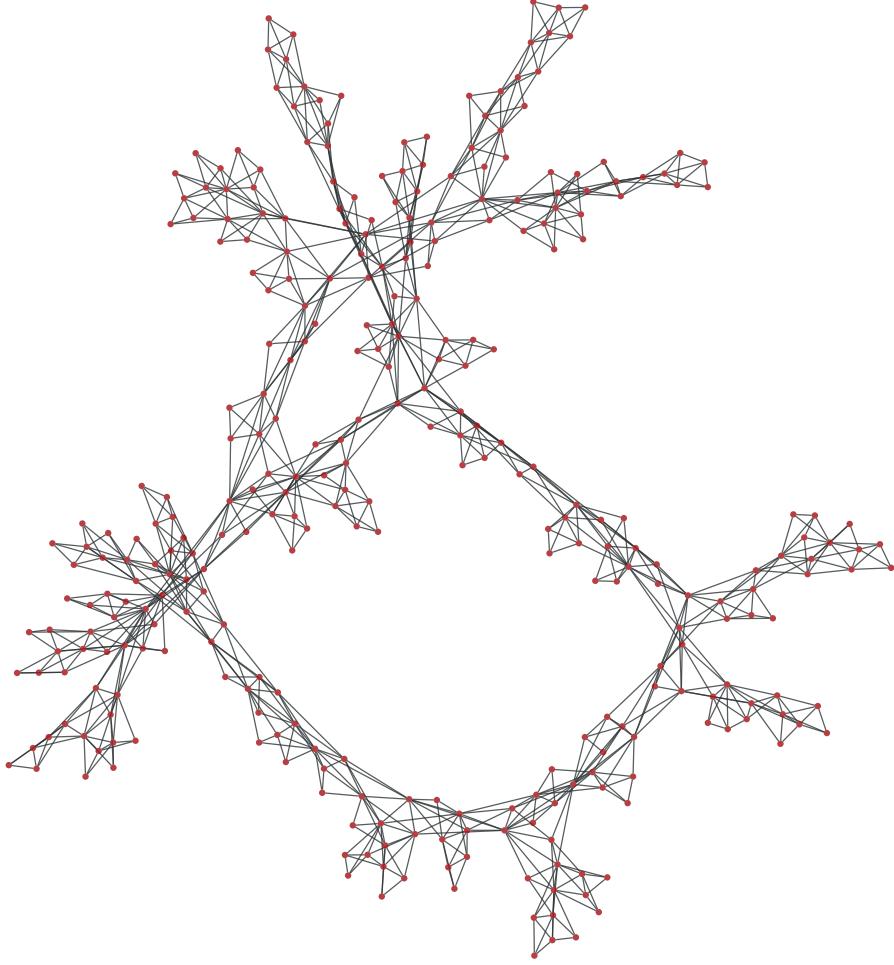


Figure 3.5: Graph showing the connections between the companies in the first quarter of 2010 post-TMFG application.

## 3.2 Model Architecture

Our exploration of multi-modal models focused on the integration of GNNs with various types of RNNs. However, given the temporal characteristics of both datasets, it is important that the GNNs have an integrated RNN component. This integration allows the model to not only leverage the spatial relationships encoded in the graph structure, but also to capture and utilize temporal information, preserving critical data from previous time steps for more accurate predictions and analyses. Therefore, we propose a Recurrent Graph Neural Network (RGNN) architecture to effectively process the graph data. This is then combined with an RNN architecture for the financial statement data, and both streams are combined in a multi-modal model. The following sections describe each component in the architecture.

### 3.2.1 RGNN Architecture

In our RGNN architecture, we incorporate GCNs as the GNN component. GCNs utilize the principles of convolutional processes, similar to those found in traditional

convolutional neural networks (CNNs), but adapt these processes to the unique structure of graph data. In a GCN, each node's representation is updated by aggregating features from its neighboring nodes, effectively leveraging the local graph structure. This aggregation typically involves multiplying the adjacency matrix of the graph with the nodes' feature matrix, then applying a trainable transformation. By doing so, GCNs are capable of capturing the topological relationships inherent in the graph.

GCNs get two main types of input: edge indices and edge weights. Edge indices show which companies are connected. Edge weights, in our case, are based on the distance correlation between firms. This measures the strength of their relationship based on their return-volatility series. This setup helps our models understand both the structure of the corporate network and the intensity of relationships, giving a more complete view of the interconnected financial system.

We also use learnable node (company) embeddings that capture each company's unique traits. These embeddings start as random and improve during training through backpropagation. With each iteration, the embeddings update to better represent the nodes within the graph's structure. This helps the GNNs capture and integrate the evolving features of each company, improving the model's ability to understand complex and changing business environments.

In complement to the GCNs, our RGNN architecture includes the RNN component, where we experiment with both LSTMs and GRUs architectures. These RNNs are tasked with processing the temporal sequences derived from the graph-based features produced by the GCNs. Specifically, LSTMs are employed for their ability to manage longer dependencies in time series data, receiving not only the GCN output but also the last hidden state and cell state from the previous timestep. This enables the LSTM to retain memory over extended sequences, crucial for tracking the progression of corporate financial health.

GRUs, on the other hand, are used for their efficiency in similar settings. Like LSTMs, GRUs receive the output of the GCNs, but they operate with a single hidden state that combines the roles of the hidden and cell states in LSTMs. This simplification allows GRUs to update and reset information through a set of gates, managing the flow of information without the separate cell state, thus making them less computationally intensive while still capturing essential temporal features.

Both LSTM and GRU variants are integrated into the architecture to evaluate their relative efficacy in capturing and leveraging temporal dynamics alongside the structural insights provided by GCNs. This dual approach harnesses the strengths of both graph-based and sequence-based modeling, aiming to provide a holistic analysis of the financial stability and potential risks associated with each entity in the dataset.

### 3.2.2 RNN Architecture

We compare several models on the numerical data, including RNN variants like GRUs and LSTMs, as well as transformers. Transformers [64] leverage self-attention mechanisms to weigh the importance of different parts of the input data, making them highly effective for capturing global dependencies. This improves upon the attention mechanism that GRU and LSTM posses, allowing for a more complex representation of time series.

The RNNs are designed to process numerical sequential data derived from quarterly financial statements of companies. This data encapsulates multiple financial indicators such as revenue, profit margins, liabilities, and assets, arranged in a time series for-

mat. By analyzing these temporal sequences, the RNNs capture the dynamic financial health and performance trends of each company over time. This sequential analysis enables the detection of patterns and potential predictors of future financial stability or distress.

To enhance this process, our LSTM and GRU models harness the hidden state  $h$  from the previous time step,  $t-1$ , as an input at the current time step,  $t$ . This mechanism allows the models to retain information from previous sequences, which is crucial for maintaining context and continuity in the financial data analysis. LSTMs further enhance this capability with an additional cell state that carries relevant information across extended time intervals, facilitating the connection between events that are distant in time. This dual-state mechanism in LSTMs, and the hidden state usage in GRUs, enable these models to capture and learn from long-term dependencies within the data, making them highly effective for complex financial time series analysis where past financial events significantly influence future outcomes. The encoder in our LSTM and GRU architectures computes the hidden state  $h_t$  based on the input sequence  $x_t$  and the previous hidden state  $h_{t-1}$ , formalized as:

$$h_t = RNN_{enc}(x_t, h_{t-1}) \quad (3.3)$$

Here,  $x_t$  represents the input at time step  $t$ , and  $RNN_{enc}$  updates the hidden state based on this input and the preceding state. The encoder processes the entire input sequence, refining its hidden state at each step, culminating in a final state  $h_T$  that encapsulates a compressed representation of the input sequence. This state then initializes the decoder's hidden state  $s_t$ , providing the contextual groundwork necessary for generating the output sequence. The decoder operates under a similar mechanism but focuses on generating the output sequence step-by-step:

$$s_t = RNN_{dec}(y_t, s_{t-1}) \quad (3.4)$$

In this setup,  $y_t$  denotes the output at time  $t$ , influenced by the preceding output. The decoder iteratively builds the output sequence, using its recurrent architecture to remember and incorporate previous outputs, ensuring that the generated sequence is coherent and contextually aligned.

### 3.2.3 Multi-modal Architecture

In our multi-modal model architecture, we employ Intermediate Fusion (IF) [65] as the technique to integrate the outputs from the RGNNs and RNNs. This method of fusion involves combining the outputs from both model streams before they undergo the final prediction layer. By doing this, IF allows the model to exploit the combined strengths of both structural and sequential insights at a deeper level within the network, rather than simply combining inputs at the beginning (Early Fusion) or aggregating final outputs (Late Fusion) [65].

After the fusion, the combined outputs are passed to a decoder layer, which comprises two linear layers. These layers output a vector corresponding to the number of classes in our problem. This vector is then used by the log-softmax function to produce predictions for our classes. The softmax function, widely used in multiclass classification problems, converts the output scores from the model into probabilities. It achieves this by taking the exponent of each output and normalizing these values

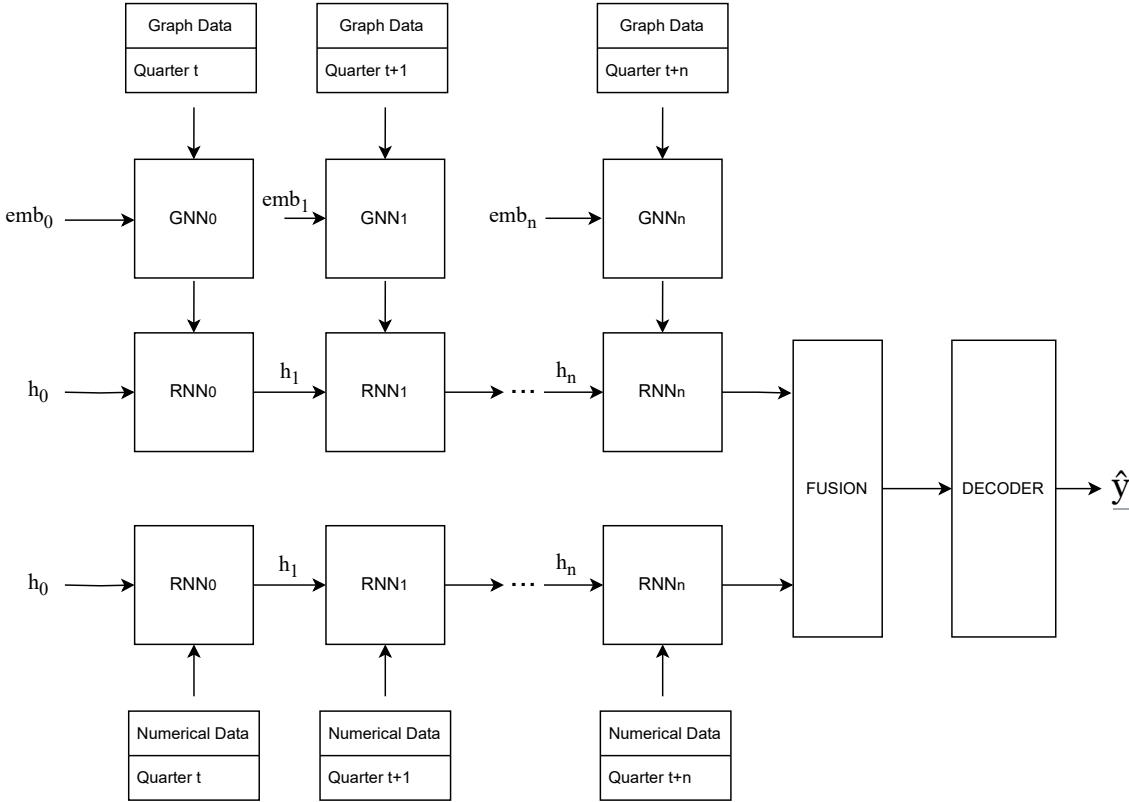


Figure 3.6: An overview of the RNN-RGNN multi-modal architecture.

by dividing by the sum of all the exponents, thus producing the model's final output for multiclass classification.

$$\text{softmax} = \sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \text{for } i = 1, 2, \dots, K \quad (3.5)$$

This results in a probability distribution over the classes, where the sum of probabilities equals one. The log-softmax is a logarithm of the softmax function, combining the steps of applying softmax and then taking the logarithm of the output. This is particularly useful for numerical stability and efficiency in computation. Using softmax ensures that our model's predictions can be interpreted directly as probabilities for each class, facilitating a clear and actionable understanding of the credit risk associated with each company.

### 3.3 Loss Function and Optimization Techniques

#### 3.3.1 Loss Function

The Cross Entropy Loss function is utilized as our loss metric. Cross Entropy Loss is particularly suited for multiclass classification tasks, as it measures the performance of a classification model whose output is a probability value between 0 and 1. It effectively captures the distance between the model's predicted probability distribution and the true distribution, with a lower loss indicating a model that's more accurate. This loss function is advantageous because it penalizes incorrect classifications more severely when the model is confident about its incorrect predictions, thus driving the model

towards more accurate and confident classifications. Cross Entropy loss for multiclass classification is calculated as follows:

$$CE = - \sum_{c=1}^M y_{o,c} \log(p_{o,c}) \quad (3.6)$$

where  $M$  is the total number of classes. The summation runs over all classes, where  $y_{o,c}$  is a binary indicator (0 or 1) that is 1 if class label  $c$  is the correct classification for observation  $o$ , and  $p_{o,c}$  is the predicted probability that observation  $o$  belongs to class  $c$ .

### 3.3.2 Optimizer

For optimization, we selected the Adam optimizer, which stands for Adaptive Moment Estimation. Adam is renowned for its efficiency in handling sparse gradients and its adaptability to the problem's scale, thanks to its moment-based approach [66]. It combines the advantages of two other extensions of stochastic gradient descent: AdaGrad, which works well with sparse gradients, and RMSProp, which works well in online and non-stationary settings. Adam achieves this by maintaining a learning rate for each parameter, which it adjusts as learning progresses. This optimizer starts with an initial learning rate of 0.01 and implements a weight decay of 5e-4 to regularize and aid with overfitting.

In our hyper-parameter tuning phase, we also considered the AdamW optimizer, an extension of the traditional Adam optimizer that includes a more sophisticated weight decay mechanism. AdamW separates the weight decay from the gradient updates, which can lead to better training stability and model performance in certain cases [67]. However, in our tests, Adam consistently outperformed AdamW, leading us to select Adam as the optimizer for our final models.

### 3.3.3 Adaptive Learning Rate

To further mitigate overfitting and adaptively adjust the learning rate, we made use of a learning rate scheduler, with 10-epoch patience, a reduction factor of 0.1, and a minimum learning rate set at 1e-6. The method (Pytorch's *ReduceLROnPlateau*) adjusts the learning rate when a metric has stopped improving, effectively allowing the model to escape plateaus or local minima in the loss landscape. By reducing the learning rate after the specified 'patience' period of no improvement, it helps fine-tune the model convergence, leading to better overall performance.

### 3.3.4 Gradient Clipping

To prevent the common issue of exploding and vanishing gradients often encountered with RNNs [68], we incorporated gradient clipping into our training process. Gradient clipping is a technique where gradients are truncated during the backpropagation phase to ensure they do not exceed a specified threshold. This is crucial because too large or too small gradients can cause the learning process to diverge, leading to model instability. By applying gradient clipping, we cap the gradients to a manageable range, thus maintaining the stability of the network's updates. This method not only prevents the detrimental effects of exploding gradients but also ensures that our model trains

reliably over extensive epochs, which is particularly important given the complexity and depth of our RNN and RGNN architectures. During training, we employ the PyTorch’s `clip_grad_norm_` function to limit the gradients of the neural network parameters to a maximum norm of 1.0. This procedure scales down the gradients to a manageable size, effectively preventing exploding gradients and ensuring stable training progression.

## 3.4 Evaluation Metrics and Validation

Given the imbalance present in our multiclass classification problem, selecting appropriate evaluation metrics is crucial to accurately assess each model’s performance. To this end, we utilize the Area Under the Receiver Operating Characteristic Curve (AUC), Area Under the Precision-Recall Curve (AUPRC), and the F1 score as our primary metrics.

### 3.4.1 AUC

The AUC measures the ability of a model to distinguish between classes and is used as a summary of the Receiver Operating Characteristic (ROC) curve. The ROC curve plots the true positive (TP) rate against the false positive (FP) rate at various threshold settings, making the AUC a measure of a model’s ability to classify with accuracy irrespective of the classification threshold. Its value ranges from 0 to 1, where a value of 1 indicates a perfect model and 0.5 represents a model performing no better than random chance. The AUC is particularly valuable in the context of our imbalanced dataset because it is scale-invariant, meaning it measures how well predictions are ranked rather than their absolute values. It is also classification-threshold-invariant, offering a measure of model quality that is independent of the specific decision threshold used to classify observations into different classes. According to Davis and Goadrich [69], these properties make AUC an ideal metric for assessing classification performance, especially in imbalanced situations where positive and negative classes do not contribute equally to the overall accuracy.

Given the multiclass nature of our data, we employ a weighted average of AUCs for each class in a one-versus-all approach. This method treats each class as a separate binary classification problem (the given class versus all other classes), allowing us to capture the model’s performance across all classes despite the class imbalance.

### 3.4.2 F1 Score

The F1 score is another critical metric we use, which combines precision and recall into a single measure. Precision is the number of true positives divided by the number of true positives and false positives, indicating the accuracy of positive predictions. Recall, on the other hand, is the number of true positives divided by the number of true positives and false negatives (FN), reflecting the model’s ability to identify all relevant instances. The F1 score is the harmonic mean of precision and recall, providing a balance between them and can be written as:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \quad (3.7)$$

The F1 score ranges from 0 to 1, where 1 indicates perfect precision and recall, and a higher score signifies better model performance. F1 score is particularly useful in situations where there is an uneven class distribution, as is the case in our dataset. Using the F1 score in conjunction with the AUC allows for a more comprehensive evaluation of our models, considering both the imbalance in class distribution and the need for a balanced trade-off between precision and recall.

### 3.4.3 AUPRC

In addition to the AUC and F1 score, we also incorporate the AUPRC as a critical metric for evaluating our models. The Precision-Recall Curve (PRC) plots the precision against the recall at various threshold levels. AUPRC then measures the area under this curve, providing a single value that represents the model’s ability to identify positive instances across different thresholds of precision and recall. It is particularly useful when the class distribution is imbalanced [70].

## 3.5 Baseline Models

As a baseline to compare our RNN-RGNN model against, we employed XGBoost, a highly efficient and scalable implementation of gradient boosting decision trees, known for its performance in a wide range of tasks [71]. For the XGBoost model, we utilized only the quarterly financial statement dataset, excluding the graph-based components. Additionally, we also use our RNN and RGNN models individually as baseline models. This strategy enables us to further isolate and understand the contributions of each model type and data source –whether based on sequential financial data or graph-based data– when compared to the integrated multi-modal models. This comprehensive approach allows us to evaluate the added value of our RNN-RGNN architectures and establish a clear benchmark for assessing the effectiveness of combining multiple data modalities in credit risk prediction.



# Chapter 4

## Experimental setup

### 4.1 Dataset

Our dataset encompasses 44 graphs representing 318 nodes, which correspond to 44 quarterly periods for 318 companies from 2010 to 2020. This results in 13,992 entries within the quarterly financial statement dataset, with each of the 318 companies contributing 44 quarterly statements. To further challenge and evaluate the robustness of our models, we expanded the test set to include data from additional 112 "unseen" companies. These unseen companies do not have any entries in the training set, effectively serving as new instances that the model must evaluate without prior direct knowledge. Adding unseen companies to the test set is a significant step in assessing the generalization capabilities of our models [52]. This approach ensures that the models are not only good at predicting outcomes for data similar to what they were trained on but also capable of evaluating the credit risk for entirely new companies. This is particularly important in the real-world application of these models, where financial institutions frequently encounter new clients or businesses that do not have historical data in their systems.

### 4.2 Training Procedure

To facilitate training and testing, we employ a rolling-window approach, shifting by one quarter at a time. Each window comprises four snapshots, with each snapshot corresponding to a quarter's worth of data. The objective during the training of each window is to predict the credit rating class for all 318 firms for the quarter immediately following the last snapshot within that window. We divided the dataset into three parts: the training set includes the first 28 windows, which correspond to 70% of the data, the validation set encompasses the next 4 windows, making up 10% of the data, and the test set comprises the final 8 windows, accounting for 20% of the total dataset. This distribution ensures a comprehensive evaluation of the model's performance across different stages of the data while keeping the train and test set separated in time in order to avoid data leakage. Other sequence lengths, specifically 2, 6, 8, 10, and 12, were also evaluated and their outcomes are detailed in the results chapter.

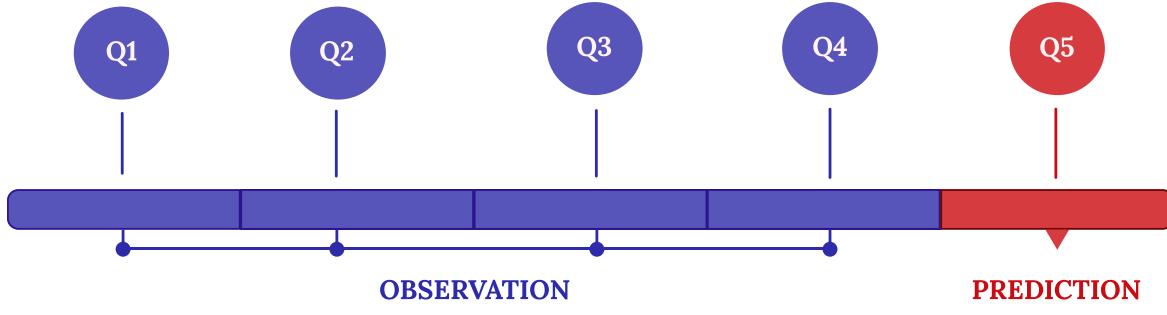


Figure 4.1: An example of one observation window in the rolling window approach.

### 4.3 Hyper-parameter Tuning

To optimize the performance of our models, we employed a comprehensive approach for hyper-parameter tuning. Each model underwent training for 100 epochs, starting with an initial learning rate of 0.01. To adapt to the training dynamics, we used an adaptive learning rate scheduler that gradually reduced the rate to a minimum of 1e-6, ensuring efficient convergence. The models were optimized using either the Adam or AdamW optimizer, depending on which yielded better performance during preliminary tests.

The selection of the best hyper-parameter configurations was performed using an exhaustive grid search strategy. This method systematically explored a predefined hyper-parameter space, as detailed in Table 4.1. The space included variations in the number of layers and hidden dimensions for both RGNN and RNN architectures, as well as the node embedding size for GNN nodes. This thorough exploration was aimed at identifying the optimal settings that enhance model learning and generalization, thereby improving the accuracy and robustness of the predictions. Each combination of parameters was evaluated to determine its impact on the model's performance, facilitating the identification of the most effective model configuration.

Table 4.1: Hyper-parameter space

Hyper-parameter	Value range	Model
RGNN layers	{2, 16}	RGNN, RNN-RGNN
RNN layers	{2, 16}	RNN, RNN-RGNN
RGNN hidden dimensions	{16, 512}	RGNN, RNN-RGNN
RNN hidden dimensions	{16, 512}	RNN, RNN-RGNN
GNN node embedding size	{16, 201}	RGNN
Optimizer	{Adam, AdamW}	RGNN, RNN-RGNN

### 4.4 Experiments

We aim to compare various RNN-RGNN configurations and benchmark their performance against baseline models to understand their efficacy in handling both seen and unseen nodes. This comparative analysis will help identify the strengths and limitations of each configuration in recognizing patterns and predicting outcomes accurately across different sets of data. To enhance the reliability of our evaluation, the results are presented with 95% confidence intervals derived from bootstrapping over the test set.

This statistical method involves repeatedly sampling with replacement from the test set and recalculating the performance metrics for each sample. Presenting results with confidence intervals offers a clear view of the variability and stability of the model’s performance, providing insights into how the results might generalize to similar but unseen data and is replicable in practical scenarios.



# Chapter 5

## Results and Discussion

### 5.1 Baseline Models

The performance table of baseline models in Table 5.1 and Table 5.2 provide a clear overview of how individual models, either solely based on financial data or graph data, compare in terms of F1 score, AUC, and AUPRC for both seen and unseen nodes with a window of size 4. In the 'Modality' column of the tables, 'G' indicates that the model receives graphs as input, specifically edge indices and edge weights combined with learnable node embeddings. Conversely, 'N' denotes that the models utilize numerical data sourced from quarterly financial statement reports.

Table 5.1: Performance of Baseline Models on Seen Companies

Model	Modality	F1 Score	AUC	AUPRC
GRU	N	<b>0.7267 ± 0.019</b>	<b>0.9429 ± 0.006</b>	<b>0.7659 ± 0.024</b>
Transformers	N	0.6779 ± 0.020	0.7704 ± 0.006	0.3159 ± 0.014
LSTM	N	0.7046 ± 0.020	0.9327 ± 0.007	0.7456 ± 0.023
GCNLSTM	G	0.4094 ± 0.021	0.8358 ± 0.010	0.4242 ± 0.023
GCNGRU	G	0.4281 ± 0.021	0.8477 ± 0.011	0.4652 ± 0.011
XGBoost	N	0.6967 ± 0.011	0.9224 ± 0.005	0.7464 ± 0.005

Table 5.2: Performance of Baseline Models on Unseen Companies

Model	Modality	F1 Score	AUC	AUPRC
GRU	N	0.1760 ± 0.045	0.6930 ± 0.041	0.2256 ± 0.046
Transformers	N	0.1865 ± 0.011	0.6927 ± 0.036	0.2016 ± 0.036
LSTM	N	0.1834 ± 0.039	<b>0.7389 ± 0.040</b>	<b>0.3060 ± 0.053</b>
GCNLSTM	G	0.1244 ± 0.024	0.6151 ± 0.051	0.1407 ± 0.032
GCNGRU	G	0.0945 ± 0.032	0.5853 ± 0.050	0.1401 ± 0.045
XGBoost	N	<b>0.2465 ± 0.040</b>	0.7077 ± 0.035	0.2682 ± 0.034

#### 5.1.1 RNN Models, Transformers

In the comparison of baseline RNN models, the GRU exhibits the best performance among seen nodes with an F1 score of 0.7267 and an AUC of 0.9429, highlighting its

efficiency in handling the temporal dynamics of financial data. Notably, its AUPRC of 0.7659 also suggests good balance in precision and recall. In contrast, the Transformers model shows the lowest seen performance metrics with an F1 of 0.6779 and a significantly lower AUC of 0.7704, which might indicate its lesser effectiveness in this specific financial context. Transformers tend to be data hungry to train effectively, and the data diversity in this case may not be enough to achieve significant gains. The LSTM model, while performing better than the Transformer with an F1 of 0.7046 and an AUC of 0.9327, shows a marked ability to generalize to unseen data with the highest AUC of 0.7389 and AUPRC of 0.3060 among the RNNs, emphasizing its potential for handling new, unseen financial patterns more effectively than its counterparts.

### 5.1.2 RGNN Models

For models incorporating GCNs combined with RNNs, the results for the best models are significantly lower than standalone RNNs. The models perform poorly, particularly on unseen nodes, indicating a critical limitation in handling new or structurally different data. This poor performance could be due to the lack of sufficient node features (since only learnable node embeddings were used) and perhaps indicates that GCNs require richer feature sets or fail to capture complex patterns solely with structural data. Another potential explanation is that there is significant information in the dynamism of corporate default, so using models that only embed graph information is not sufficient to accurately describe the dynamics that lead to a large corporation to default.

### 5.1.3 XGBoost

As a non-sequential and non-graph baseline model, XGBoost demonstrates strong performance on seen nodes, with results comparable to some RNN models. However, XGBoost maintains the highest F1 score among all baseline models on unseen nodes. This underscores XGBoost's robustness and ability to handle diverse data scenarios better than complex models like GNNs, making it a reliable choice for scenarios where generalization across varied data samples is crucial.

While the RNNs and XGBoost show promise in balancing precision and recall and handling unseen data to some extent, the GNNs exhibit significant limitations in this regard. This suggests the need for further research into improving the robustness and adaptability of the GNN architectures, possibly through techniques such as meta-learning, few-shot learning, or enhanced feature engineering to better capture the nuances of new and diverse data scenarios.

## 5.2 RNN-RGNN Models

Table 5.3 and Table 5.4 detail the performance metrics of our top five RNN-RGNN models for both seen and unseen nodes, using a window size of 4.

The examination of the performance of RNN-RGNN models relative to standalone RNN and RGNN models reveals some distinct advantages of the multi-modal approach, particularly in handling unseen data. As shown in Table 5.3, the RNN-RGNN models generally match or surpass the highest scores obtained by standalone models in seen

Table 5.3: Performance of Top 5 RNN-RGNN Models on Seen Companies

Model	Modality	F1 Score	AUC	AUPRC
LSTM-GCNLSTM	N+G	$0.7318 \pm 0.020$	<b><math>0.9474 \pm 0.006</math></b>	<b><math>0.7838 \pm 0.023</math></b>
TRA-GCNGRU	N+G	$0.5013 \pm 0.021$	$0.8957 \pm 0.009$	$0.5785 \pm 0.026$
GRU-GCNLSTM	N+G	<b><math>0.7337 \pm 0.021</math></b>	$0.9467 \pm 0.007$	$0.7801 \pm 0.024$
LSTM-GCNGRU	N+G	$0.7015 \pm 0.021$	$0.9377 \pm 0.007$	$0.7673 \pm 0.024$
GRU-GCNGRU	N+G	$0.7275 \pm 0.021$	$0.9444 \pm 0.007$	$0.7776 \pm 0.025$

Table 5.4: Performance of Top 5 RNN-RGNN Models on Unseen Companies

Model	Modality	F1 Score	AUC	AUPRC
LSTM-GCNLSTM	N+G	<b><math>0.3443 \pm 0.060</math></b>	$0.7734 \pm 0.049$	$0.4105 \pm 0.080$
TRA-GCNGRU	N+G	$0.3068 \pm 0.051$	<b><math>0.8247 \pm 0.036</math></b>	<b><math>0.4232 \pm 0.069</math></b>
GRU-GCNLSTM	N+G	$0.1963 \pm 0.045$	$0.7919 \pm 0.043$	$0.3232 \pm 0.064$
LSTM-GCNGRU	N+G	$0.2737 \pm 0.053$	$0.7669 \pm 0.046$	$0.3566 \pm 0.071$
GRU-GCNGRU	N+G	$0.2233 \pm 0.049$	$0.7532 \pm 0.049$	$0.3621 \pm 0.069$

data scenarios. For instance, the LSTM-GCNLSTM model demonstrates superior performance with the highest F1 score (0.7318), AUC (0.9474), and AUPRC (0.7838) among seen nodes, suggesting that the integration of LSTM and GCN in a single framework effectively captures both temporal and structural dependencies better than standalone models over known data.

For unseen nodes, the enhancement is even more pronounced and suggests that a temporal graph model is indeed displaying structures that are generalizable and improve prediction versus other methods, even in evolving landscapes. The LSTM-GCNLSTM model, for example, achieves an F1 score (0.3443), AUC (0.7734), and AUPRC (0.4105), which are significantly higher than any standalone model’s performance metrics on unseen nodes.

Moreover, while the TRA-GCNGRU did not lead in seen nodes performance, it marked the best AUC (0.8247) and AUPRC (0.4232) for unseen nodes, underscoring the potential of integrating Transformer models with GCN for enhancing predictive accuracy on novel data. This again hints at the complexity of the data. In more dynamic markets, with a significant number of entrants, it may be worth it to collect data to allow a transformer model to operate effectively. In more stable environments, a simpler LSTM/GRU model would suffice.

Overall, these diverse performances across the models highlight the intricate trade-offs involved in model architecture choices for corporate credit risk prediction. These results suggest that multi-modal models, by leveraging complementary features from both RNNs and GNNs, provide a more holistic understanding of the data, thus enabling more reliable predictions across different scenarios.

### 5.3 Window Size Comparison

In the analysis of various RNN-RGNN configurations across multiple observation window sizes, a recurring theme emerged, as presented in Figure 5.1, where the window size of 4 (one full year) consistently demonstrated either peak or near-peak performance

metrics across models for F1 Score, AUC, and AUPRC. Although the differences in performance metrics between the window sizes are generally minor, the size of 4 had the highest count of highest scores, suggesting that a 4-quarter look-back period strikes an optimal balance in capturing relevant temporal dependencies and patterns, specifically for our model architectures and datasets. This window size is long enough to encompass significant historical data that may influence the current credit standing, but short enough to prevent the dilution of predictive power through the inclusion of outdated or irrelevant information.

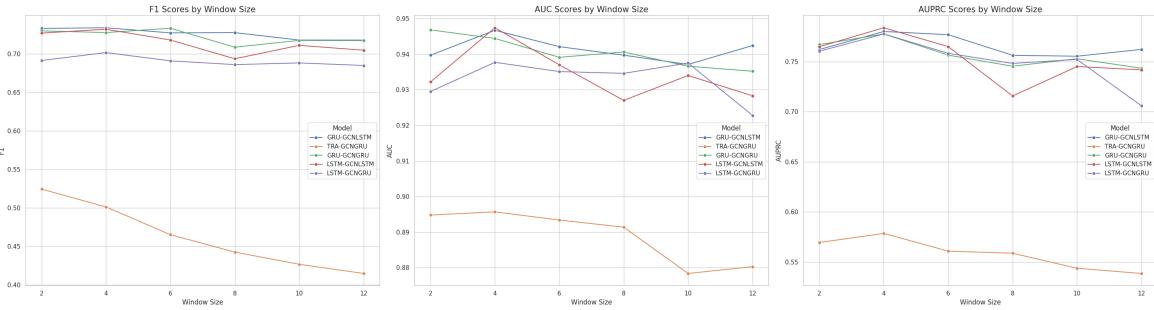


Figure 5.1: Comparison of model performances across different observation window sizes.

## 5.4 Performance and Generalization Capabilities

In the analysis of corporate credit risk prediction models, the convergence plots for training and validation losses, AUC, and AUPRC across different epochs provide critical insights into model performance and generalization capabilities. In Figure 5.2, we present the training and validation loss curves, along with the AUC and AUPRC convergence plots, for the consistently effective LSTM-GCNLSTM model. Figure 5.2 a) displays the training and validation loss curves over 100 epochs, showcasing a steady decrease in loss for both training and validation sets, indicating effective learning and generalization without significant overfitting. The validation loss demonstrates slight fluctuations but maintains a consistent downward trend, affirming the model's robustness.

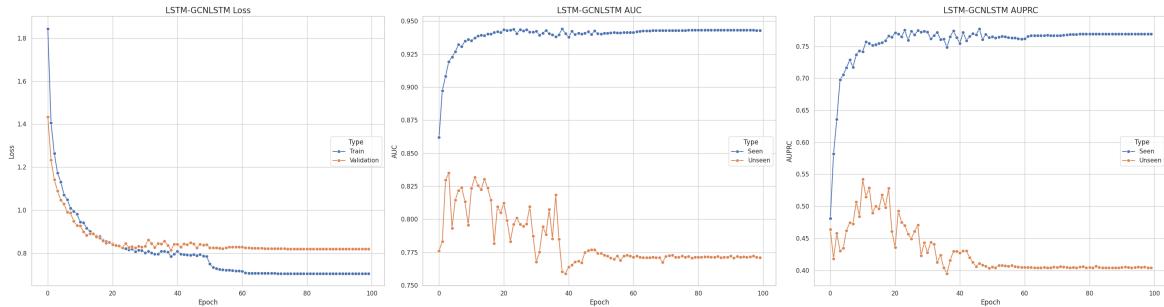


Figure 5.2: a) Train/Validation loss. b) AUC convergence plot. c) AUPRC convergence plot.

Figure 5.2 b) displays the AUC scores for both seen and unseen nodes. For seen nodes, the AUC consistently rises until it stabilizes at approximately 0.94 around the 50th epoch, indicating the model's increasing proficiency in class differentiation as

training advances. In contrast, the AUC scores for unseen nodes follow a more erratic path but eventually converge around the 50th epoch with an AUC of approximately 0.77. This highlights the challenges the model faces in generalizing to new data, yet it also shows a significant capability to adapt and learn from patterns not previously encountered during training. This ability is crucial for practical applications where models must perform well across both familiar and novel scenarios.

In Figure 5.2 c), the AUPRC curves similarly demonstrate the model's performance with a detailed focus on precision-recall balance. The AUPRC scores for seen nodes are significantly higher than those for unseen nodes, showing a consistent upward trend until approximately the 60th epoch, at which point the model converges with an AUPRC score of 0.7734. In contrast, the AUPRC scores for unseen nodes display more variability initially as the model adapts to new patterns. However, these scores begin to stabilize around the 50th epoch with an AUPRC score of 0.41, indicating that the model achieves a degree of reliability in handling classes with fewer examples even in new, unencountered data. This stability is crucial for applications in dynamic environments where the ability to generalize effectively to new entities can significantly impact decision-making processes.

The marked performance disparity between the seen and unseen nodes observed in Figure 5.2 underscores the significant challenges in model generalization to new nodes. This variation not only highlights the difficulty of extending learned patterns to novel data but also emphasizes the critical need for developing strategies aimed at enhancing the model's predictive accuracy for underrepresented classes in unseen data. Such strategies could include more sophisticated data augmentation techniques, advanced ensemble methods to boost the robustness of predictions, or employing transfer learning to leverage knowledge from related tasks. Implementing these approaches could potentially improve the model's utility in practical settings, where it is often required to make reliable predictions on entirely new entities.

## 5.5 Error Analysis

In this section, we analyze the performance of all models using their respective confusion matrices, focusing on accuracy and the distribution of adjacent versus non-adjacent misclassifications. This analysis helps us understand the strengths and weaknesses of each model, especially in the context of credit risk assessment, where minimizing underestimations and overestimations is crucial.

### 5.5.1 Accuracy and Misclassifications

We calculated the accuracy for each model and examined the pattern of misclassifications to determine if they are adjacent or non-adjacent. Adjacent misclassifications occur between neighboring classes and are usually less concerning in ordinal classification tasks. In contrast, non-adjacent misclassifications can lead to significant overestimations or underestimations of financial stability, posing higher risks for the lender.

- GCNGRU (unimodal graph data):

- Accuracy: 37.92%
- Adjacent Misclassifications: 817 (59.1%)
- Non-Adjacent Misclassifications: 565 (40.9%)

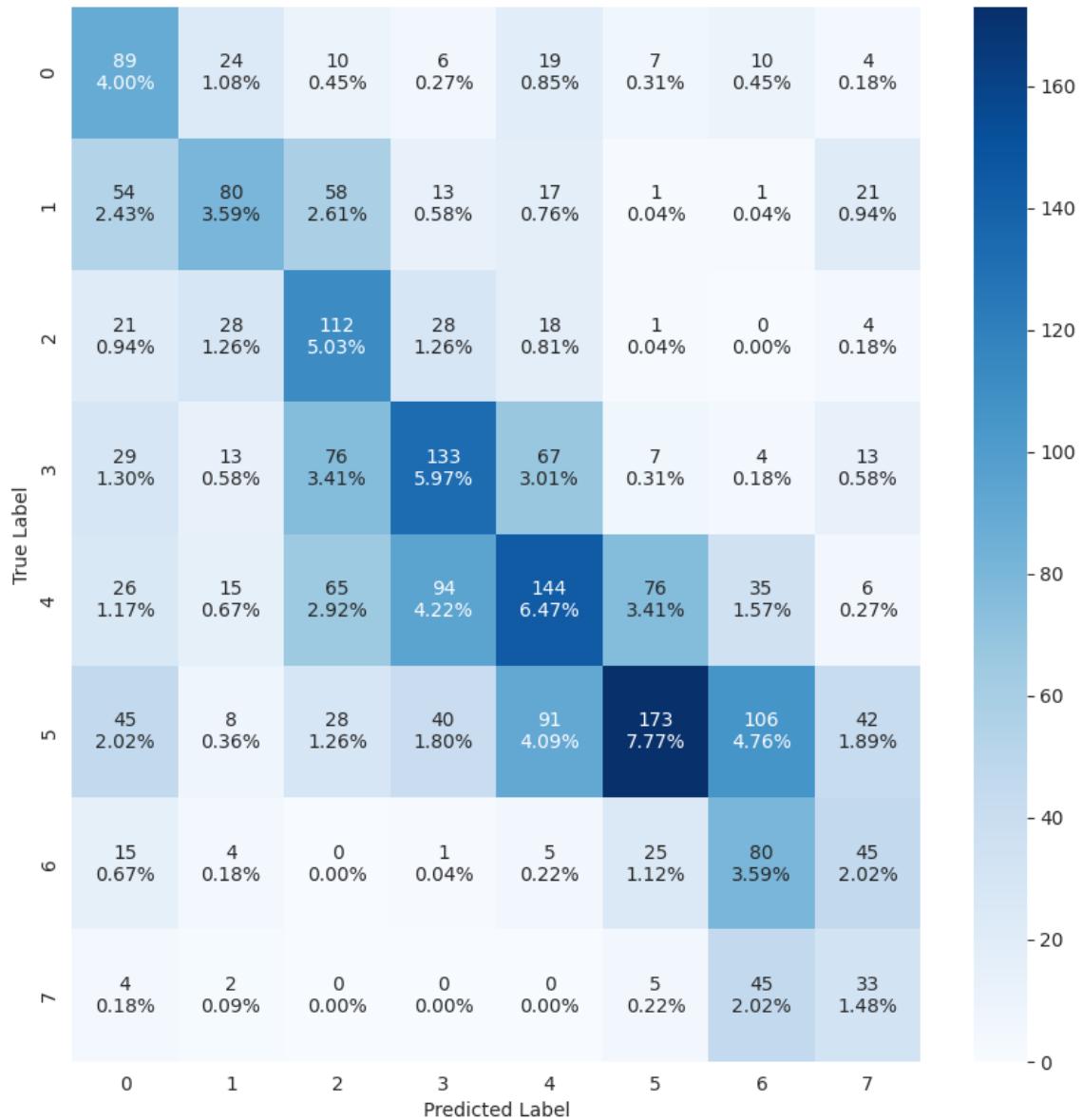


Figure 5.3: Confusion Matrix for the GCNGRU model.

- GCNLSTM (unimodal graph data):

- Accuracy: 43.80%
- Adjacent Misclassifications: 737 (58.9%)
- Non-Adjacent Misclassifications: 514 (41.1%)

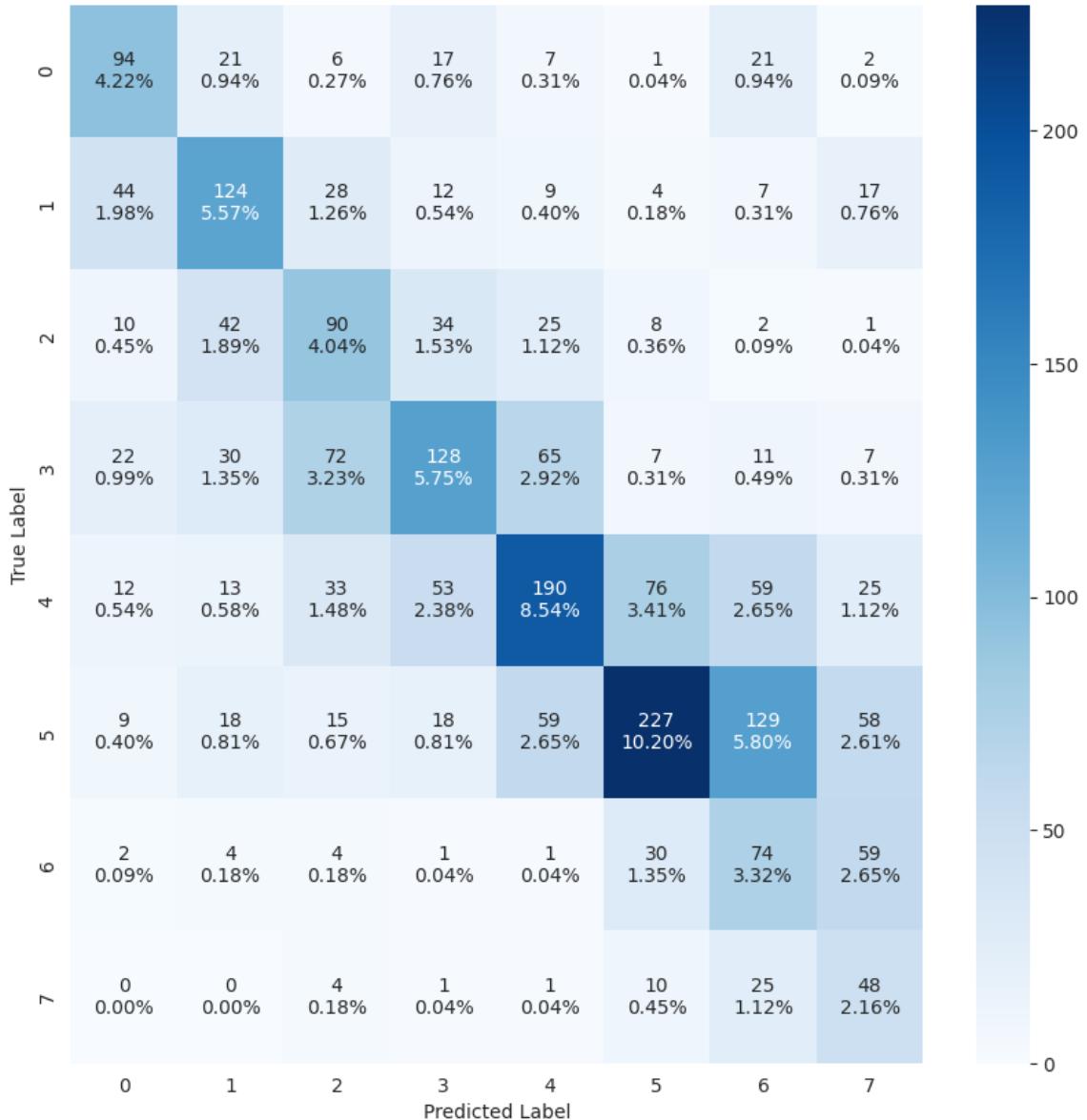


Figure 5.4: Confusion Matrix for the GCNLSTM model.

- LSTM (unimodal numerical data):
  - Accuracy: 73.76%
  - Adjacent Misclassifications: 391 (66.9%)
  - Non-Adjacent Misclassifications: 193 (33.1%)

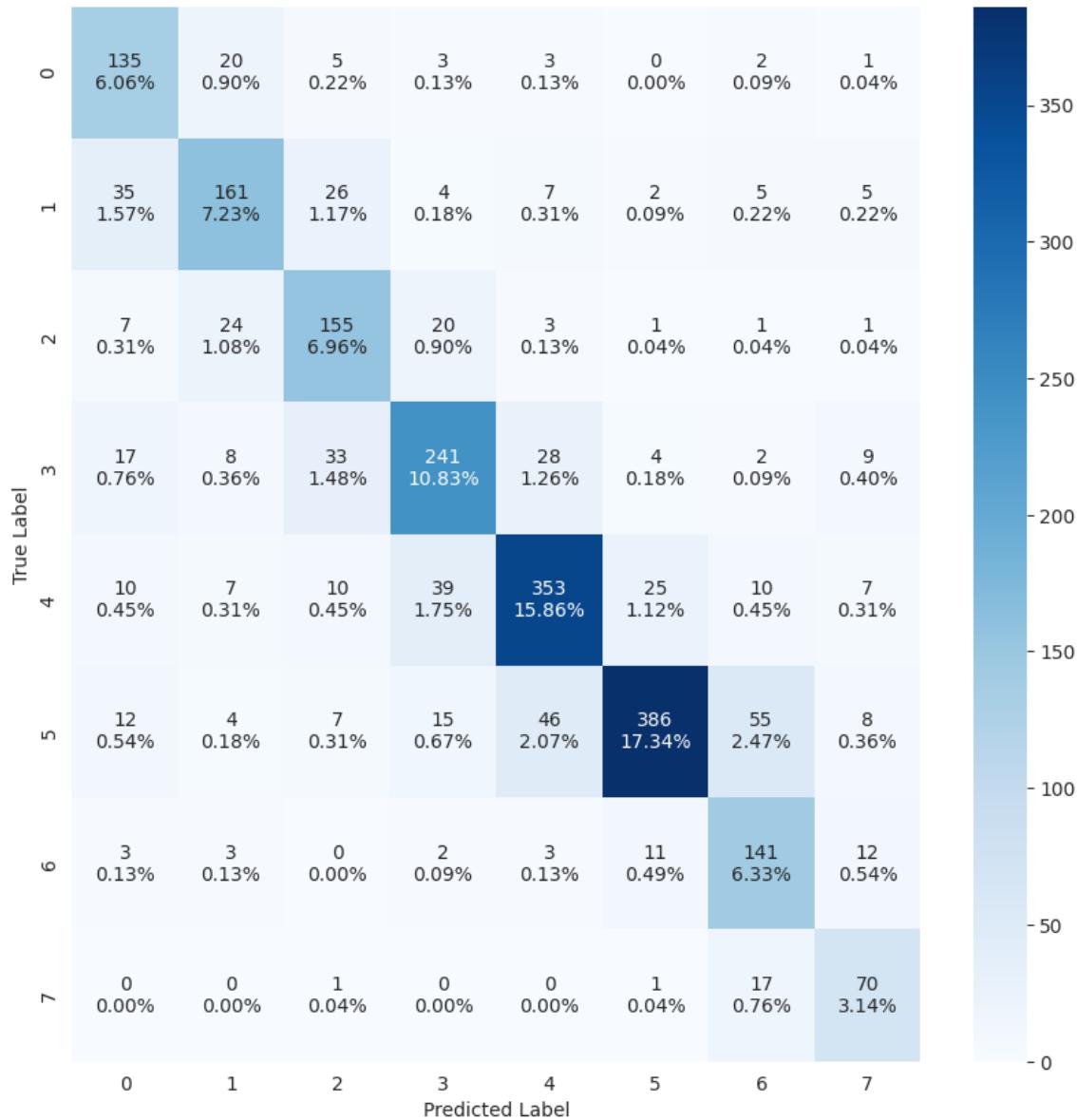


Figure 5.5: Confusion Matrix for the LSTM model.

- GRU (unimodal numerical data):

- Accuracy: 75.25%
- Adjacent Misclassifications: 368 (66.8%)
- Non-Adjacent Misclassifications: 183 (33.2%)

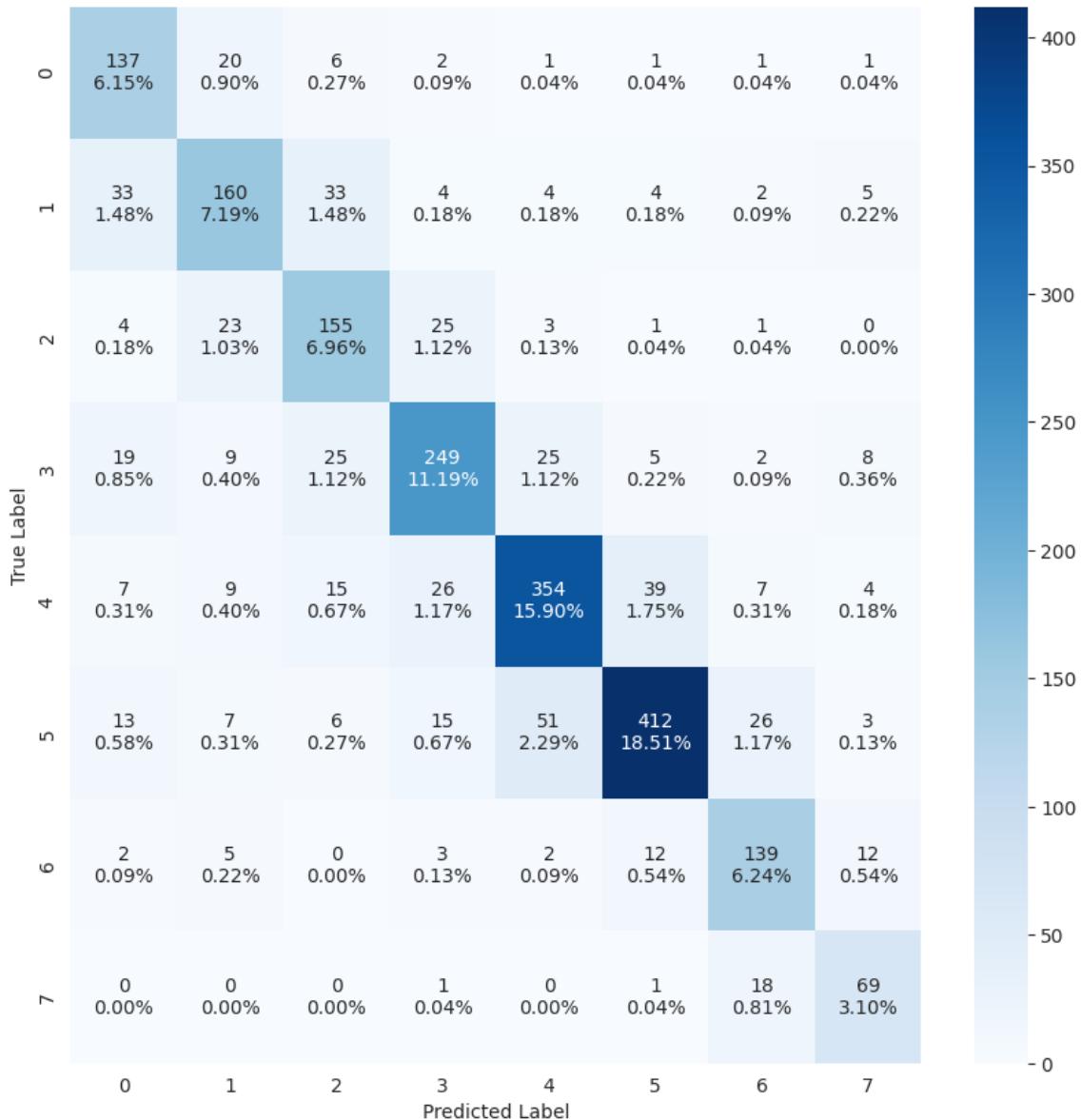


Figure 5.6: Confusion Matrix for the GRU model.

- Transformers (unimodal numerical data):

- Accuracy: 31.04%
- Adjacent Misclassifications: 703 (45.8%)
- Non-Adjacent Misclassifications: 832 (54.2%)

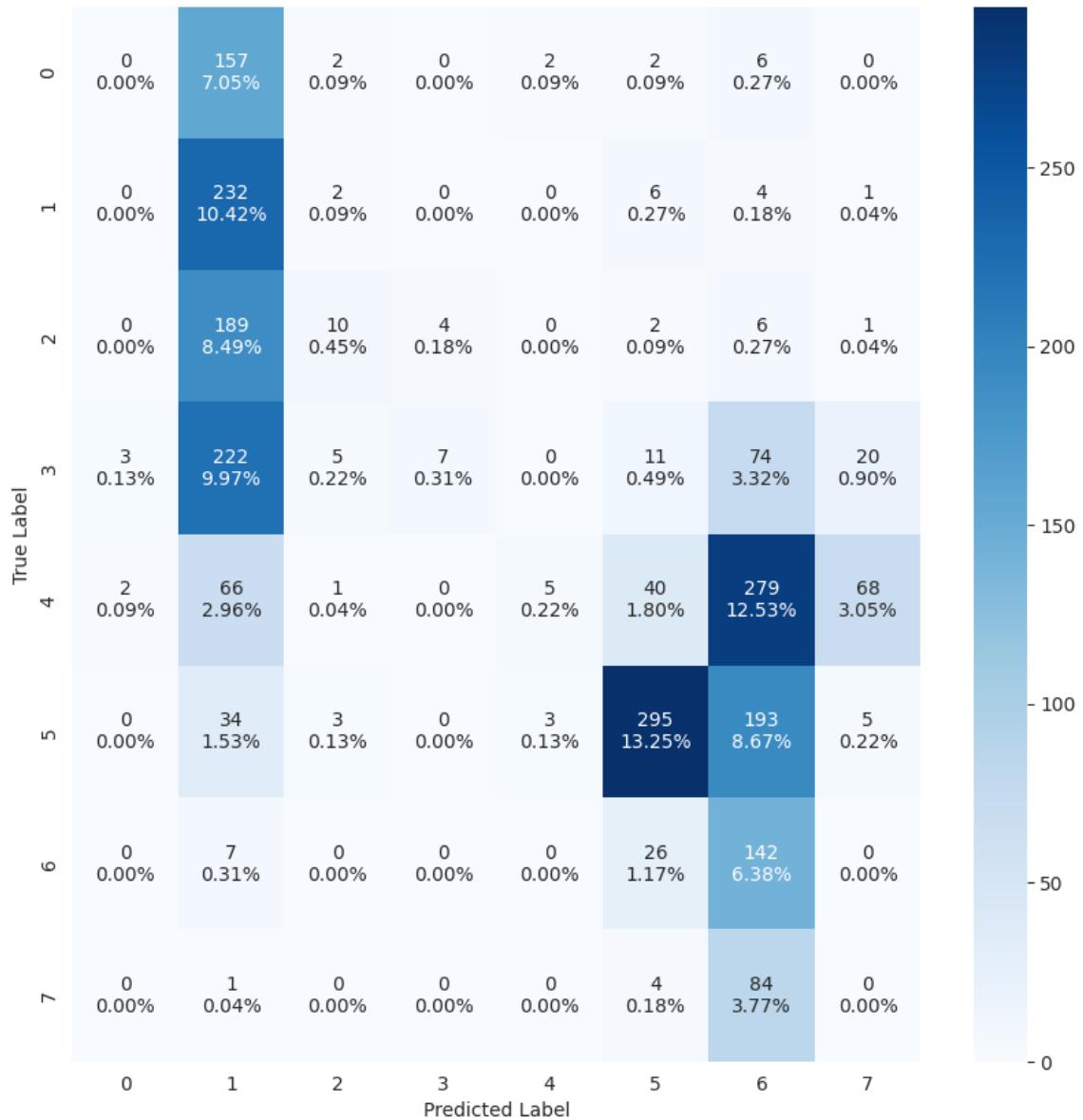


Figure 5.7: Confusion Matrix for the Transformers model.

- XGBOOST (unimodal numerical data):

- Accuracy: 44.79%
- Adjacent Misclassifications: 874 (62.3%)
- Non-Adjacent Misclassifications: 530 (37.7%)

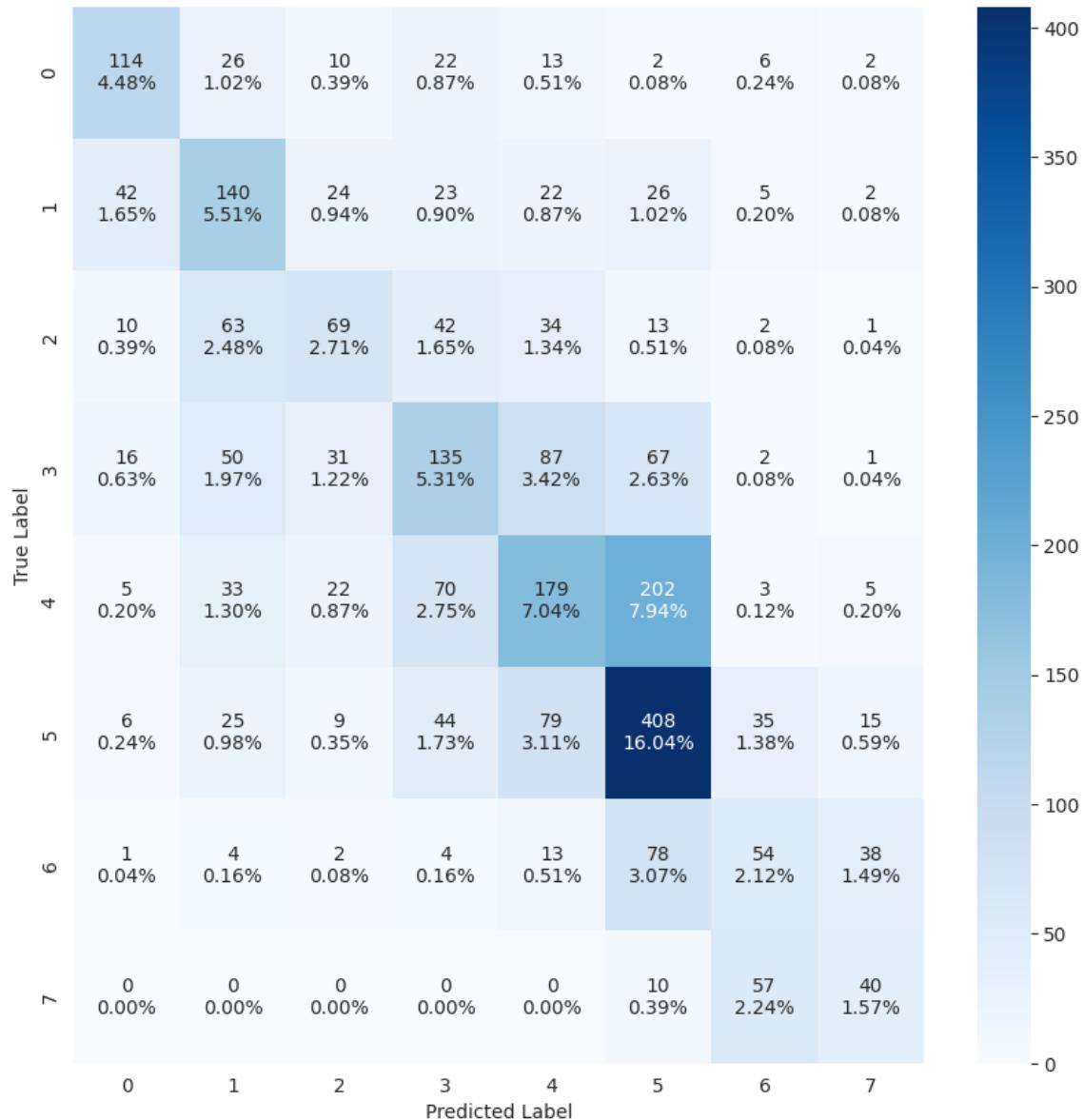


Figure 5.8: Confusion Matrix for the XGBoost model.

- GRU-GCNLSTM (multi-modal):

- Accuracy: 73.99%
- Adjacent Misclassifications: 361 (62.4%)
- Non-Adjacent Misclassifications: 218 (37.6%)

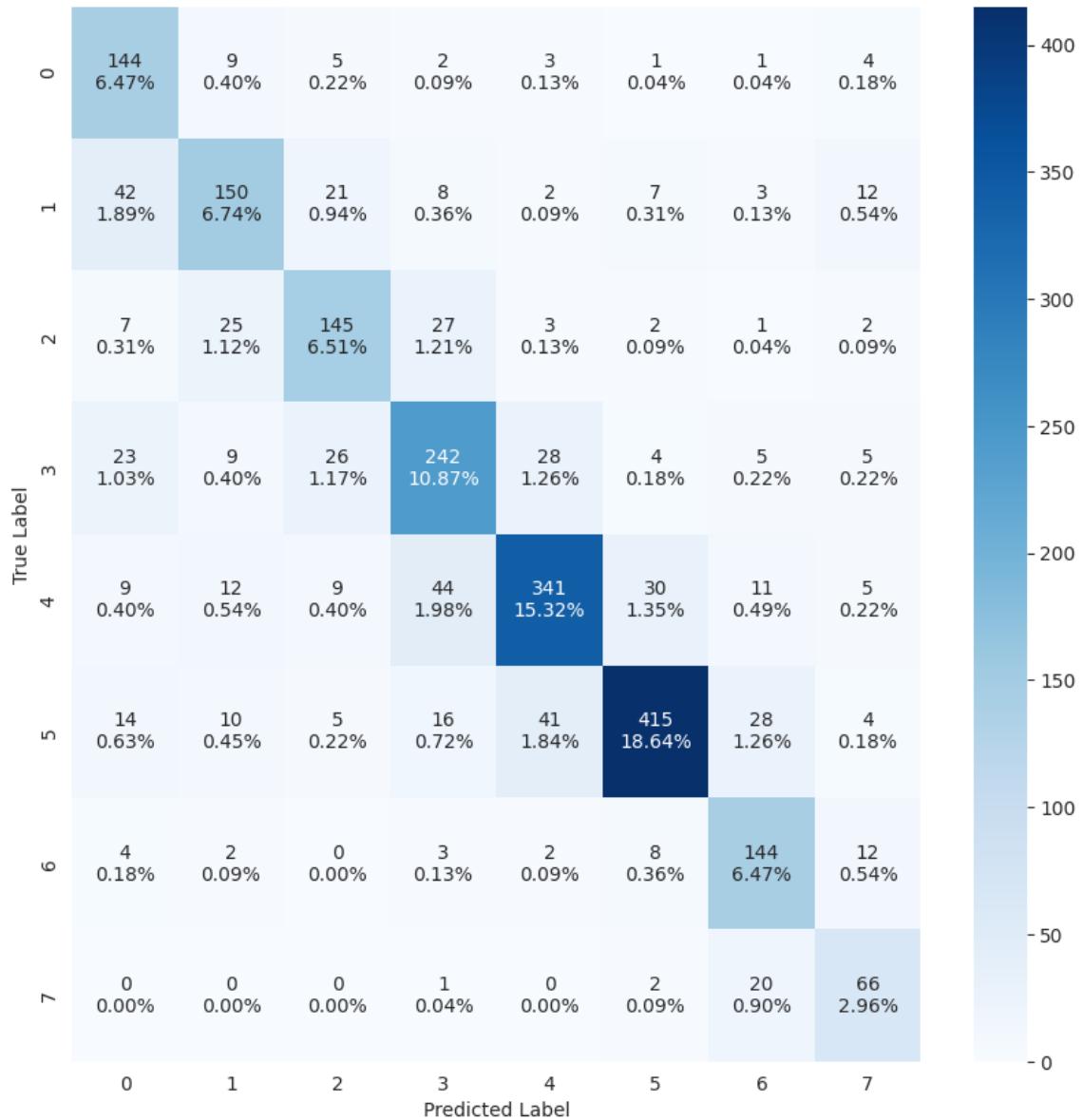


Figure 5.9: Confusion Matrix for the GRU-GCNLSTM model.

- TRA-GCNGRU (multi-modal):

- Accuracy: 36.07%
- Adjacent Misclassifications: 1000 (70.1%)
- Non-Adjacent Misclassifications: 423 (29.9%)

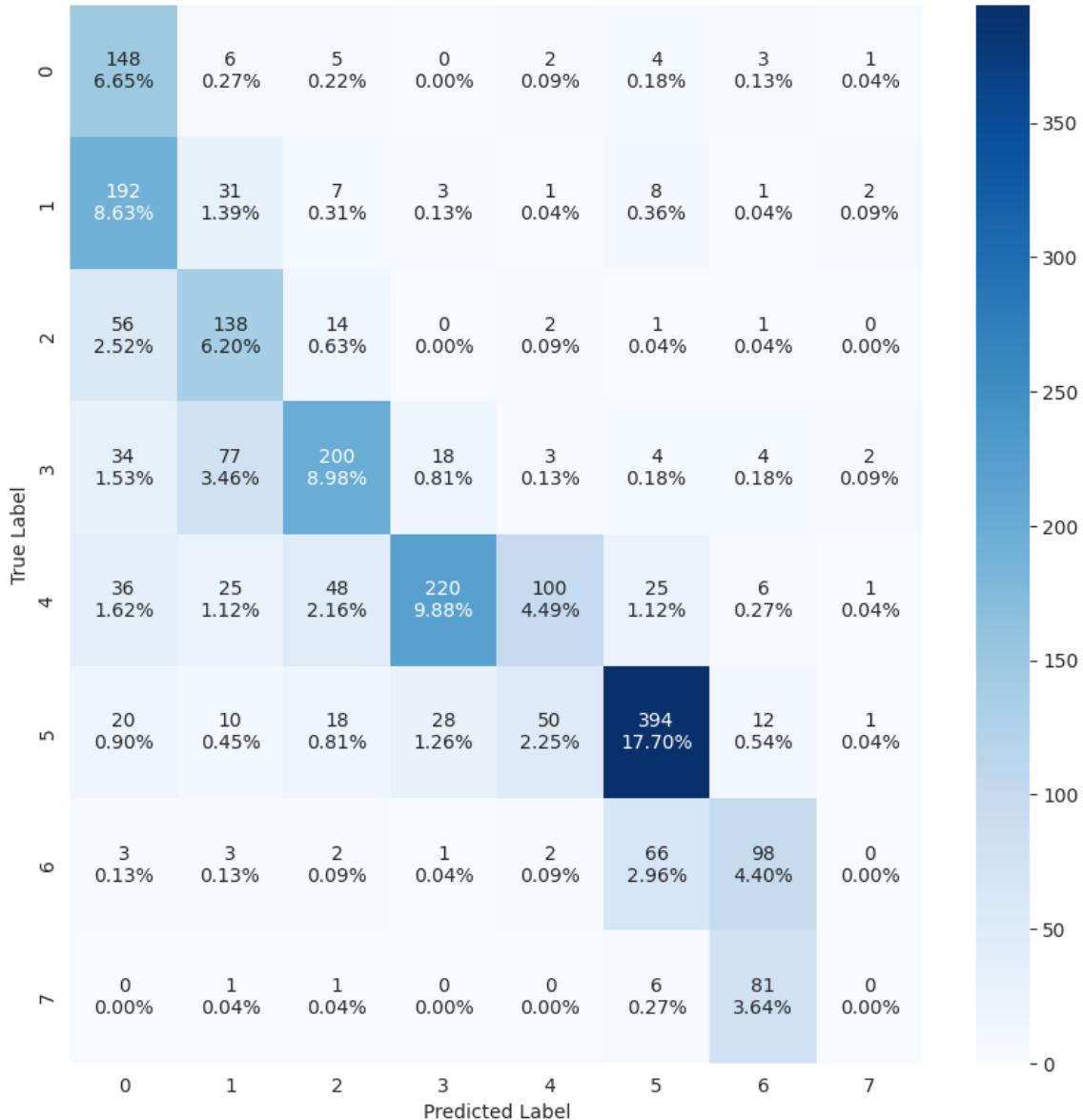


Figure 5.10: Confusion Matrix for the TRA-GCNGRU model.

- **LSTM-GCNGRU (multi-modal):**

- Accuracy: 74.03%
- Adjacent Misclassifications: 366 (63.3%)
- Non-Adjacent Misclassifications: 212 (36.7%)

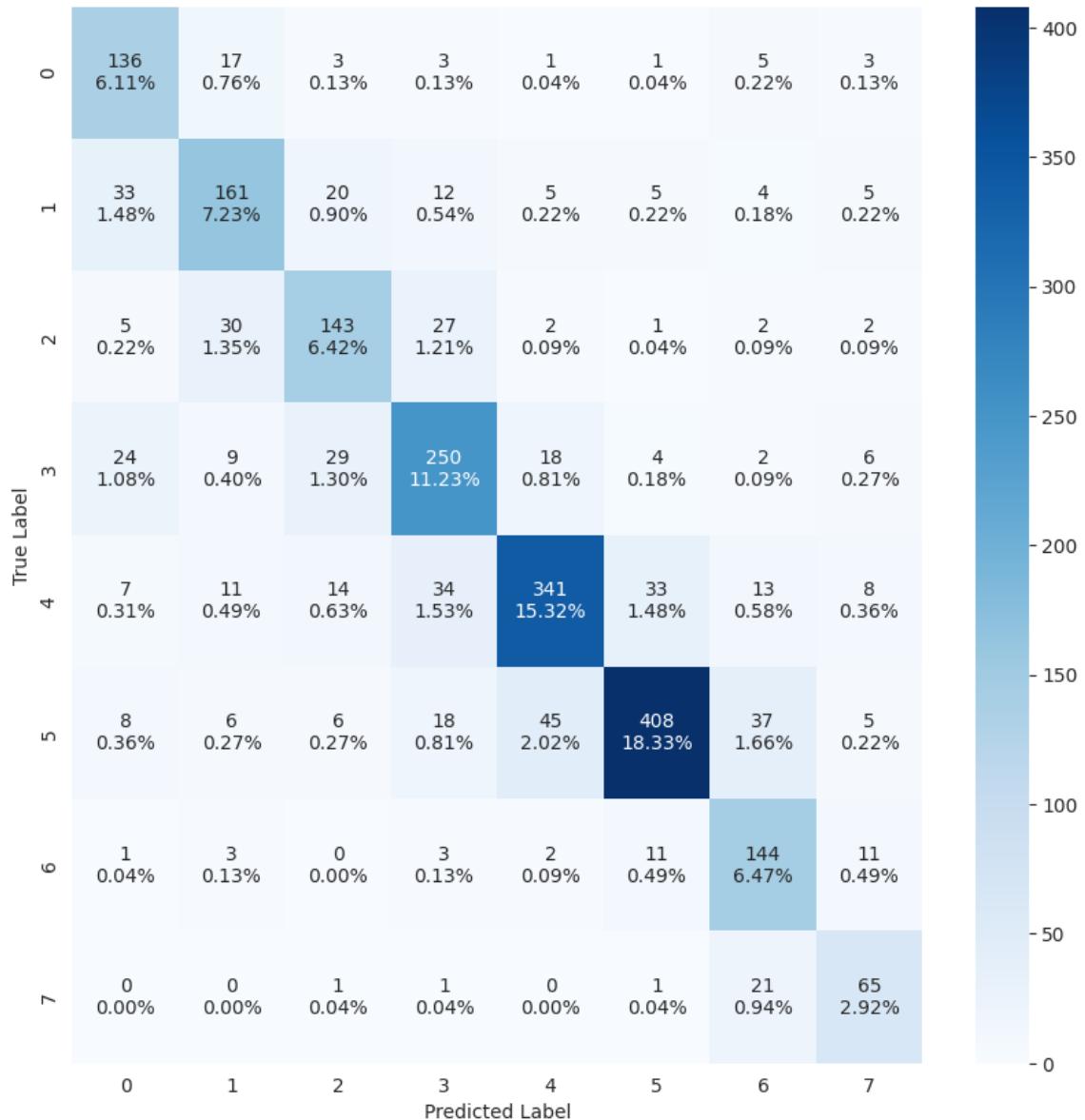


Figure 5.11: Confusion Matrix for the LSTM-GCNGRU model.

- GRU-GCNGRU (multi-modal):

- Accuracy: 74.44%
- Adjacent Misclassifications: 370 (65.0%)
- Non-Adjacent Misclassifications: 199 (35.0%)

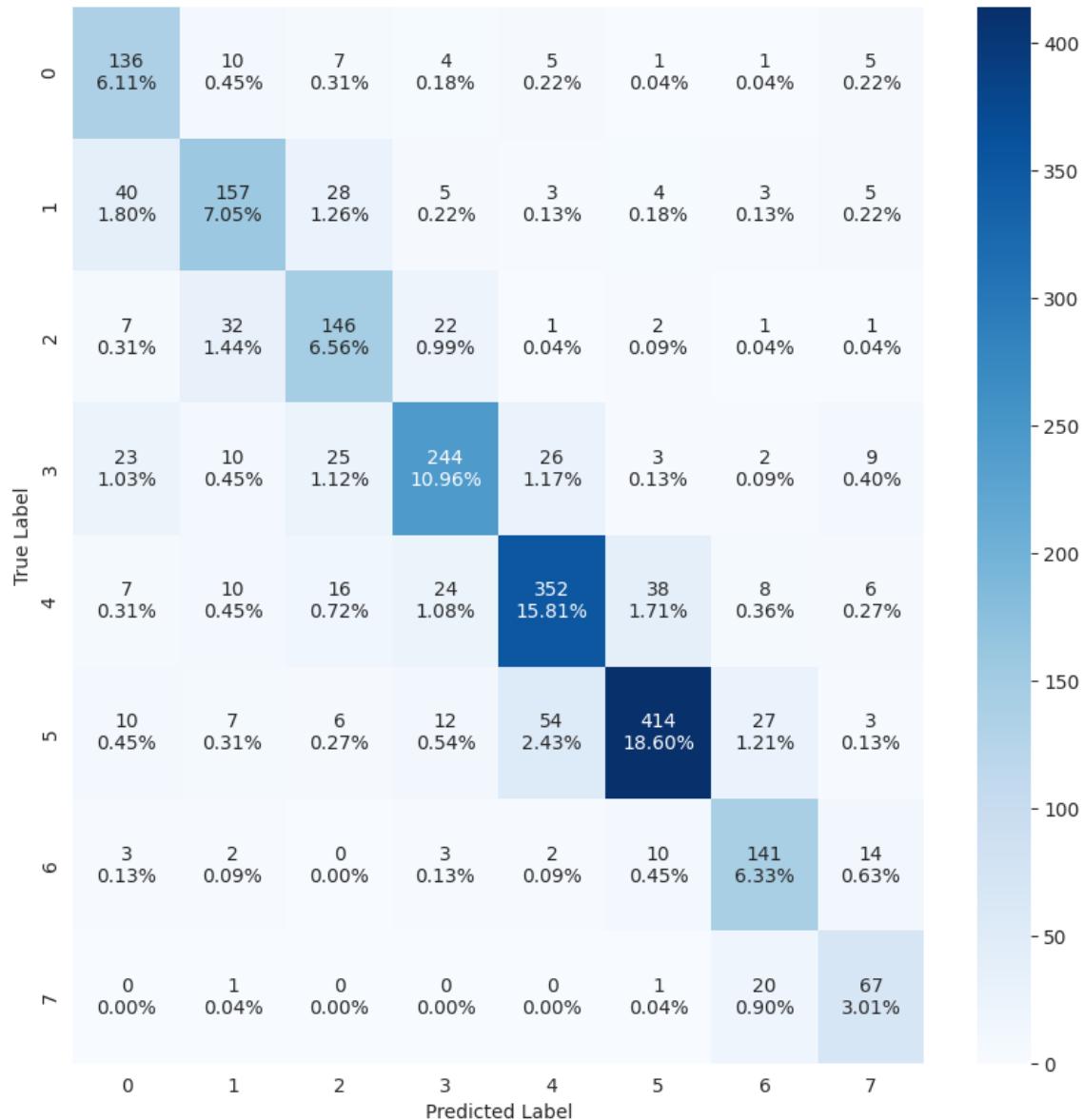


Figure 5.12: Confusion Matrix for the GRU-GCNGRU model.

- **LSTM-GCNLSTM (multi-modal):**

- Accuracy: 72.96%
- Adjacent Misclassifications: 387 (64.3%)
- Non-Adjacent Misclassifications: 215 (35.7%)

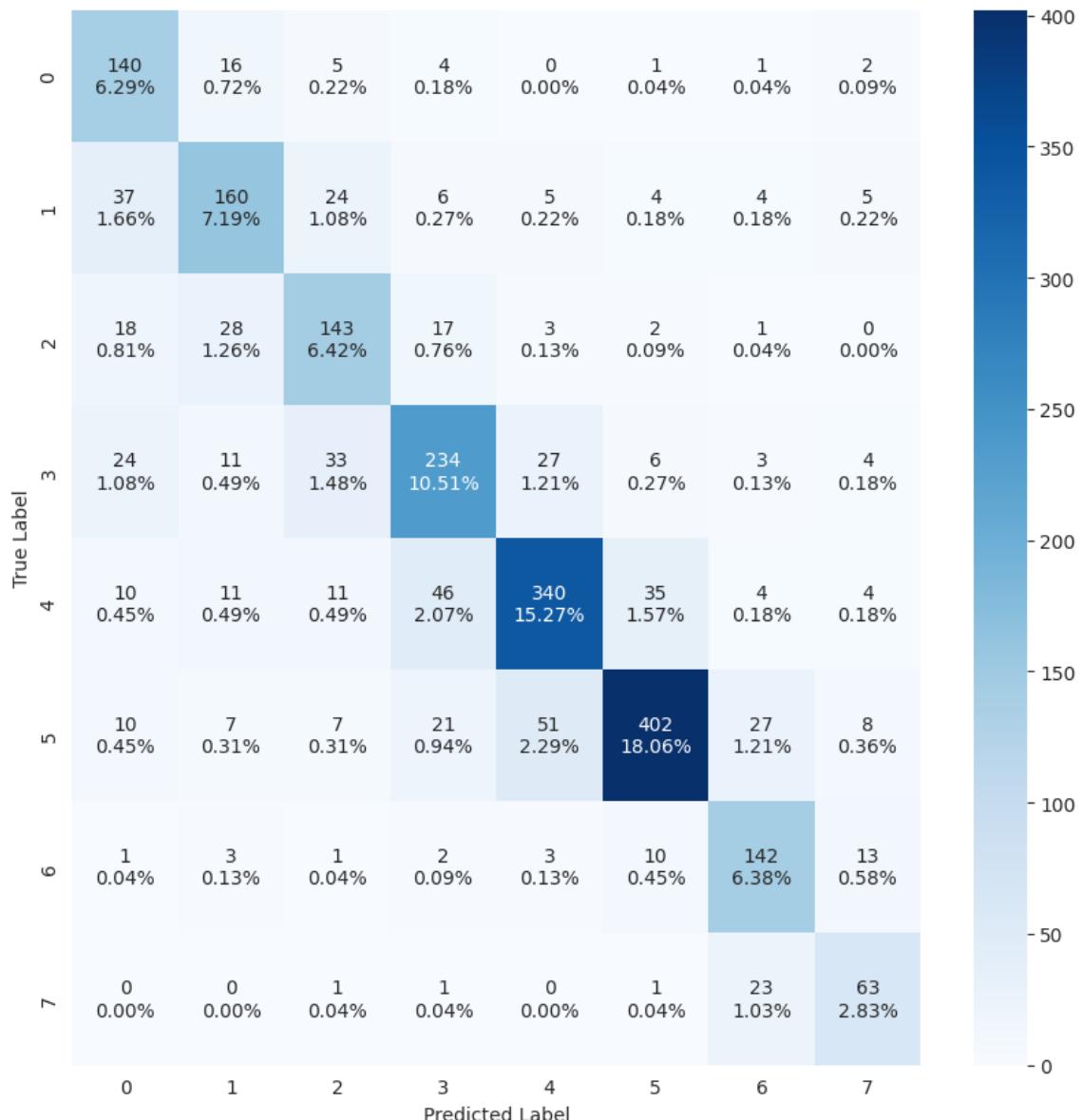


Figure 5.13: Confusion Matrix for the LSTM-GCNLSTM model.

Table 5.5: Model Prediction Summary

Model	Accuracy	Adjacent	Non-adjacent
GCNGRU	37.92%	817	565
GCNLSTM	43.80%	737	514
LSTM	73.76%	391	193
GRU	75.25%	368	183
Transformer	31.04%	703	832
XGBOOST	44.79%	874	530
GRU-GCNLSTM	73.99%	361	218
TRA-GCNGRU	36.07%	1000	423
LSTM-GCNGRU	74.03%	366	212
GRU-GCNGRU	74.44%	370	199
LSTM-GCNLSTM	72.96%	387	215

Table 5.5 summarizes the predictions of all the models. It includes the model name, overall accuracy, the number of adjacent and non-adjacent misclassifications for each model.

### 5.5.1.1 Baseline Models

Models based on numerical data, such as GRU and LSTM, generally achieve higher accuracy compared to those based on graph data like GCNGRU and GCNLSTM. Specifically, the GRU and LSTM models achieved an accuracy of 75.25% and 73.76%, respectively. In contrast, the GCNGRU and GCNLSTM models had accuracies of 37.92% and 43.8%, respectively. This indicates a clear advantage for numerical data models in terms of accuracy.

- **Highest Accuracy:** GRU (75.25%)
- **Lowest Non-Adjacent Misclassifications:** GRU with 183 (33.2%) non-adjacent misclassifications

XGBoost, another baseline numerical model, had a lower accuracy of 44.79% with 530 (37.7%) non-adjacent misclassifications. This suggests that while XGBoost is a powerful tool, it underperformed compared to LSTM and GRU in this context.

### 5.5.1.2 Multi-modal Models

When comparing the multi-modal models, which integrate both numerical and graph data, we observe that these models generally match the accuracy of the baseline models. Among the multi-modal models, the GRU-GCNGRU model had the highest accuracy at 74.44%, closely followed by LSTM-GCNGRU at 74.03%. The best multi-modal model in terms of lowest amount of non-adjacent misclassifications was GRU-GCNGRU, or 199.

- **Highest Accuracy:** GRU-GCNGRU (74.44%)
- **Lowest Non-Adjacent Misclassifications:** GRU-GCNGRU with 199 (35.0%) non-adjacent misclassifications

### 5.5.2 Performance Over Time

Another interesting aspect to examine is the performance of models over different time frames in the test set. This section analyzes the accuracies and misclassifications of the best performing baseline model, GRU, and the best performing multi-modal model, GRU-GCNGRU, over time. The test set spans from the first quarter of 2019 to the last quarter of 2020, covering a total of eight quarters.

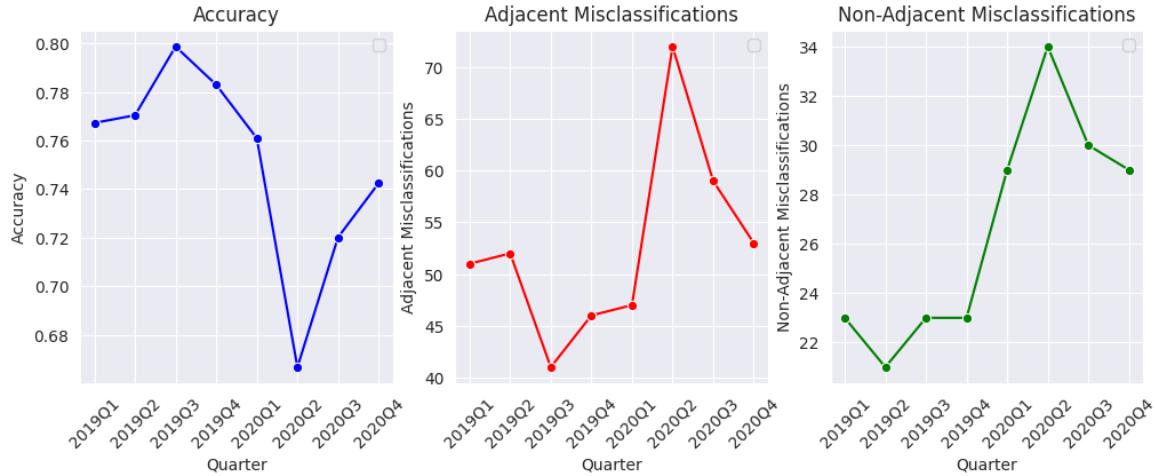


Figure 5.14: Accuracy and misclassifications over time for the GRU model.

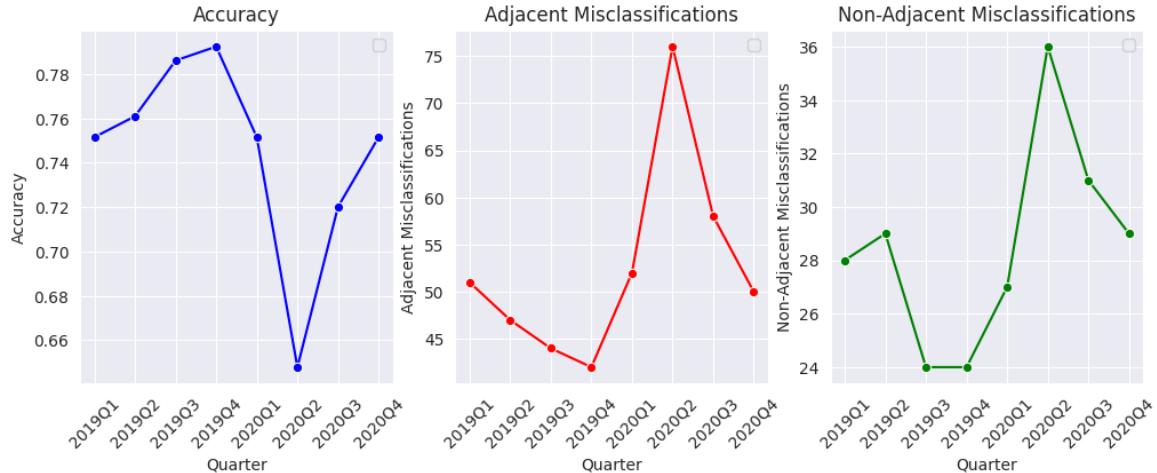


Figure 5.15: Accuracy and misclassifications over time for the GRU-GCNGRU model.

Figure 5.14 and Figure 5.15 reveal similar patterns in accuracy and misclassifications for the GRU and GRU-GCNGRU models over time, with consistent results for both seen and unseen companies. Both models experience a significant drop in accuracy in the last quarter of 2019, with an improvement starting in late 2020. This trend is likely due to the COVID-19 pandemic and its unprecedented impact on global financial markets. Since the training data, spanning from the first quarter of 2010 to the last quarter of 2018, did not include any financial crisis scenarios, the models were not exposed to patterns comparable to those caused by COVID-19. This analysis indicates that the pandemic affected our test results. This is to be expected as the pandemic is a ‘black swan event’, which is not predictable by past data. Drift in that

time would be expected and it is so shown in the data. However, and as the plot shows, this drift is quickly replaced by satisfactory performance after two quarters, showing that, even if the pandemic did have a drift event that was not captured by the model, after a semester the patterns that we had originally captured correctly described the post-pandemic shock situation.

In conclusion, the error analysis reveals that RNN baseline models, particularly LSTM and GRU, achieve higher accuracy compared to RGNN baseline models and are on par with the best multi-modal models. The multi-modal models demonstrate comparable accuracy while maintaining a balanced rate of misclassifications. Although adjacent misclassifications are common across all models, reflecting the ordinal classification task, non-adjacent misclassifications are slightly lower in the best numerical models. These findings suggest that while numerical data models are effective, multi-modal models offer a robust alternative by combining the strengths of both data types. It is important to note that the COVID-19 period skewed our test results, impacting the models' performance. The pandemic's unprecedented effect on financial markets introduced patterns not present in the training data, highlighting the need for models to be trained on diverse scenarios to improve their robustness.



# Chapter 6

## Conclusion

This research successfully engineers a multi-modal temporal network model to enhance corporate credit rating prediction. By using a multi-modal approach that combines graph-based and sequential data processing, the study demonstrates significant improvements in predicting corporate financial health. The models, trained on data from 318 US companies, effectively capture intricate financial patterns through the synthesis of numerical data from quarterly financial statements and graph data from stock market interactions. The effectiveness of these models is validated on both the training data and an additional set of 112 "unseen" companies, showcasing their robust generalization capabilities. This indicates that the multi-modal models are well-suited for practical deployment in financial risk assessment, offering a more accurate and comprehensive tool for predicting corporate credit risk.

The primary objective of this research was to develop an advanced model that improves corporate credit rating prediction by leveraging both numerical and graph data. The implementation of RNNs combined with GNNs aimed to address the complexity and dynamic nature of financial data. By achieving high performance metrics and demonstrating robust generalization, this study successfully meets this objective, showcasing the potential of multi-modal models in providing a nuanced understanding of corporate financial health.

In terms of the main takeaways from the sub-objectives, this research highlights the significant benefits of integrating RNNs with GNNs. The developed multi-modal models leverage the strengths of both graph-based and sequential data, resulting in enhanced predictive performance and robustness. The analysis demonstrates that models like LSTM-GCNLSTM outperform standalone single-modality machine learning models, particularly in handling dynamic and complex data structures. Additionally, the performance of these models on unseen data underscores their ability to generalize well, which is crucial in evolving financial environments where new entities and conditions frequently emerge.

The error analysis shows that RNN baseline models, like LSTM and GRU, achieve higher accuracy compared to RGNN baseline models and are comparable to the best multi-modal models. Multi-modal models demonstrate not only similar accuracy but also maintain a balanced rate of misclassifications. Adjacent misclassifications are common across all models, reflecting the ordinal nature of the classification task, while non-adjacent misclassifications are notably lower in the best numerical models. This indicates that while numerical data models are effective, multi-modal models offer a robust alternative by combining the strengths of both data types. Additionally, the

impact of the COVID-19 pandemic on the test results highlights the importance of training models on diverse scenarios to improve their robustness.

Future research should address and further minimize non-adjacent misclassifications, as these errors can significantly impact credit risk assessment. Refining model architectures, exploring a broader range of machine learning models, and incorporating additional data sources may help achieve this goal. Moreover, extending the applicability of these multi-modal techniques to other financial prediction tasks could enhance their utility. It is also crucial for models to be trained on diverse scenarios to improve their robustness, as demonstrated by the impact COVID-19 had on our results. By continuing to innovate and enhance these methods, there is significant potential to advance financial risk management, leading to more informed decision-making and greater stability within the financial system.

# Bibliography

- [1] S. Poor's, *Guide to Credit Rating Essentials*. Standard Poor's, 2022.
- [2] S. Poor's, *Corporate Ratings Criteria*. Standard Poor's, 2001.
- [3] E. Benmelech and J. Dlugosz, "The credit rating crisis", *NBER Macroeconomics Annual*, vol. 24, no. 1, pp. 161–208, Jan. 2010, ISSN: 1537-2642. DOI: 10.1086/648293. [Online]. Available: <http://dx.doi.org/10.1086/648293>.
- [4] S. Shi, R. Tse, W. Luo, S. D'Addona, and G. Pau, "Machine learning-driven credit risk: A systemic review", *Neural Computing and Applications*, vol. 34, no. 17, pp. 14327–14339, Jul. 2022, ISSN: 1433-3058. DOI: 10.1007/s00521-022-07472-2. [Online]. Available: <http://dx.doi.org/10.1007/s00521-022-07472-2>.
- [5] A. Niu, B. Cai, and S. Cai, "Big data analytics for complex credit risk assessment of network lending based on smote algorithm", *Complexity*, vol. 2020, pp. 1–9, Sep. 2020, ISSN: 1099-0526. DOI: 10.1155/2020/8563030. [Online]. Available: <http://dx.doi.org/10.1155/2020/8563030>.
- [6] L. Marceau, L. Qiu, N. Vandewiele, and E. Charton, *A comparison of deep learning performances with other machine learning algorithms on credit scoring unbalanced data*, 2020. arXiv: 1907.12363 [cs.LG].
- [7] C.-C. Yeh, F. Lin, and C.-Y. Hsu, "A hybrid kmv model, random forests and rough set theory approach for credit rating", *Knowledge-Based Systems*, vol. 33, pp. 166–172, 2012, ISSN: 0950-7051. DOI: <https://doi.org/10.1016/j.knosys.2012.04.004>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S095070511200086X>.
- [8] C. Bravo, R. Calabrese, S. Lessmann, C. Mues, and M. Óskarsdóttir, "Credit risk and artificial intelligence: On the need for convergent regulation", *SSRN Electronic Journal*, 2023, ISSN: 1556-5068. DOI: 10.2139/ssrn.4615412. [Online]. Available: <http://dx.doi.org/10.2139/ssrn.4615412>.
- [9] S. Bhatore, L. Mohan, and Y. R. Reddy, "Machine learning techniques for credit risk evaluation: A systematic literature review", *Journal of Banking and Financial Technology*, vol. 4, no. 1, pp. 111–138, Apr. 2020, ISSN: 2524-7964. DOI: 10.1007/s42786-020-00020-3. [Online]. Available: <http://dx.doi.org/10.1007/s42786-020-00020-3>.
- [10] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019. DOI: 10.1109/TPAMI.2018.2798607.
- [11] R. Dolphin, B. Smyth, and R. Dong, *A multimodal embedding-based approach to industry classification in financial markets*, 2022. arXiv: 2211.06378 [q-fin.ST].

- [12] M. Tavakoli, R. Chandra, F. Tian, and C. Bravo, *Multi-modal deep learning for credit rating prediction using text and numerical data streams*, 2023. arXiv: 2304.10740 [q-fin.GN].
- [13] Z. C. Lipton, J. Berkowitz, and C. Elkan, *A critical review of recurrent neural networks for sequence learning*, 2015. arXiv: 1506.00019 [cs.LG].
- [14] G. Tsang, J. Deng, and X. Xie, “Recurrent neural networks for financial time-series modelling”, in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 892–897. DOI: 10.1109/ICPR.2018.8545666.
- [15] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory”, *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: <https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf>. [Online]. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [16] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, *On the properties of neural machine translation: Encoder-decoder approaches*, 2014. arXiv: 1409.1259 [cs.CL].
- [17] J. Zhou, G. Cui, S. Hu, *et al.*, “Graph neural networks: A review of methods and applications”, *AI Open*, vol. 1, pp. 57–81, 2020, ISSN: 2666-6510. DOI: <https://doi.org/10.1016/j.aiopen.2021.01.001>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666651021000012>.
- [18] Y. Wu, D. Lian, Y. Xu, L. Wu, and E. Chen, “Graph convolutional networks with markov random field reasoning for social spammer detection”, in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 1054–1061.
- [19] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, *et al.*, “Graph networks as learnable physics engines for inference and control”, in *International conference on machine learning*, PMLR, 2018, pp. 4470–4479.
- [20] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, *et al.*, “Interaction networks for learning about objects, relations and physics”, *Advances in neural information processing systems*, vol. 29, 2016.
- [21] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, “Protein interface prediction using graph convolutional networks”, *Advances in neural information processing systems*, vol. 30, 2017.
- [22] T. Hamaguchi, H. Oiwa, M. Shimbo, and Y. Matsumoto, “Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach”, *arXiv preprint arXiv:1706.05674*, 2017.
- [23] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, 2017. arXiv: 1609.02907 [cs.LG].
- [24] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph attention networks*, 2018. arXiv: 1710.10903 [stat.ML].
- [25] P. Cui, X. Wang, J. Pei, and W. Zhu, “A survey on network embedding”, *IEEE transactions on knowledge and data engineering*, vol. 31, no. 5, pp. 833–852, 2018.

- [26] D. Zhang, C.-Y. Chow, A. Liu, X. Zhang, Q. Ding, and Q. Li, “Efficient evaluation of shortest travel-time path queries through spatial mashups”, *GeoInformatica*, vol. 22, no. 1, pp. 3–28, Jan. 2017, ISSN: 1573-7624. DOI: 10.1007/s10707-016-0288-4. [Online]. Available: <http://dx.doi.org/10.1007/s10707-016-0288-4>.
- [27] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: Problems, techniques, and applications”, *IEEE transactions on knowledge and data engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.
- [28] P. Goyal and E. Ferrara, “Graph embedding techniques, applications, and performance: A survey”, *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.
- [29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space”, *arXiv preprint arXiv:1301.3781*, 2013.
- [30] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations”, in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.
- [31] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks”, in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.
- [32] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding”, in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.
- [33] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, “Network representation learning with rich text information.”, in *IJCAI*, vol. 2015, 2015, pp. 2111–2117.
- [34] W. L. Hamilton, R. Ying, and J. Leskovec, “Representation learning on graphs: Methods and applications”, *arXiv preprint arXiv:1709.05584*, 2017.
- [35] Q. Cappart, D. Chételat, E. Khalil, A. Lodi, C. Morris, and P. Veličković, *Combinatorial optimization and reasoning with graph neural networks*, 2022. arXiv: 2102.09544 [cs.LG].
- [36] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021. DOI: 10.1109/TNNLS.2020.2978386.
- [37] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric deep learning: Going beyond euclidean data”, *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, Jul. 2017, ISSN: 1558-0792. DOI: 10.1109/msp.2017.2693418. [Online]. Available: <http://dx.doi.org/10.1109/MSP.2017.2693418>.
- [38] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, “Graph convolutional networks: A comprehensive review”, *Computational Social Networks*, vol. 6, no. 1, Nov. 2019, ISSN: 2197-4314. DOI: 10.1186/s40649-019-0069-y. [Online]. Available: <http://dx.doi.org/10.1186/s40649-019-0069-y>.

- [39] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, “Simplifying graph convolutional networks”, in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, PMLR, Sep. 2019, pp. 6861–6871. [Online]. Available: <https://proceedings.mlr.press/v97/wu19e.html>.
- [40] H. Gao, Y. Chen, and S. Ji, *Learning graph pooling and hybrid convolutional operations for text representations*, 2019. arXiv: 1901.06965 [cs.AI].
- [41] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks”, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649. DOI: 10.1109/ICASSP.2013.6638947.
- [42] T. Mikolov and G. Zweig, “Context dependent recurrent neural network language model”, in *2012 IEEE Spoken Language Technology Workshop (SLT)*, 2012, pp. 234–239. DOI: 10.1109/SLT.2012.6424228.
- [43] Y. Zhang, D. Wang, Y. Chen, H. Shang, and Q. Tian, “Credit risk assessment based on long short-term memory model”, in *Intelligent Computing Theories and Application*, D.-S. Huang, K.-H. Jo, and J. C. Figueira-García, Eds., Cham: Springer International Publishing, 2017, pp. 700–712, ISBN: 978-3-319-63312-1.
- [44] C. Wang, D. Han, Q. Liu, and S. Luo, “A deep learning approach for credit scoring of peer-to-peer lending using attention mechanism lstm”, *IEEE Access*, vol. 7, pp. 2161–2168, 2018.
- [45] J. Jurgošky, M. Granitzer, K. Ziegler, et al., “Sequence classification for credit-card fraud detection”, *Expert systems with applications*, vol. 100, pp. 234–245, 2018.
- [46] D. Babaev, M. Savchenko, A. Tuzhilin, and D. Umerenkov, “E.t.-rnn: Applying deep learning to credit loan applications”, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, ser. KDD ’19, ACM, Jul. 2019. DOI: 10.1145/3292500.3330693. [Online]. Available: <http://dx.doi.org/10.1145/3292500.3330693>.
- [47] M. Ala’raj, M. F. Abbod, and M. Majdalawieh, “Modelling customers credit card behaviour using bidirectional lstm neural networks”, *Journal of Big Data*, vol. 8, no. 1, May 2021, ISSN: 2196-1115. DOI: 10.1186/s40537-021-00461-7. [Online]. Available: <http://dx.doi.org/10.1186/s40537-021-00461-7>.
- [48] B. Chen and S. Long, “A novel end-to-end corporate credit rating model based on self-attention mechanism”, *IEEE Access*, vol. 8, pp. 203 876–203 889, 2020, ISSN: 2169-3536. DOI: 10.1109/access.2020.3036469. [Online]. Available: <http://dx.doi.org/10.1109/access.2020.3036469>.
- [49] J. Skarding, B. Gabrys, and K. Musial, “Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey”, *IEEE Access*, vol. 9, pp. 79 143–79 168, 2021. DOI: 10.1109/ACCESS.2021.3082932.
- [50] J. Li, Z. Han, H. Cheng, et al., “Predicting path failure in time-evolving graphs”, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD ’19, Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 1279–1289, ISBN: 9781450362016. DOI: 10.1145/3292500.3330847. [Online]. Available: <https://doi.org/10.1145/3292500.3330847>.

- [51] H. Peng, H. Wang, B. Du, *et al.*, “Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting”, *Information Sciences*, vol. 521, pp. 277–290, 2020, ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2020.01.043>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020025520300451>.
- [52] E. Tiukhova, E. Penaloza, M. Óskarsdóttir, B. Baesens, M. Snoeck, and C. Bravo, “Inflect-dggn: Influencer prediction with dynamic graph neural networks”, *arXiv preprint arXiv:2307.08131*, 2023.
- [53] Y. Wang, P. Li, C. Bai, V. Subrahmanian, and J. Leskovec, “Generic representation learning for dynamic social interaction”, in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining Workshop*, 2020.
- [54] S. M. Kazemi, R. Goel, K. Jain, *et al.*, *Representation learning for dynamic graphs: A survey*, 2020. arXiv: 1905.11485 [cs.LG].
- [55] Y. Xie, C. Li, B. Yu, C. Zhang, and Z. Tang, *A survey on dynamic network embedding*, 2020. arXiv: 2006.08093 [cs.SI].
- [56] C. D. T. Barros, M. R. F. Mendonça, A. B. Vieira, and A. Ziviani, *A survey on embedding dynamic graphs*, 2021. arXiv: 2101.01229 [cs.LG].
- [57] K. Korangi, C. Mues, and C. Bravo, *Large-scale portfolio optimisation using graph attention networks*, 2023. arXiv: to appear [q-fin.GN].
- [58] D. Cheng, F. Yang, S. Xiang, and J. Liu, “Financial time series forecasting with multi-modality graph neural network”, *Pattern Recognition*, vol. 121, p. 108218, 2022, ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2021.108218>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S003132032100399X>.
- [59] W. Zhang, S. Yan, J. Li, X. Tian, and T. Yoshida, “Credit risk prediction of smes in supply chain finance by fusing demographic and behavioral data”, *Transportation Research Part E: Logistics and Transportation Review*, vol. 158, p. 102611, 2022, ISSN: 1366-5545. DOI: <https://doi.org/10.1016/j.tre.2022.102611>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1366554522000096>.
- [60] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2019, vol. 793.
- [61] S. van Buuren and K. Groothuis-Oudshoorn, “Mice: Multivariate imputation by chained equations in r”, *Journal of Statistical Software*, vol. 45, no. 3, pp. 1–67, 2011. DOI: 10.18637/jss.v045.i03. [Online]. Available: <https://www.jstatsoft.org/index.php/jss/article/view/v045i03>.
- [62] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, “Scikit-learn: Machine learning in python”, *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
- [63] G. P. Massara, T. Di Matteo, and T. Aste, “Network Filtering for Big Data: Triangulated Maximally Filtered Graph”, *Journal of Complex Networks*, vol. 5, no. 2, pp. 161–178, Jun. 2016, ISSN: 2051-1310. DOI: 10.1093/comnet/cnw015. eprint: <https://academic.oup.com/comnet/article-pdf/5/2/161/13794756/cnw015.pdf>. [Online]. Available: <https://doi.org/10.1093/comnet/cnw015>.

- [64] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL].
- [65] S. Y. Boulahia, A. Amamra, M. R. Madi, and S. Daikh, “Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition”, *Machine Vision and Applications*, vol. 32, no. 6, Sep. 2021, ISSN: 1432-1769. DOI: 10.1007/s00138-021-01249-8. [Online]. Available: <http://dx.doi.org/10.1007/s00138-021-01249-8>.
- [66] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].
- [67] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, 2019. arXiv: 1711.05101 [cs.LG].
- [68] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult”, *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994. DOI: 10.1109/72.279181.
- [69] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves”, in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML ’06, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 233–240, ISBN: 1595933832. DOI: 10.1145/1143844.1143874. [Online]. Available: <https://doi.org/10.1145/1143844.1143874>.
- [70] M. Óskarsdóttir, W. Ahmed, K. Antonio, *et al.*, “Social network analytics for supervised fraud detection in insurance”, *Risk Analysis*, vol. 42, no. 8, pp. 1872–1890, 2022.
- [71] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system”, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16, San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794, ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. [Online]. Available: <https://doi.org/10.1145/2939672.2939785>.

# Appendix A

## Numerical Features

Feature	Description
pastrating	Credit rating (target label).
curncdq	Currency code for the financial data.
costat	Company status (e.g., active, inactive).
fic_CHE	Flag indicating if the company is in Switzerland.
fic_IRL	Flag indicating if the company is in Ireland.
fic_USA	Flag indicating if the company is in the USA.
ajexq	Adjustment for stock splits and dividends, quarterly.
ajpq	Adjustment for prior periods, quarterly.
acomincq	Accumulated other comprehensive income, quarterly.
acoq	Accounts receivable, net of allowances, quarterly.
anoq	Total assets not otherwise classified, quarterly.
aociderglq	Accumulated other comprehensive income due to derivative gains/losses, quarterly.
aociotherq	Other accumulated comprehensive income, quarterly.
aocipenq	Accumulated other comprehensive income due to pension gains/losses, quarterly.
aocisecglq	Accumulated other comprehensive income due to securities gains/losses, quarterly.
aol2q	Level 2 assets and liabilities, quarterly.
aoq	Total assets and other obligations, quarterly.
apq	Accounts payable, quarterly.
aqpl1q	Assets and liabilities at Level 1, quarterly.
atq	Total assets, quarterly.
aul3q	Level 3 assets and liabilities, quarterly.
capsq	Capital surplus, quarterly.
ceqq	Common equity, quarterly.
cheq	Cash and cash equivalents, quarterly.
chq	Cash holdings, quarterly.
cibegniq	Comprehensive income, beginning of the period, quarterly.
cicurrq	Comprehensive income, current period, quarterly.
ciderglq	Comprehensive income due to derivative gains/losses, quarterly.
cimiiq	Comprehensive income, minority interest, quarterly.
ciotherq	Other comprehensive income, quarterly.

Feature	Description
cipenq	Comprehensive income due to pension gains/losses, quarterly.
ciq	Comprehensive income, quarterly.
cisecglq	Comprehensive income due to securities gains/losses, quarterly.
citotalq	Total comprehensive income, quarterly.
cogsq	Cost of goods sold, quarterly.
csh12q	Common shares outstanding, 12 months, quarterly.
cshfd12	Common shares fully diluted, 12 months.
cshfdq	Common shares fully diluted, quarterly.
cshiq	Common shares issued, quarterly.
cshopq	Common shares options, quarterly.
cshoq	Common shares outstanding, quarterly.
cshprq	Common shares preferred, quarterly.
cstkeq	Common stock equity, quarterly.
cstkq	Common stock, quarterly.
dcomq	Deferred Compensation, quarterly.
diladq	Dilution adjustments, quarterly.
dilavq	Dilution available, excluding, quarterly.
dlcq	Debt, current liabilities, quarterly.
dlttq	Debt, long-term, total, quarterly.
doq	Debt, other, quarterly.
dpq	Depreciation, quarterly.
dvpq	Dividends paid, quarterly.
epsf12	Earnings per share, fully diluted, 12 months.
epsf12	Earnings per share, fully diluted, before extraordinary items, 12 months.
epsfiq	Earnings per share, fully diluted, before extraordinary items, quarterly.
epsfxq	Earnings per share, fully diluted, excluding extraordinary items.
epspi12	Earnings per share, primary, before extraordinary items, 12 months.
epspiq	Earnings per share, primary, before extraordinary items, quarterly.
epspxq	Earnings per share, primary, excluding extraordinary items, quarterly.
epsx12	Earnings per share, excluding extraordinary items, 12 months.
esopctq	ESOP contribution, quarterly.
esopnrq	ESOP non-redeemable, quarterly.
esoptq	ESOP total, quarterly.
gdwlq	Goodwill, quarterly.
ibadjq	Income before extraordinary items, adjusted, quarterly.
ibcomq	Income before extraordinary items, common shares, quarterly.
ibq	Income before extraordinary items, quarterly.
icaptq	Invested capital, quarterly.
intanq	Intangible assets, quarterly.
invtq	Inventory, quarterly.
ivstq	Investments in short-term securities, quarterly.
lcoq	Liabilities, current other, quarterly.
lnoq	Liabilities, non-current other, quarterly.
lol2q	Level 2 liabilities, quarterly.
loq	Liabilities, other, quarterly.
lpql1q	Level 1 liabilities, quarterly.
lseq	Liabilities and stockholder equity, quarterly.

Feature	Description
ltmibq	Liabilities, noncontrolling interest, total, quarterly.
ltq	Liabilities, total, quarterly.
lul3q	Level 3 liabilities, quarterly.
mibnq	Minority interest, beginning of the period, quarterly.
mibq	Minority interest, quarterly.
mibtq	Minority interest, total, quarterly.
msaq	Marketable securities adjustments, quarterly.
niq	Net income, quarterly.
nopiq	Non-operating income, total, quarterly.
oepf12	Operating earnings per share, fully diluted, 12 months.
oops12	Operating earnings per share, 12 months.
oepsxq	Operating earnings per share, excluding extraordinary items, quarterly.
oiadpq	Operating income after depreciation, quarterly.
oibdpq	Operating income before depreciation, quarterly.
opepsq	Operating earnings per share, quarterly.
piq	Pretax income, quarterly.
pnrshoq	Preferred shares outstanding, quarterly.
ppentq	Property, plant, and equipment, net, quarterly.
prcraq	Repurchase price, average per share, quarterly.
prshoq	Preferred shares outstanding, quarterly.
pstkknq	Preferred stock, nonredeemable, quarterly.
pstkq	Preferred stock, total, quarterly.
pstkrq	Preferred stock, redeemable, quarterly.
rdipaq	Research and development expenses, paid, quarterly.
rdipq	Research and development expenses, quarterly.
rectaq	Receivables, cumulative translation adjustments, quarterly.
rectq	Receivables, total, quarterly.
req	Retained earnings, quarterly.
revtq	Revenue, total, quarterly.
saleq	Sales, quarterly.
seqoq	Stockholder equity, other, adjustments, quarterly.
seqq	Stockholder equity, parent, quarterly.
spiq	Special items, quarterly.
stkcoq	Stock compensation expense, quarterly.
teqq	Stockholders total equity, quarterly.
tfvaq	Total fair value assets, quarterly.
tfvlq	Total fair value liabilities, quarterly.
tstknq	Treasury stock, number of common shares, quarterly.
tstkq	Treasury stock, total, quarterly.
txdbaql	Taxes, deferred, long term, quarterly.
txdbq	Taxes, deferred, balance sheet, quarterly.
txtq	Taxes, total, quarterly.
txwq	Taxes withheld, quarterly.
xidoq	Extraordinary items and discontinued operations, quarterly.
xiq	Extraordinary items, quarterly.
xoprq	Operating expenses, other, quarterly.
aolochy	Level 2 and 3 assets, yearly.

Feature	Description
aqcy	Acquisitions, yearly.
capxy	Capital expenditures, yearly.
chechy	Cash and cash equivalents, yearly.
cibegniy	Comprehensive income, beginning of the year.
cicurry	Comprehensive income, current year.
cidergly	Comprehensive income due to derivative gains/losses, yearly.
cimiyy	Comprehensive income, minority interest, yearly.
ciothecy	Other comprehensive income, yearly.
ciothecy	Other comprehensive income, yearly.
cipeny	Comprehensive income due to pension gains/losses, yearly.
cisecgly	Comprehensive income due to securities gains/losses, yearly.
citotaly	Total comprehensive income, yearly.
c iy	Comprehensive income, yearly.
cogsy	Cost of goods sold, yearly.
cshfdy	Common shares fully diluted, yearly.
cshpry	Common shares preferred, yearly.
cstkey	Common stock equity, yearly.
dilady	Dilution adjustment, yearly.
dilavy	Dilution available, excluding extraordinary items, yearly.
dltisy	Debt, long-term, issuance, yearly.
dltry	Debt, long-term, reduction, yearly.
doy	Discontinued operations, yearly.
dpcy	Depreciation & amortization, statement of cash flows, yearly.
dpy	Depreciation & amortization, total, yearly.
dvpv	Dividend, preferred, yearly.
dvv	Dividends, yearly.
epsfiy	Earnings per share, fully diluted, including extraordinary items, yearly.
epsfxy	Earnings per share, fully diluted, excluding extraordinary items, yearly.
epspiy	Earnings per share, primary, including extraordinary items, yearly.
epspxy	Earnings per share, primary, excluding extraordinary items, yearly.
exrey	Exchange rate effect, yearly.
fiaoy	Financing activities, other, yearly.
fincfy	Financing activities, net cash flow, yearly.
fopoy	Funds from operations, other, yearly.
ibadjy	Income before extraordinary items, adjusted, yearly.
ibcomy	Income before extraordinary items, common shares, yearly.
ibcy	Income before extraordinary items, statement of cash flows, yearly.
iby	Income before extraordinary items, yearly.
intpny	Interest, paid, net, yearly.
ivacoy	Investments in associated companies, other, yearly.
ivch y	Increase in investments, yearly.
ivncfy	Investment activities, net cash flow, yearly.
niy	Net income, yearly.
nopiy	Non-operating income, yearly.
oancfy	Operating activities, net cash flow, yearly.
oepsxy	Operating earnings per share, diluted, from operations, yearly.
oiadpy	Operating income after depreciation, yearly.

Feature	Description
oibdpy	Operating income before depreciation, yearly.
opepsy	Operating earnings per share, yearly.
piy	Pretax income, yearly.
prstkcy	Purchase of common and preferred stock, yearly.
rdipay	Research and development expenses, after tax, yearly.
rdipy	Research and development expenses, yearly.
revty	Revenue, total, yearly.
saley	Sales, yearly.
sivy	Sale of investments, yearly.
spiy	Special items, yearly.
sppivy	Sale of PP&E and investments, loss data code, yearly.
sstky	Sale of common and preferred stock, yearly.
stkcoy	Stock compensation expense, yearly.
txbcfoy	Excess tax benefit of stock options, cash flow financing, yearly.
txbcoy	Excess tax benefit of stock options, cash flow operating, yearly.
txdcy	Income taxes, deferred, yearly.
txpdy	Income taxes, paid, yearly.
txty	Income taxes, total, yearly.
txwy	Excise taxes, yearly.
xidocy	Extraordinary items and discontinued operations, statement of cash flows, yearly.
xidoy	Extraordinary items and discontinued operations, yearly.
xopry	Operating expenses, total, yearly.
cshtrq	Common shares traded, quarterly.
dvpsspq	Dividends per share, quarterly.
dvpssxq	Dividends per share, ex-date, quarterly.
mkvaltq	Market value of the company, total, quarterly.
prccq	Price per common share, close, quarterly.
prchq	Price per common share, high, quarterly.
prclq	Price per common share, low, quarterly.
adjex	Cumulative adjustment factor by ex-date.
spcsrc	S&P quality ranking.

