Graph Neural Networks for predicting loan defaults: a comparative study with traditional ML models

HIDDE MUNTINGA, University of Twente, The Netherlands

With a newly emerging financial online presence, such as Peer-to-Peer (P2P) lending and a growing loan market for Small and Medium-sizes enterprises, new Machine learning (ML) technologies can be used to predict loan defaults in financial markets. Graph Neural Networks (GNNs) can offer a solution to assess risk for a bank when a company wants to apply for a loan. This research informs about different GNN methods used in recent literature, then chooses GCN as our model to perform research on and establish a framework for developing a GCN in our methodology. An experiment is performed to determine if GCN is indeed better at predicting loan default between loans that have a similar bank. The results show that GCN are performing better in all the benchmarking metrics: recall, precision, AUC (Area under the Curve), and accuracy, compared to Logistic Regression (LR).

Additional Key Words and Phrases: Graph Neural Network, Loan Default, Graphs, Financial Institutions, Artificial Intelligence, Credit Default, Graph Neural Network, Graph Convolutional Network, Logistic Regression, Database, Features

1 INTRODUCTION

Artificial Intelligence (AI) is reshaping many different industries, with numerous technological advancements being made in this field. Some of these developments are transforming how people understand the world. Benefits this field can provide are, automation of tasks, making rational decisions, enhance customer experiences and analyse vast amounts of data. Many different industries are also implementing this technology for their businesses.

The applications of Artificial Intelligence (AI), particularly those enabled through Machine Learning (ML) techniques, can be found across multiple fields. For example, Deep Learning (DL) techniques within ML have been used to identify: brain and gut failures for schizophrenia patients[18], traffic forecasting [10], fraud detection in banks [4, 19], and assisting the military by predicting terrorist attacks [6]. ML has demonstrated its ability to uncover complex patterns in data and make accurate, data-driven predictions. In the financial sector, ML techniques are increasingly being used to assess creditworthiness, detect fraudulent transactions, and optimise risk management strategies [4, 18]. Loan default prediction, in particular, has become a key focus area due to the high financial stakes involved.

Globally, financial markets have an increasing online presence, which is still increasing every year. The rise of Peer-to-Peer (P2P) lending and the emergence of fintech lenders have all contributed to an increase in the loan market for small and medium-sized enterprises (SME) [1]. This market acts as both a profit source and an investment opportunity for financial institutions. It also acts as an instrument for personal loans and adds flexibility in financial planning for people [19].

There is a big interest in predicting loan defaults for banks, as escalating instances of loan defaults cause massive losses in money lending companies (Ali Albastaki et al. [2]). The paper describes how financial institutions urge to develop methods that can minimise their losses and deal with time-sensitive tasks such as accepting or denying loans. Many financial institutions try to employ data analysis algorithms to help them to discover trends and patterns that can lead to insights into a persons' financial history (Ali Albastaki et al. [2]).

Neural Networks (NN) are one of the financial algorithms used to help banks assess borrowers' risk potential currently. Lately a type of NN, GNN (Graph Neural Network), has been getting more attention in literature, due to its convincing performance. This is due to the great expressive power of non-Euclidean data structures (graphs), which GNNs are able to analyse. By contrast, traditional ML methods, such as Logistic Regression (LR), XGBoost and Random Forest, only capture data that does not cover complex relationships. This makes GNNs an interesting tool for financial institutions, as they could potentially improve currently used ML models and can enhance the predictive power of loan defaults, in particular for banks.

This research will focus on the application domain of loan defaults in the banking industry, using a GNN. There are many types of default cases, such as voluntary bankruptcy and forfeiture or foreclosure proceedings, however the focus of this paper will be on monetary defaults. The goal is of this research is to help financial institutions get better insights into which models are more suitable for predicting loan defaults. In the long term, the results will potentially help banks financially and be more time-efficient in the lending process, by having better insights into model performance of different ML methods that assess risks for a particular bank.

To accommodate the financial industry with new instruments to improve financial prediction models, this paper will evaluate the following research question (RQ) to address the difficulty in predicting loan defaults based on banks:

RQ: To what extent does the predictive performance of GNN compare to LR, evaluated on banks in terms of recall, precision, AUC, and accuracy?

2 LITERATURE REVIEW

2.1 What are GNN

Jie Zhou et al. 2024 [21] describes theory on and proposes a general design pipeline for GNN models. The paper describes how GNN can be used for node classification, link prediction and clustering by utilising non-Euclidean data structures such as graphs, rather than Euclidean data structures such as images, or other grid-structured data. The graph in a GNN consists of nodes (objects) and their edges

TScIT 37, July 8, 2022, Enschede, The Netherlands

(relations between nodes). An example is illustrated in Figure 1, where each node can be a person in a social network and the relation could be the relation from person to person. However, Graphs can be constructed in various ways to reflect different social structures or tailored to represent domain-specific relationships in other industries. As such, the structure and semantics of a graph are highly adaptable to the context in which it is applied.

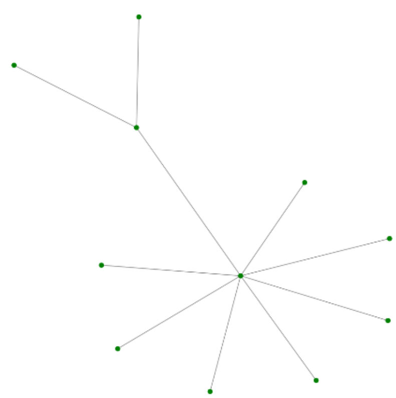

Fig. 1. Example graph structure used in GNNs

Nodes in a graph can be characterised by multiple features such as the borrower of the bank and their ZIP code. The primary objective of Graph Neural Networks (GNNs) is to aggregate information from neighbouring nodes to learn meaningful node representations [21]. As highlighted by Jie Zhou et al. (2024) [21], different GNN architectures primarily differ in how they perform this aggregation and combination of neighbouring information. Additionally, GNNs transform node features into low-dimensional embeddings, enabling the model to capture complex patterns from both numerical and encoded categorical features (e.g., a person's name).

2.2 Types of GNN

This section introduces unique types of GNNs, including state-of-the-art models that can complement or enhance standalone models, which only use a single methodological approach.

Hybrid methods for NN, called Hybrid Convolutional Neural Network (HCNN) are explained in Mengfang Sun et al. 2024 [16]. According to the article, HCNN can integrate various neural network (NN) models and algorithms, also the model can compare NN models with each other. The article shows that HCNN can be useful for predicting performance for (im)balanced datasets and can offer solutions for financial institutions. An application example is how Mengfang Sun. 2025 [17] implemented a hybrid GCN model, which uses local (learning from nearby neighbouring nodes) and global convolutional (learning from distant nodes) operations to assess the creditworthiness of the borrower. The graph in this paper was set up with the nodes being the characteristics of the borrower and the edges representing relationships between borrowers. The results show that the hybrid model is better compared to the standalone GCN model, and the hybrid model can also perform "over-smoothing and inadequate feature consideration by adaptively selecting and integrating features across different scales of graph structure, leading to a more nuanced and accurate representation of credit risk" (Menfang Sun. 2025). Additionally

Dynamic Graph Neural Network (DGNN) is another form of GNN, in which the graph contains temporal information, according to ZhengZhao et al. 2024 [7]. If graph are not static, but is prone to change then this type of NN is more efficient than a regular GNN. For example, in a network of relationships online, a graph structure with friend requests would change every time you be/de-friend someone. In this case a type of DGNN would be used. DGNN outperform static GNN (ZhengZhau et al., 2024). This article also describes some bottlenecks for that same technology and existing frameworks.

2.3 Applied GNN cases

Jona Becker et al. 2024, [4] is a study about fraud detection. This study presents their GNN using a heterogeneous graph, where nodes represent policyholders and vehicles, and edges encode relational connections. Specifically, policyholders are connected to one another through shared IBANs or phone numbers, while policyholders are linked to vehicles through claims involving the same vehicle. The evaluation concerned both standalone GNN models (GCN, GraphSAGE, and XENet) and hybrid models (XENet-GCN, XENet-GraphSAGE). Their findings indicate that the hybrid models outperform the standalone ones, with GCN performing the weakest overall across the evaluated metrics.

Jong Wook Lee at al. 2021 [9] uses a standalone GCN model, but divides the loans into three different relational types. These types of edges are: loan information, history information and soft information (social or behavioural information) of the borrower. According to the article, "the combination of the three types of information independently with different weights significantly improved the prediction of the default risk of the borrower". This led them to create their own combined GCN that outperforms a regular GCN model.

Sahab Zandi et al. 2024 [20] uses a dynamic model that is trained on data spanning 18 months from Januari 2012 to June 2013 and tests the data from July 2013 until December 2013. The paper evaluated two different graphs structures, one has the geographical location of the borrower and the other uses the company of the loan as a connector variable. The results highlight better performance of dynamic models, and looking at the static models, the Graph Attention Network (GAT) outperformed the GCN (Sahab Zandi et al. 2024 [20]).

Yuran Zhou 2024 [22] highlights the difference in performance among several different ML methods in predicting loan default in the financial sector. The study focuses on different ML methods: LR, Decision Tree (DT), Random Forest (RF) and XGBoost. What was observed in the research was that "DT has the highest Precision but the lowest AUC, and RF has the highest Accuracy but the lowest Recall. Additionally, XGBoost obtains the highest Recall and AUC but the lowest Accuracy and Precision". These results reveal that neither a singular traditional ML method is better for predicting loan defaults, as a few underperform in some areas but make a trade-off by outperforming on different performance indicators.

2.4 Critics on GNN types

There are also critical reviews of GNN. According to Zihao Li et al. 2024 [11], GCN performs reasonably well on the datasets, Lending Club dataset and Home Credit dataset. However, the article also clearly mentions limitations of the model, such as oversmoothing, difficulty in capturing rich topology and a fixed activation function. Oversmoothing arises due to the repetitive execution of the aggregate and update function, which causes the resulting node embeddings to become increasingly similar, thereby reducing their discriminative capability and expressive power [4]. Also mentioned by Simon Delarue etc al. 2024 [5] is that efforts have been make to reduce the complexity of NN. However, these models still have scalability and generalisation challenges. Therefore, we conclude that applying specific tools for certain conditions is necessary and needs to be researched.

2.5 Conclusion of literary review

In this review, inform the reader on what GNN are, give examples of state-of-the-art variations of GNN and their performance indicators. This gives insights into current practices of GNN applications. In this paper, we propose using a GCN to analyse loan default, due to its common use in existing literature [4, 9, 11]. Additionally, LR is widely used in the commercial and financial sectors due to its simplicity and ease of understanding [20].

3 METHODOLOGY

In this section, the building blocks to the experiment are laid. In here, the approach of what software (tools) was used to create the GCN, how the GCN was implemented and which considerations were taken into account for this, and how the hyperparameters were set to find a suitable solution to our model.

3.1 Dataset

A website with online communities for data scientists is searched, named Kaggle. On Kaggle a subset of the SBA_loans dataset [13], a governmental organisation called the Small Business Administration (SBA) organisation from the United States, was found. The dataset of SBA contained approximately 900.000 rows, which will provide our NN with richer data density.

In the dataset, there is mostly information on the loan and information on the borrower of the loan. In Table 5 in Appendix A.2, is a list containing all features and their respective descriptions. With this data, the relation between bank and borrower can be described as both of these features are present in the dataset.

The data needed pre-processing as some features were not immediately usable. Originally, the dataset did not have a column for "$Default$". However, this feature could be created from the original column *MIS_status*, which has the values *PIF* (Paid In Full) and *CHGOFF* (Charged Off), which can be translated to a 0 for non-default and to a 1 for default, indicating whether a loan has been paid or written as loss, respectively [14]. Also, the dollar signs in the column was filtered in excel.

Additionally, the dataset has been filtered based on the year the loan was approved to account for varying economic conditions affecting companies in different periods, which is a similar consideration taken by Sahab Zandi et al. 2025 [20]. We selected the year 2006, as it contained the highest number of entries out of all of the years present in the dataset (76040 entries) and is still seen as a period of economic stability, due to low market volatility [3]. After this selection, a check was performed if the dataset contained any null values. This was the case for the feature "*Bank*", so we filtered these values.

To ensure the representativeness of this subset, we examined the class distribution before and after filtering. The ratio of non-default to default remained approximately consistent, with the original dataset containing 82.3% non-default and 17.7% default rate, and the adjusted 2006 subset containing 79.1% non-default and 20.1% default rate, respectively.

3.2 Experimental tools

In this section, the software and its respective libraries used to execute the experiment of the GCN and LR are described. The software we have used to analyse the data is Python. Python is a popular programming language for data science and is suitable for our use-case as graphs and data can be easily implemented. Also, it is well integrated with well-known data science libraries. The libraries Pytorch Geometric (PG), Torch (T), Optuna (O) and Sklearn (S) are used in this project for implementing the GNN and LR. The main libraries we used have the following goals:

- (PG) Data, for making the undirectional graph dataset.
- (PG) GCNConv, for the basic implementation of the GCN model.
- (T) adam, for the optimiser for effective handling of sparse gradients and optimising non-stationary objectives.
- (O) Optuna was used to find the best hyperparameter tuning settings.
- (S) linear_model, this contains functions to implement the LR calculations.
- (S) labelencoder, can encode categorical data into integer labels, such as the name of the bank.
- (S) model_selection, creating a split in test data and training data.
- (S) metrics, calculating all the evaluation metrics, accuracy, precision, recall and AUC.
- pandas, for manipulating vectors and using operations on them.

3.3 Graph Convolutional Network

To analyse the GCN model, we define the input graph as $G = (V, E)$, where V is the set of nodes, each representing an individual loan l_i issued by a bank to a borrower (company) in the United States. Every node $l_i \in V$ is associated with a feature vector $\mathbf{x}_i$, which encodes attributes of the corresponding loan. E is the set of edges representing relationships between loans. An undirected edge $(l_i, l_j) \in E$ exists if loans l_i and l_j were issued by the same bank. This edge construction captures fully connected loan structures and model structural similarities based on similar banks. This edge construction leads to fully connected graphs based on similar banks.

This graph structure enables the Graph Convolutional Network (GCN) to aggregate information from neighbouring loans—those issued by the same bank—in order to predict the target label.

Design choices:

- The target label Y is set to the feature "Default", where each $y_i \in Y$, where $y_i \in \{0, 1\}$, represents non-default and default, respectively. This is the label that will be predicted for new loans.
- Categorical features are encoded using one-hot encoding, to make the data more representable for the GCN. Label encoding on the other hand assumes a categorical relationship between attributes. For example, if company 1 and company 2 were label encoded (e.g 0 and 1), then the GCN may incorrectly deduce an ordinal relationship, because company 2 might be "more important/bigger" than company 1.
- Numerical features were kept the same as how they were stated in the original dataset file.
- Optimisation model uses Adam and cross-entropy loss for training the model [8].
- ReLu for creating non-linearity in the model, creating more interesting results compared to linear functions.
- A test was performed to optimise our parameter settings. The parameters for the learning rate (LeR), number of GCN layers and embedding dimensions were tested.

3.4 Hyperparameter settings

Hyperparameters influence the model in its learning process. The GCN model does not learn these parameters by training itself, but need to be manually adjusted. Since choosing the right parameters influences our results, an analysis needs to be done on which parameter settings are best for training our model.

Parameter settings vary per study and are therefore not always clear. Therefore, a trial test was performed to find suitable variable parameter ranges for getting the best results from our GCN. To decide which values to test our trial on, will be based on multiple existing research papers with their research parameter settings, as there was no singular research that has done research on hyperparameters ranges for a graph with loan-to-loan nodes with the shared attribute of banks.

Already existing research report a LeR of 0.01 [10], 0.001 [18] [11], and 0.0005 (Lee et al. 2025 [9]. Also, the implementation of a three-layer GCN architecture is described. Zihao Li et al. (2024) [11] evaluated a range of values for embedding dimensions—8, 16, 32, and 64—as well as for the number of convolutional layers—1, 2, 3, and 4.

In Table 6, a list of hyperparameter values is summarised according to the aforementioned research on this topic. These hyperparameters are used in the later experiments, with the best-performing ones selected based on the best configuration for our GCN model. The best performing configuration is found by using Optuna, as mentioned in the experimental tools section.

4 EXPERIMENT

In this section, the experiment will be defined. Additionally, there is a need to measure the performance of the results, which will be discussed in the section evaluation metrics.

4.1 Setup

The experiment was conducted by comparing predicted loan defaults against actual defaults from 2006, based on borrower data. The predicted loan defaults are generated by training our GCN and LR models, based on the knowledge of previous loans that are connected to the same bank.

The distribution of the train-test split can be observed in Table 1. We have chosen for a distribution ratio of 60% train data versus 40% test data, as similar to Sahab Zandi et al. [20], who used a ratio of training to test data of approximately 148.000 nodes to 96.000 nodes respectively. With the train set, the experiment will model the prediction for loan defaults at specific banks for both the GCN and LR model. The test data is used for the actual loan defaults at specific banks.

The predictions are binary, non-default and default 0 and 1, respectively. Additionally, the banks will be evaluated if they have at least 20 loan requests. This decision was made to ensure that enough data is available on the network of loans for a specific bank to make a decisive prediction.

The nodes that are tested for the train-test data are randomly selected. Randomsation of the split will likely have an impact on our results slightly every time the experiment is executed. Thus, the experiment will be performed 10 times to identify the approximate range of the difference within different experiments.

Additionally, finding the best hyperparameters using Optuna, as mentioned in the Hyperparameter settings section, is excecuted here.

Split	**Nodes**	**Edges**
Total	23,417	3,592,774
Training	14,050	1,313,538
Testing	9,367	562,569

Table 1. Train-test split

4.2 Evaluation metrics

The metrics are based off of 4 different calculations TP, TN, FP, FN, where, TP is the amount of instances correctly identified as positive, TN is the amount of instances correctly identified as negative, FP is the amount of instances that are falsely identified as positive and FN is the amount of instances that are falsely identified as negative. In the context of loan default prediction:

- $Accuracy = (TP + TN)/(TP + TN + FP + FN)$, this metric measures the amount of non-defaults and defaults that are accurately predicted.
- $recall = TP/(TP + FN)$, this metric measures how many actual cases of default are correctly found by the model.
- $Precision = TP/(TP + FP)$, this metric measures how many predicted default are correct.

For the AUC metric, the variant Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) is used. This metric is used for binary, multiclass and multilabel classification problems [?].

Each metric will be indicating the performance of the GCN and LR model. For all of these metrics, it is observed that the higher the performance, the better the model is at predicting that class. The range is from 0 to 1, 0 being the worst to 1 being the best. These scores show how well the respective model has performed compared to the other.

4.3 Feature engineering

To select the most appropriate features for this experiment, several variables were chosen to represent both the borrower (node) and the bank (edge). The selected features are summarised in Table 2. In this figure the features, Name, City and Zip, SBA_Appr and GrAppr correspond to loan characteristics, while the feature Bank represents characteristics of the bank. The features for the loans were chosen as they represent both the loan from a financial side and the company's side. The Bank feature was included to represent the edge, enabling connections between loans issued by the same or similar banks.

Feature	Description	Category
Name	Name of borrower	Categorical
City	Resident city of the borrower	Categorical
Bank	Bank name	Categorical
Zip	Borrower zip address	Numerical
SBA_Appr	SBA loan guarantee	Numerical
GrAppr	Total loan amount approved	Numerical
Default	Borrower pays back loan	Numerical

Table 2. Description of the node features used for loan default prediction.

5 RESULTS

This section presents the results of our experiments using the proposed GCN model and a baseline LR model to predict loan defaults. Performance is evaluated using accuracy, precision, recall, F1-score, and AUC. For the GCN, we use a standalone architecture where the loan-to-loan graph is constructed based on the similarity of lending banks. Notably, as shown in Table 4, the improvement in precision is especially significant. Specifically, the GCN achieves improvements of 1.51% in accuracy, 12.3% in recall, 142.25% in precision, and 9.42% in AUC compared to the LR baseline. The table also shows that the GCN outperforms LR across all performance metrics: accuracy, precision, recall, and AUC.

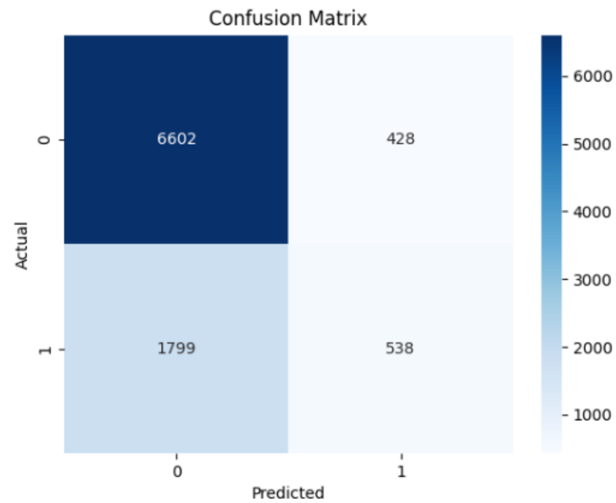

Fig. 2. Confusion matrix distribution of GCN model

The confusion matrix in Figure 2 shows the result of the TP (top left), FP (bottom left), TN (top right) and FN (bottom right). From the matrix, the distribution of our evaluation metrics performance is displayed. What can be observed from the Figure is that a lot of non-default cases get accurately predicted, 6.602 to 428 wrong identified cases. On the other hand, the distribution for default case getting accurately predicted is much lower, 1.799 FN cases compared to 538 FP cases. The confusion matrix shows high true positives for non-default, and low true positives for default. The confusion matrix suggest that there is a class imbalance, as the majority class (non-default) gets accurately predicted, but the minority class (default) does not. This is visible in the confusion matrix, as the majority class (6.602) has a high number of successful predictions, whereas the minority class has very little in comparison (538)

The results of the hyperparameter tuning are summarised in Table 3. These values were found to yield the best performance after experimenting with various combinations of hyperparameters.

Table 3. Hyperparameter tuning results

Hyperparameter	Value
Learning Rate (LeR)	0.001
Embedding dimension (emb_dim)	16
Number of GCN Layers (num_layers)	5

5.1 Discussion

This study shows that our implementation of GCN is indeed an implementation that can be useful for analysing loan defaults, as the results show a better performance for GCN compared to LR.

However, there are still some issues with the current implementation suggested in this paper. While the GCN model outperformed LR, its performance in detecting defaults remains limited, due to not many loan default being detected. The confusion matrix in Figure

2 highlights a critical limitation: the model performs substantially better on non-default predictions, with 6,602 true positives (TP) and 428 false positives (FP), compared to only 538 true negatives (TN) and 1,799 false negatives (FN). This discrepancy is likely due to the class imbalance in the dataset, where non-default cases make up approximately 80% of the data. As a result, the model becomes biased toward the majority class, leading to higher predictive accuracy for non-default cases while struggling to generalise to the minority (default) class.

In addition, it was expected that GCN would outperform LR, considering LR is not able to measure complex linear relations well, though being used in credit risk assessment quite frequently [12]

In comparison to related work, such as Sahab Zandi et al. (2024) [20], who reported AUC scores above 0.68 using dynamic multilayer GNNs on relational features (e.g., borrower location and lending institution), our model underperformed, achieving an AUC of 0.5793. Nevertheless, the novelty of our approach—using a loan-to-loan graph based on shared bank attributes—offers a unique perspective and provides a foundation for further exploration.

Overall, the effect on financial institutions, such as banks, when a predictive model is improved can be quite significant. Although there is limited literature providing a detailed, quantitative analysis of the exact costs incurred by banks when a loan defaults, it is generally accepted that defaults result in substantial financial losses for lenders. Beyond the direct monetary impact, loan defaults often render institutions inefficient due to operational challenges, including delays in customer service and the additional costs associated with recovering the owed funds [2].

6 CONCLUSION

This study performed an experiment to conclude whether GCN performs better compared to LR at predicting loan defaults for specific banks.

Financial institutions are in a constant situation of needing to reduce loan defaults, because it creates a significant financial burden and makes them inefficient. This is why financial institutions strive to decrease the loan defaults rate with better predictive models. We proceed to give financial institutions a better understanding of models that can improve loan default prediction. We have aided the model prediction of new loan defaults by setting up a GCN and a LR model compared to four different evaluation metrics, accuracy, recall, precision and AUC. The results show that our GCN model has a class imbalance, which affected the results. Nevertheless, the result shows in Table 4 indicates that GCN indeed outperform LR in for all of the evaluation metrics, accuracy, recall, precision and AUC. This does lead to an underperformance of the model compared to other studies of GCN. However, research into the GCN model with correct class balances is needed and with more models to compare, as it can reduce time efficiency and operational costs for financial institutions.

7 LIMITATIONS AND FUTURE RESEARCH

7.1 Limitations

One clear limitation of the study is the comparison between only two models, GCN and LR. This restriction makes additional research necessary, in contrast to when a combination of different models would be used. Models such as the previously mentioned hybrid models (see section 2.2) or alternative traditional ML methods could be a great addition to the comparative scope of this study. An exclusion of these methods allows for less comparative analysis between different models.

7.2 Future research

The scope of detecting loan defaults can be expanded, because this topic is much broader than only monetary loans. There are instances of loan defaults that the current solution can not provide, as different domains require different data. For example for predicting voluntary bankruptcy, the dataset needs to contain data on.

Also, the knowledge graph we have represented can be constructed in different ways. For predicting loan defaults, other types of loan default indicators have been used to identify loan defaults, such as using geographic proximity. However, other indicators such as economic indicators (e.g. inflation), do not have research on them. This can be an idea for future improvement of predicting monetary loan defaults.

For a real-life application of loan defaults in banks, we suggest researching dynamic graph implementation of GCN. Dynamic graphs address real-time change in models and are therefore more appropriate for handling real-time loan applications for financial institutions.

Furthermore, additional research needs to be done into the input graph. The proposed graph is made as one single graph with all nodes of each specific bank being interconnected, which can lead to the train data training the model partially on test data and vice versa. Instead, the train and test graph need to be independent of each other, meaning that nodes from the train data are not connected to nodes from the test data.

While this research does not focus on analysing the relative importance of individual features in predicting loan defaults, it may serve as a foundation for future studies aiming to identify feature weights using GCNs. Such analyses could prove valuable for financial institutions by offering insights into the underlying factors contributing to loan defaults, thereby enabling more effective model improvements and decision-making.

Future work should address the problem of a class imbalance. This can be done by using resampling techniques such as Synthetic Minority Oversampling Technique (SMOTE) to improve the model's ability to detect defaults more effectively.

ACKNOWLEDGMENTS

Thank you to my supervisors Manuele Massei and Wouter van Heeswijk for assisting me through the work of my thesis. The meetings we had were helpful and informative in the process of creating this my thesis.

REFERENCES

[1] Kaleemullah Abbasi, Ashraful Alam, Noor Ahmed Brohi, Imtiaz Ali Brohi, and Shahzad Nasim. 2021. P2P lending Fintechs and SMEs' access to finance. (2021).
[2] Ali Albastaki and Ali Abdullatif. 2021. *Loan Default Prediction System.* Major Qualifying Project (MQP). Rochester Institute of Technology. https://repository.rit.edu/cgi/viewcontent.cgi?article=12544&context=theses Accessed: 2025-06-28.
[3] Samad Sarferaz Albrecht Ritschl and Martin Uebele. 2016. The U.S. Business Cycle, 1867–2006: A Dynamic Factor Approach. *Cliometrica* 10, 1 (2016), 1–19.

Table 4. Comparison of GCN and LR Performance Metrics

Metric	GCN (Mean ± Range)	LR (Mean ± Range)	Relative Difference (%)
Accuracy	0.7614 (0.7602 − 0.7626)	0.7501 (fixed)	+1.51%
Precision	0.5561 (0.5519 − 0.5604)	0.4952 (fixed)	+12.30%
Recall	0.2158 (0.2004 − 0.2312)	0.0890 (fixed)	+142.25%
AUC	0.5793 (0.5735 − 0.5851)	0.5294 (fixed)	+9.42%

https://doi.org/10.1007/s11698-015-0114-0

[4] Jona Becher and Andres Schäfer. 2024. Application of Graph Neural Networks to fraud classification problems in the insurance and financial domain. *Gesellschaft fur Informatik (GI)* (2024). https://doi.org/10.18420/inf2024_117 Likely peer-reviewed workshop paper.

[5] Simon Delarue, Thomas Bonald, and Tiphaine Viard. 2024. Graph as a feature: improving node classification with non-neural graph-aware logistic regression. *arXiv preprint arXiv:2411.12330* (2024). https://doi.org/10.48550/arXiv.2411.12330

[6] Yingjie Du, Ning Ding, and Hongyu Lv. 2025. Spatio-temporal prediction of terrorist attacks based on GCN-LSTM. *Computers in Security* (2025). Uses GCN-LSTM model for spatio-temporal prediction of terrorist attacks..

[7] ZhengZhao Feng, Rui Wang, TianXing Wang, Mingli Song, Sai Wu, and Shuibing He. 2024. A Comprehensive Survey of Dynamic Graph Neural Networks: Models, Frameworks, Benchmarks, Experiments and Challenges. *arXiv preprint arXiv:2405.00476* (2024). https://arxiv.org/abs/2405.00476 Under review at PVLDB.

[8] Tilmann Gneiting and Adrian E Raftery. 2007. Strictly Proper Scoring Rules, Prediction, and Estimation. *J. Amer. Statist. Assoc.* 102, 477 (2007), 359–378. https://doi.org/10.1198/016214506000001437

[9] Jong Wook Lee, Won Kyung Lee, and So Young Sohn. 2021. Graph convolutional network-based credit default prediction utilizing three types of virtual distances among borrowers. *Expert Systems with Applications* 168 (2021), 114411. https://doi.org/10.1016/j.eswa.2020.114411

[10] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. *ICLR 2018* (2018). Introduced DCRNN for traffic forecasting; demonstrates GCN's value in spatial-temporal modelling tasks..

[11] Zihao Li, Yakun Chen, Xianzhi Wang, Lina Yao, and Guandong Xu. 2024. Multi-view GCN for loan default risk prediction. (2024). https://doi.org/10.1007/s00521-024-09695-x

[12] Qiang Liu, Zhaocheng Liu, Haoli Zhang, and Yuntian Chen. 2021. Mining Cross Features for Financial Credit Risk Assessment. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*. 1069–1078. https://doi.org/10.1145/3459637.3482371

[13] Kevin Miller. 2020. *SBA loan approval dataset*. https://www.kaggle.com/code/kevinm6720/sba-loan-approval-analysis/input Accessed: 2025-06-27.

[14] Li Min, Amy Mickel, and Stanley Tayler. 2018. Sampling error vs. measurement error in survey research: Norming in the context of standards for educational and psychological tests. *The Journal of Statistics Education* 26, 3 (2018), 116–139. https://doi.org/10.1080/10691898.2018.1434342

[15]]rocAucScore scikit-learn developers. [n. d.]. *roc_auc_score*. Accessed: 2025-06-24.

[16] Mengfang Sun, Wenying Sun, Ying Sun, Shaobo Liu, Mohan Jiang, and Zhen Xu. 2024. Applying Hybrid Graph Neural Networks to Strengthen Credit Risk Analysis. *arXiv preprint arXiv:2410.04283* (2024). https://arxiv.org/abs/2410.04283

[17] Mengfang Sun, Wenying Sun, Ying Sun, Shaobo Liu, Mohan Jiang, and Zhen Xu. 2024. Applying Hybrid Graph Neural Networks to Strengthen Credit Risk Analysis. *arXiv preprint arXiv:2410.04283* (2024). https://arxiv.org/abs/2410.04283

[18] Haiyuan Wang, Runlin Peng, Yuanyuan Huang, Liqin Liang, Wei Wang, Baoyuan Zhu, Chenyang Gao, Minxin Guo, Jing Zhou, Hehua Li, Xiaobo Li, Yuping Ning, Fengchun Wu, and Kai Wu. 2025. MO-GCN: A multi-omics graph convolutional network for discriminative analysis. *Neurocomputing* (2025). https://doi.org/10.1016/j.brainresbull.2025.111199 Identifies 10 key features contributing to schizophrenia using graph classification. GCN shown to be effective in spatial-temporal and relational data analysis..

[19] Yuxuan Yang, Minghui He, and Jiaxi Zhang. 2024. Comparative study of Optuna-optimized LightGBM, XGBoost and their hybrid stacking models in personal loan default prediction. *Artificial Intelligence and Computer Engineering (ICAICE 2024)* (2024). https://doi.org/10.1145/3716895.3717020

[20] Sahab Zandi, Kamesh Korangi, María Óskarsdóttir, Christophe Mues, and Cristián Bravo. 2024. Attention-based Dynamic Multilayer Graph Neural Networks for Loan Default Prediction. *Expert Systems with Applications* 238 (2024), 119400. https://doi.org/10.1016/j.eswa.2024.119400

[21] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph Neural Networks: A Review of Methods and Applications. *AI Open* 1 (2020), 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

[22] Yuran Zhou. 2022. Loan Default Prediction Based on Machine Learning Methods. In *Proceedings of the 2022 2nd International Conference on Big Data Engineering and Applications (BDEIM) (EAI/Springer Innovations in Communication and Computing)*. Springer. https://doi.org/10.4108/eai.2-12-2022.2328740

7.3 Appendix A.1

Notice of AI use. The author has used AI as an assisting tool to check cohesion and spelling of the text. The input of the text was a product of the author himself. The author takes full responsibility for the text represented in this paper, having reviewed and edited the text generated from AI accordingly. Additionally, the author has asked AI to provide information on the topic, as it was sometimes hard to understand. This text was not used directly in the paper, but was used to improve the understanding of the topic.

7.4 Appendix A.2

Feature	Description
LoanNr_ChkDgt	Loan Identifier
Name	Borrower name
City	Borrower city
State	Borrower state
Zip	Borrower zip code
Bank	Bank name
BankState	Bank state
NAICS	North American Industry Classification System (NAICS) code
ApprovalDate	Date SBA commitment issued
ApprovalFY	Fiscal year of commitment
Term	Loan term in months
NoEmp	Number of business employees
NewExist	1 = Existing business, 2 = New business
CreateJob	Number of jobs created
RetainedJob	Number of jobs retained
FranchiseCode	Franchise code, (00000 or 00001) = No franchise
UrbanRural	1 = Urban, 2 = rural, 0 = undefined
RevLineCr	Revolving line of credit: Y = Yes, N = No
LowDoc	LowDoc Loan Program: Y = Yes, N = No
ChgOffDate	The date when a loan is declared to be in default
DisbursementDate	Disbursement date
DisbursementGross	Amount disbursed
BalanceGross	Gross amount outstanding
MIS_Status	Loan status charged off = CHGOFF, Paid in full = PIF
ChgOffPrinGr	Charged-off amount
GrAppv	Gross amount of loan approved by bank
SBA_Appv	SBA's guaranteed amount of approved loan

Table 5. SBA loans dataset feature description

7.5 Appendix A.3

Table 6. Reported ranges for GCN hyperparameters across studies

Hyperparameter	Reported Values
LeR (LeR)	0.0005, 0.001, 0.01
Number of GCN Layers	1, 2, 3, 4, 5
Embedding Dimensions	8, 16, 32, 64