Uncertainty-aware Machine Learning with Applications to Credit Risk

A dissertation in partial fulfillment of the requirements for the degree of
Doktor der Wirtschaftswissenschaft (Dr. rer. pol.)

submitted to the

Faculty of Business, Economics,
and Management Information Systems

University Regensburg

submitted by

Matthias Nagl, M.Sc. in Survey Statistics

Advisors

Prof. Dr. Daniel Rösch

Prof. Dr. Ralf Kellner

Date of disputation

November 4th, 2024

Acknowledgments

First and foremost, I would like to thank Prof. Dr. Daniel Rösch for his never ending support and encouragement to always improve my work. It was an honor to work with you on these projects. Thank you for taking the time to listen to my rough ideas and advising me what is really important to make a good paper out of them. Further, I want to express my gratitude to Prof. Dr. Ralf Kellner for being my second advisor. Thank you for your ongoing support and motivation.

I'm especially grateful for my colleagues at the University of Regensburg. You made this journey a wonderful and fascinating experience. Thank you all for always having an open door and answering my questions when I needed your help. Working on ideas is only really fun when you are doing it together.

Further, I want to thank my brother and colleague Maximilian who served as a mentor in the scientific world to me. Thank you for your guidance and your continuous feedback. I get asked a lot: "How is it to work with your brother?" and I can say it works very well.

Furthermore, I want to thank my dear friends, Florian Scholze and Alexander Stemke, who accompanied me through my master and this journey. Doing a PhD can be challenging and it's good to have friends who can relate to it.

I dedicate this thesis to my amazing wife Regina. I'm most grateful for you and all the support you gave me. Thank you for raising me up every time I fell and listen to my thoughts about statistical problems that are only my problems. And lastly, I want to thank my parents, Lorenz and Kathrin, for letting me find my own way. The name by which we are known is part of our identity. And with this work, my dear father, I try to ensure that our name will never be forgotten.

Contents

List of Figures **iii**

List of Tables **vi**

Introduction **1**

1 Quantifying uncertainty of machine learning methods for loss given default **10**

1.1 Introduction 11

1.2 Data 14

1.3 Methods 17

1.4 Results 21

1.4.1 Learning strategy 21

1.4.2 Aleatoric and epistemic uncertainty in predictions 24

1.4.3 Explaining LGD predictions 26

1.5 Conclusion 28

2 Non-linearity and the distribution of market-based loss rates **30**

2.1 Introduction 31

2.2 Literature review 33

2.3 Data 35

2.4 Methods 37

2.5 Results 44

2.5.1 Feature selection & model estimation 44

2.5.2 Drivers of μ and ϕ 48

2.5.3 Scenario analysis 54

2.6 Conclusion 58

2.A Descriptive statistics 60

2.B Trainable activation functions 63

2.C Robustness 65

2.D Scenario analysis 68

3 GAMME - Advances in Predictive Mean Matching 70

3.1 Introduction 71

3.2 Background 72

3.2.1 Missing data mechanisms 72

3.2.2 Literature 73

3.3 Methodology 76

3.3.1 Predictive mean matching 76

3.3.2 Neural network 77

3.3.3 Accumulated local effect (ALE) plots 78

3.3.4 GAMME 79

3.3.5 GAMME with interactions 81

3.4 Simulation study 83

3.4.1 Simulation setup 83

3.4.2 Hyperparameters 85

3.4.3 Metrics 86

3.4.4 Results 86

3.5 Conclusion 93

3.A Pseudocode 95

3.B Simulation results large sample (uniform) without interactions 97

3.C Simulation results large sample (normal) without interactions 98

3.D Simulation results small sample (uniform) without interactions 100

3.E Simulation results small sample (normal) without interactions 102

3.F Simulation results large sample (uniform) with interactions 104

3.G Simulation results large sample (normal) with interactions 105

3.H Simulation results small sample (uniform) with interactions 107

3.I Simulation Results small sample (normal) with interactions 109

Conclusion 111

References 113

List of Figures

1.1 Histogram of LGDs 15

1.2 Uncertainty estimation in sample 24

1.3 Uncertainty estimation out-of-sample 25

1.4 Uncertainty estimation - out-of-time 25

1.5 Bond-related drivers 26

1.6 Macroeconomy-related drivers 27

1.7 Uncertainty-related drivers 28

2.1 Histogram of LGDs 36

2.2 Graphical illustration BRANN and G-BRANN 42

2.3 ALE plots of bond characteristics | μ 49

2.4 ALE plots of macroeconomic variables | μ 50

2.5 ALE plots of uncertainty related variables | μ 51

2.6 ALE plots of bond characteristics | ϕ 51

2.7 ALE plots of macroeconomic variables | ϕ 52

2.8 ALE plots of uncertainty related variables | ϕ 53

2.9 Beta distributions per seniority 55

2.10 Beta distributions per industry type 56

2.11 Beta distributions per macroeconomic states 57

2.12 Degree of overlapping across the macroeconomic states 57

2.B.1 Trainable exponential function 63

2.B.2 Trainable softplus function 63

2.B.3 Trainable sigmoid function 64

2.D.1 Overlappings | Seniority & Macroeconomic state 68

2.D.2 Overlappings | Industry 69

3.1 Illustrative example of the the ALE transformation 80

3.2 Illustrative example of second order ALE plots 82

List of Tables

1.1 Descriptive statistics of LGDs across the whole sample 15

1.2 Selected variables for the network 15

1.3 Upper triangle of the correlation matrix of macroeconomic and uncertainty features 16

1.4 Descriptive statistics of LGDs according to the seniority of the defaulted bond 16

1.5 Descriptive statistics of LGDs according to the default type 16

1.6 Setup and final values of the hyperparameter search 22

1.7 Evaluation metrics 23

2.1 Upper triangle of the correlation matrix of uncertainty features in percentages 37

2.2 Trainable activation functions 41

2.3 Selected variables for the sub-networks 45

2.4 Setup and final values of the hyperparameter search 46

2.5 Evaluation Metrics 47

2.A.1 Descriptive statistics of LGDs across the whole sample. 60

2.A.2 Descriptive statistics of LGDs according to the seniority of the defaulted bond. 60

2.A.3 Descriptive statistics of LGDs according to the industry sector of the defaulted bond. 61

2.A.4 Descriptive statistics of LGDs according to the default type 62

2.C.1 Setup and final values of the hyperparameter search - robustness 66

2.C.2 Evaluation Metrics 67

3.1 Average relative bias in % for 10,000 observations (uniform) under MAR without interactions 88

3.2 Coverage rate in % for 10,000 observations (uniform) under MAR without interactions 89

3.3 Average relative bias in % for 10,000 observations (uniform) under MAR with interactions 91

3.4 Coverage rate in % for 10,000 observations (uniform) under MAR with interactions 92

3.B.1 Average relative bias in % for 10,000 observations (uniform) under MCAR without interactions . . . 97
3.B.2 Coverage rate in % for 10,000 observations (uniform) under MCAR without interactions . . . 97
3.C.1 Average relative bias in % for 10,000 observations (normal) under MAR without interactions . . . 98
3.C.2 Coverage rate in % for 10,000 observations (normal) under MAR without interactions . . . 98
3.C.3 Average relative bias in % for 10,000 observations (normal) under MCAR without interactions . . . 99
3.C.4 Coverage rate in % for 10,000 observations (normal) under MCAR without interactions . . . 99
3.D.1 Average relative bias in % for 2,000 observations (uniform) under MAR without interactions . . . 100
3.D.2 Coverage rate in % for 2,000 observations (uniform) under MAR without interactions . . . 100
3.D.3 Average relative bias in % for 2,000 observations (uniform) under MCAR without interactions . . . 101
3.D.4 Coverage rate in % for 2,000 observations (uniform) under MCAR without interactions . . . 101
3.E.1 Average relative bias in % for 2,000 observations (normal) under MAR without interactions . . . 102
3.E.2 Coverage rate in % for 2,000 observations (normal) under MAR without interactions . . . 102
3.E.3 Average relative bias in % for 2,000 observations (normal) under MCAR without interactions . . . 103
3.E.4 Coverage rate in % for 2,000 observations (normal) under MCAR without interactions . . . 103
3.F.1 Average relative bias in % for 10,000 observations (uniform) under MCAR with interactions . . . 104
3.F.2 Coverage rate in % for 10,000 observations (uniform) under MCAR with interactions . . . 104
3.G.1 Average relative bias in % for 10,000 observations (normal) under MAR with interactions . . . 105
3.G.2 Coverage rate in % for 10,000 observations (normal) under MAR with interactions105

3.G.3 Average relative bias in % for 10,000 observations (normal) under MCAR with interactions 106

3.G.4 Coverage rate in % for 10,000 observations (normal) under MCAR with interactions106

3.H.1 Average relative bias in % for 2,000 observations (uniform) under MAR with interactions 107

3.H.2 Coverage rate in % for 2,000 observations (uniform) under MAR with interactions107

3.H.3 Average relative bias in % for 2,000 observations (uniform) under MCAR with interactions 108

3.H.4 Coverage rate in % for 2,000 observations (uniform) under MCAR with interactions108

3.I.1 Average relative bias in % for 2,000 observations (normal) under MAR with interactions 109

3.I.2 Coverage rate in % for 2,000 observations (normal) under MAR with interactions109

3.I.3 Average relative bias in % for 2,000 observations (normal) under MCAR with interactions 110

3.I.4 Coverage rate in % for 2,000 observations (normal) under MCAR with interactions110

Introduction

Motivation and area of research

Over the last decades machine learning and artificial intelligence became increasingly important for scientific research and practical applications, see, e.g., Bharadiya et al. (2023). The superior performance compared to standard algorithms makes machine learning a preferred choice in many applications, see, e.g., Bellotti et al. (2021) or Bohr and Memarzadeh (2020). However, most of the modern algorithms are considered as "back boxes" and are difficult to interpret, see, e.g., Burkart and Huber (2021). Therefore, the need of explainable artificial intelligence (XAI) approaches is becoming more important. To justify the use of machine learning algorithms those XAI approaches are crucial, see, e.g., Burkart and Huber (2021) or European Banking Authority (2023a). Furthermore, the uncertainty related to the prediction should also be a key component for machine learning algorithms since it helps to improve the reliability of predictions (Tyralis and Papacharalampous, 2024).

One example where reliability of the model and their predictions are crucial are financial institutions. This holds especially for the credit risk sector, see, e.g., Fritz-Morgenthal et al. (2022). According to the risk assessment report of the European Banking Authority credit risk is of great importance as it causes 84% of all risk weighted assets (European Banking Authority, 2023b). Credit risk is typically characterized by the expected loss (EL) amount which contains three components. The probability of default (PD), the loss given default (LGD) and the exposure at default (EAD), see Basel Committee on Banking Supervision (2006). Recent studies show that more complex approaches like machine learning models tend to perform superior compared to classical statistical approaches, see, e.g., Bellotti et al. (2021). Most standard techniques especially the machine learning approaches neglects the distribution of the prediction and therefore the associated uncertainty. There is an active area of research that combines statistical methods and machine learning to obtain a flexible model that can characterize the the distribution of the prediction. For an overview, see Tyralis and Papacharalampous (2024).

According to Baesens and Smedts (2023) preprocessing the raw data is a crucial step in credit risk modeling. One part of this preprocessing is handling missing values. Modeling uncertainty plays also an important role in handling missing data. Most imputation approaches replace the missing values by one estimated or one observed value. Using such an imputed dataset assumes that the imputed value corresponds to the observed value since it does not account for the uncertainty due to missingness. This often leads to problems when statistical inference is conduced, see, e.g., (Van Buuren, 2018, Ch. 1). There is a quote, that according to Kniss (2008) is attributed to Richard Feynman that says: *"What is not surrounded by uncertainty cannot be the truth."*. Despite related to another context this quote summarizes this fundamental problem in missing data very well. As a solution to this problem multiple imputation was introduced in the 1970s (Rubin, 2004). In this framework as pointed out in (Van Buuren, 2018, Ch. 1) the dataset is completed multiple times. Each completed dataset is analyzed separately and the results are pooled afterwards to conduct valid statistical inference.

This thesis aims to deal with the topic of uncertainty-aware machine learning approaches. The application of and the extension to uncertainty-aware machine learning techniques are covered in three independent research papers that correspond to the Chapters 1, 2 and 3. The first paper deals with a recent machine learning approach, that can calculate the uncertainty that comes along with the prediction and divide it into aleatoric and epistemic uncertainty. This method is applied to market-based LGDs in order to get a deeper understanding of the uncertainty and the challenges that are associated with it. The second paper uses the insights of the first paper and combines a statistical approach with a machine learning technique that allows flexible modeling in distributional parameters. This is also applied to market-based LGDs and further analyzed with an XAI method to get a better insight in LGDs and in its most important features for their prediction. The last and third research paper deals with the important task of handling missing data. In this paper a well established imputation method is extended by combining the learned relationships of a neural network with it. This allows the in this paper introduced novel imputation approach to handle missing values such that statistical inference is valid under a non-linear data generating process. Furthermore, a heuristic is proposed to extend this method to interactions as well.

Research paper I | *Quantifying uncertainty of machine learning methods for loss given default*

The expectes loss (EL) amount plays a central role in risk management. It is defined as the product of the probability of default (PD), the exposure at default (EAD) and the loss given default (LGD), see Basel Committee on Banking Supervision (2006). PD denotes the probability that an creditor is unable to fulfill his agreed obligations. EAD is defined as the outstanding obligations at the time of default. The last component LGD denotes the proportion of the credit line that can not be recovered due to the default. This paper focuses on market-based LGD that represents the average relative decrease in the market price of corporate bonds 30 days after default, see, e.g. Gambetti et al. (2019). As stated in SIFMA Research (2022) this market has a estimated volume of 10 trillion USD and therefore, is an essential part of the financial market and its stability. There is a board range of studies on market-based LGDs focusing on predicting and explaining LGDs, see, e.g., Loterman et al. (2012); Kaposty et al. (2020) or Bastos and Matos (2022), but not the corresponding uncertainty of the prediction. Therefore, the research question remains: How big is the uncertainty that comes along with the prediction and is this uncertainty caused by the data or rather by the model? This paper aims to close this gap by quantifying the uncertainty in the LGD prediction by using the deep evidential regression as introduced in Amini et al. (2020) and extended in Meinert et al. (2022). This allows to separate the uncertainty into aleatoric uncertainty, that is related to the uncertainty in the data itself, and into the epistemic uncertainty, that corresponds to the uncertainty due to the model.

Research paper II | *Non-linearity and the distribution of market-based loss rates*

The LGD is one key component of the EL. Due to its complex structure and difficult to model characteristics like multi modality and bounded support, advanced methods are applied, see, e.g., Gambetti et al. (2019). One well established approach to meet these characteristics is to model the LGD by a beta regression. This method assumes that the LGD follows a beta distribution with a mean and a precision parameter, whereas the latter is often treated as a nuance parameter. Furthermore, it relies on linearity and therefore neglects non-linearities if not modeled explicitly. However, several studies like Bastos (2010); Loterman et al. (2012) or Olson et al. (2021) conclude that models, that can model non-linearity, improves the prediction of LGDs. This leads to the following research questions: First, how much non-linearity is in the modeled estimates? Second, are there differences if a precision parameter is modeled additionally? The second research paper aims to answer this questions by combining a statistical framework, the beta regression, with a machine learning approach, the neural network, and analyze the resulting model with an explainable artificial intelligence technique called Accumulated Local Effect Plot (ALE Plot) by Apley and Zhu (2020) to get a deeper understanding of marked-based LGDs.

Research paper III | *GAMME - Advances in Predictive Mean Matching*

Missing data is a common problem in most scientific and practical areas as illustrated in King et al. (2001); Rubin (2003); Bryzgalova et al. (2024). Since many applications require fully observed data, handling missing data is part of the important data preprocessing, see, e.g., Baesens and Smedts (2023). If a substantial amount of observations is missing deleting those can reduce the dataset immensely and make some applications not suitable anymore. Therefore, imputing missing values is often chosen. But improper imputation can bias the results and lead to incorrect statistical inference, see, e.g., Schafer and Graham (2002). One common approach is to use predictive mean matching which is a non-parametric imputation technique that provides good results under different assumptions, see, e.g., Kleinke (2017). This and many other imputation models rely on linear dependencies which does not necessary meet the truth. As a consequence several imputation models are proposed to overcome this problem, see, e.g, Stekhoven and Bühlmann (2012); Doove et al. (2014); Deng and Lumley (2023). But they can all result in invalid statistical inference. Therefore, the research question remains: How can missing values being imputed such that statistical inference is valid if the data generating process is not linear? The aim for this research paper is to answer that question and provide a solution. The well established predictive mean matching approach is extended by utilizing an explainable artificial intelligence approach to reveal non-linearities and account for them. Furthermore, a heuristic is proposed that extends this method to interactions.

Literature

Uncertainty and its estimation is a essential part in classical statistics. It is the basis for statistical tests, confidence intervals or prediction intervals. One branch of statistics that is closely related to uncertainty is bayesian statistic. This allows to especially model epistemic uncertainty that is associated with the model itself and differs from the aleatoric uncertainty, that is inherent to the data (Gawlikowski et al., 2022). Besides the classical statistical models uncertainty becomes more important in the field of machine learning. The “natural” extension is to apply bayesian theory to machine learning algorithms. For neural networks this is possible by using Bayes by Backprop (Blundell et al., 2015), that allows to learn a distribution over the neural network weights. Due to the increased computational burden there are some alternatives published in recent years. Lakshminarayanan et al. (2017) use an ensemble of neural networks with a negative log-likeikelihood as loss function to assess uncertainty. Gal and Ghahramani (2016) showed, that using the regularization technique dropout at test time can approximate a deep Gaussian process. This is extended by Mobiny et al. (2021) that drops weights instead of neurons. A further possibility to estimate uncertainty is the deep evidential regression by Amini et al. (2020) and its extension by Meinert et al. (2022) and Meinert and Lavin (2022). This approach has the advantage of separating aleatoric and epistemic uncertainty by only minor changes to the neural network architecture. A literature review on this topic can be found in Chapter 1 (Section 1.1, Introduction).

One possible field of application is the modeling of Loss Given Defaults (LGDs). The LGD is one key parameter in calculating the expected loss (EL) that is essential for credit risk management. In fact, the adequate modeling of LGDs is crucial for financial institutions as they are allowed to use their own models (Basel Committee on Banking Supervision, 2017). Furthermore, from a economic point of view LGDs for the US corporate bond market are of great importance due to the large volume of this market (SIFMA Research, 2022) . In classical statistics LGDs are primarily modeled with linear models such as beta regression (Ferrari and Cribari-Neto, 2004; Gambetti et al., 2019) or fractional response regression (Bastos, 2010). With the rise of artificial intelligence complex machine learning algorithms found their way into the LGD literature. This lead to the opportunity of allowing non-linearity and interactions without specifying them in advance. One example are regression trees as used in Bastos (2010) or random forests and neural networks in Kaposty et al. (2020). A vast selection of different models are evaluated in Bellotti et al. (2021) including random forests, boosted trees and Cubist that are superior in terms of forecasting. Most studies focus on forecasting LGDs and only a few consider distributional characteristics of the LGDs as in Gambetti et al. (2019) or Kellner et al. (2022). A more in depth literature review can be found in Chapter 1 (Section 1.1, Introduction) and Chapter 2 (Section 2.2, Literature review).

Uncertainty is also an important aspect in the research area of missing values. Missing values are a common problem in many practical applications and scientific fields, see, e.g., King et al. (2001); Boeschoten et al. (2019); Bryzgalova et al. (2024). As stated in Baesens and Smedts (2023) the preprocessing of the data including handling missing values can affect the performance of credit risk models. To impute missing values there is a broad body of literature on this topic. Most approaches can be divided into single imputation (SI) and multiple imputation (MI) techniques. Standard SI approaches are mean imputation, regression imputation or stochastic regression imputation, see, e.g., (Van Buuren, 2018, Ch. 1). They often assume only linear dependencies. Therefore, they can be extend to more complex and flexible models. One prominent example is MissForest by Stekhoven and Bühlmann (2012) that rely on iterative random forests. On the neural network side the imputation approaches are often based on generative adversarial networks (GANs) by Goodfellow et al. (2014), see, e.g., GAIN by Yoon et al. (2018) or MisGAN byLi et al. (2019). These single imputation techniques treat the imputed values as observed ones and thus neglect the uncertainty due to missingness. This is especially important if the goal is to conduct valid statistical inference, see, e.g., (Van Buuren, 2018, Ch. 1). As a consequence multiple imputation was introduced that produces multiple imputed datasets. On each of these datasets a model is applied e.g. a linear regression. The results of these regressions can further be pooled to get valid statistical inference. Most SI methods can be extended to MI approaches. A well established MI method is predictive mean matching (PMM) that imputes only observed values by matching predictive means. PMM also relies on linear dependencies, see, e.g., (Van Buuren, 2018, Ch. 3), but there are a few possible extensions to overcome this burden e.g. by using a classification and regression tree (CART) as in Doove et al. (2014) instead of a linear regression. One recent development by Deng and Lumley (2023) uses extreme gradient boosting (XGBoost) as the underlying model. A comprehensive overview on the current literature can be found in Chapter 3 (Subsection 3.2.2, Literature).

Contributions

This thesis contributes to the literature by studying various aspects of uncertainty in the application of machine learning. In particular it contributes by quantifying the uncertainty in the LGD prediction and proposing an approach that allows more flexibility in modeling the dominant aspects of the LGD distribution. Furthermore, for a broader range of applications an imputation technique is proposed to correct the uncertainty due to missingness in order to make statistical inference valid. The Chapters 1, 2 and 3 represent the independent research papers that structure the main contributions of this thesis.

Contribution I | *Quantifying uncertainty of machine learning methods for loss given default*

Although Loss Given Default is studied in many recent publications, see, e.g., Calabrese and Zanin (2022), most of them focus on prediction as in Loterman et al. (2012); Bellotti et al. (2021) or finding the most relevant drivers, see, e.g., Gambetti et al. (2019). There are only very few publications that study the uncertainty that is associated with it. Research paper I (see Chapter 1) aims to contribute to the literature by closing this gap. Therefore, a method known as deep evidential regression by Amini et al. (2020) and its extension by Meinert et al. (2022) is applied to model LGDs. This approach has the advantage of obtaining the uncertainty along with the prediction. Furthermore, this uncertainty can be divided into aleatoric and epistemic uncertainty. As stated in Gawlikowski et al. (2022) the latter can be reduced by increasing the number of training samples and is also known as model uncertainty. Aleatoric uncertainty covers the uncertainty of the data itself and can not be reduced. Furthermore, the features are analyzed with ALE plots by Apley and Zhu (2020). This reveals non-linearities in the mean prediction.

Applying the deep evidential regression framework leads to several interesting findings. First, although the deep evidential regression has some additional parameter in order to model the uncertainty the performance of this approach is comparable to common methods for LGD estimation. Second, for the mean prediction the proportion of the uncertainty that is associated with the model is smaller than the proportion of the aleatoric uncertainty. Therefore, most of the uncertainty can be attributed to data inherent uncertainty.

Contribution II | *Non-linearity and the distribution of market-based loss rates*

Loss Given Default is an essential part of credit risk and therefore, received a lot of attention in recent studies, see, e.g., Gambetti et al. (2019); Bellotti et al. (2021); Bastos and Matos (2022). Frequently LGDs are modeled by statistical models like a beta regression, see, e.g., Gambetti et al. (2019) or a local logit regression as proposed in Sopitpongstorn et al. (2021). Other studies focus on machine learning methods and compare several approaches, see, e.g., Loterman et al. (2012); Bellotti et al. (2021). There are only a few publications that combine those two approaches as e.g. in Kellner et al. (2022), which focused on the estimation of quantiles. Research paper II (see Chapter 2) aims to close the gap and combine the well-known (generalized) linear beta regression by Ferrari and Cribari-Neto (2004); Smithson and Verkuilen (2006) and a neural network resulting in the Generalized Beta Regression Neural Network (G-BRANN). This is accomplished by choosing a neural network with two output neurons one for each parameter of the beta distribution and minimizing the corresponding negative log-likelihood as the loss function. This flexible approach allows interactions and non-linearities in the mean and the precision parameter. Since the bounded support of the precision parameter can be challenging to model trainable activation functions in the sense of He et al. (2015) are derived as a novelty to improve robustness and allowing further flexibility.

G-BRANN models the LGDs as a beta distribution with two parameters, mean and precision, using a neural network structure. This has the advantage that non-linearities and interactions are learned without specifying them in advance. This flexible structure improves the fit that leads to the conclusion that non-linearites and interactions play an important role in modeling the LGDs. To reveal these non-linearities and quantify the proportion of non-linearity ALE plots by Apley and Zhu (2020) are used as in the sense of Nagl (2023). For the mean parameter it can be shown that the proportion of non-linearity is 14.10%. The proportion of non-linearity for the precision parameter on the other hand is 80.37%. Therefore, allowing non-linearity is especially important for the precision parameter. Furthermore, the increased flexibility in modeling the distributions allows a more refined distinction between bond characteristics and between macroeconomic states.

Contribution III | *GAMME - Advances in Predictive Mean Matching*

Research paper III (see Chapter 3) refers to the general problem of adequately handling missing data. Many standard missing data approaches rely on linear models. Even though they achieve good results under difficult settings, see, e.g., Kleinke (2017), this can lead to biased results as discussed in Doove et al. (2014). This paper extends one well established imputation approach, predictive mean matching, to allow for non-linearities. This is archived by firstly fitting a neural network to learn the presumably complex structure of the data. The learned non-linearities are revealed by an explainable artificial intelligence method the Accumulated Local Effect Plots. By utilizing the functional decomposition property of the ALE plots the prediction of the neural network can be linearized with respect to the ALE values. Therefore, this ALE values can be used to incorporate non-linearities into an additive structure that can be processed by the predictive mean matching. This extended approach is called the Generalized Adaptive Predictive Mean Matching Estimator (GAMME). Moreover, this method can also be applied to interactions, that are revealed by second order ALE plots. The properties to impute the missing values properly are analyzed in a large simulation study and compared to several other imputation approaches.

The proposed method is evaluated under different simulation settings that includes two different missing data mechanisms Missing Completely at random (MCAR) and Missing at Random (MAR). MCAR is often considered as an unrealistic assumption, but is an essential part of the missing data literature, see, e.g., Liang and Wang (2023). In this setting GAMME performs competitive compared to the best imputation approaches for MCAR. Furthermore, GAMME is challenged under MAR as well. In these conducted simulations GAMME provides the best results considering bias of the regression coefficients and coverage rate out of 12 different imputation approaches. Taking everything into account GAMME is the only approach that leads to good results under MCAR and MAR.

Structure

This thesis is structured by three independent research papers with varying co-authors[1]. Chapter 1 consists of the first paper (*Quantifying uncertainty of machine learning methods for loss given default*) that studies the uncertainty in the LGD prediction. The second paper (*Non-linearity and the distribution of market-based loss rates*) is comprised in Chapter 2, that extends the beta regression to allow non-linearities and interactions and analyzes the effects of that. Chapter 3 is subjected to the third and last paper (*GAMME - Advances in Predictive Mean Matching*). This paper introduces a novel imputation technique to allow valid statistical inference in the presence of non-linearites and interactions. The Conclusion summarizes this thesis, discusses the main results and provides an outlook.

[1] The co-authors and the current state of the research papers are mentioned at the beginning of each chapter.

Chapter 1

Quantifying uncertainty of machine learning methods for loss given default

This chapter is a joint work with Maximilian Nagl[*] and Daniel Rösch[†] and corresponds to a paper published as:
Nagl, M., Nagl, M., & Rösch, D. (2022). Quantifying uncertainty of machine learning methods for loss given default. Frontiers in Applied Mathematics and Statistics, 8, 1076083
https://doi.org/10.3389/fams.2022.1076083

Abstract
Machine learning has increasingly found its way into the credit risk literature. When applied to forecasting credit risk parameters, the approaches have been found to outperform standard statistical models. The quantification of prediction uncertainty is typically not analyzed in the machine learning credit risk setting. However, this is vital to the interests of risk managers and regulators alike as its quantification increases the transparency and stability in risk management and reporting tasks. We fill this gap by applying the novel approach of deep evidential regression to loss given defaults (LGDs). We evaluate aleatoric and epistemic uncertainty for LGD estimation techniques and apply explainable artificial intelligence (XAI) methods to analyze the main drivers. We find that aleatoric uncertainty is considerably larger than epistemic uncertainty. Hence, the majority of uncertainty in LGD estimates appears to be irreducible as it stems from the data itself.

Keywords: Machine Learning, Explainable Artificial Intelligence (XAI), Credit Risk, Uncertainty, Loss Given Default
JEL classification: G21, G32, C45

[*] University Regensburg, Chair of Statistics and Risk Management, 93040 Regensburg, Germany, email: maximilian.nagl@ur.de.

[†] University Regensburg, Chair of Statistics and Risk Management, 93040 Regensburg, Germany, email: daniel.roesch@ur.de.

1.1 Introduction

Financial institutions play a central role in the stability of the financial sector. They act as intermediaries to support the supply of money and lending as well as the transfer of risk between entities. However, this exposes financial institutions to several types of risk, including credit risk. Credit risk has the largest stake with roughly 84% of risk-weighted assets of 131 major EU banks as of June 2021 (European Banking Authority, 2021). The expected loss (EL) due to credit risk is composed of three parameters: Probability of Default (PD), Loss Given Default (LGD), and Exposure at Default (EAD). PD is defined as the probability that a creditor will not comply with his agreed obligations at a later time. LGD is defined as the relative fraction of the outstanding amount that is lost. Finally, EAD is defined as the outstanding amount at the time of default.

This article focuses on LGD as this risk parameter is important for financial institutions not only from a risk management perspective but also for pricing credit risky assets. Financial institutions can use their own models to calculate an estimate for the LGD. This estimate is subsequently used to determine the interest on the loan/bond and the capital requirement for the financial institution itself see, e.g., Altman and Kalotay (2014); Kalotay and Altman (2017); Gambetti et al. (2019); Bellotti et al. (2021) or Kellner et al. (2022). Depending on the defaulted asset, we can divide the LGD further into market-based and workout LGD. The former refers to publicly traded instruments like bonds and is commonly defined as one minus the ratio of the market price 30 days after default divided by the outstanding amount at the time of default. The latter refers to bank loans and is determined by accumulating discounted payments from creditors during the default resolution process. In this article, we use a record of nearly three decades of market-based LGDs gathered from the Moody's Default and Recovery Database starting in January 1990 until December 2019. Recent literature using a shorter history of this data documents that machine learning models due to their ability to account for non-linear relationships of drivers and LGD estimates outperform standard statistical methods, see, e.g., Bastos and Matos (2022); Olson et al. (2021); Sopitpongstorn et al. (2021). Fraisse and Laporte (2022) show that allowing for non-linearity can be beneficial in many risk management applications and can lead to a better estimation of the capital requirements for banks. Therefore, using machine learning models can increase the precision of central credit risk parameters and, as a consequence, could have the potential to yield more adequate capital requirements for banks due to the increased precision.

There is a large body of literature using advanced statistical methods for LGDs. These include beta regression, factorial response models, local logit regressions, mixture regression, and quantile regression among many others, see, e.g., Altman and Kalotay (2014); Kalotay and Altman (2017); Gambetti et al. (2019); Sopitpongstorn et al. (2021); Qi and Yang (2009); Bastos (2010); Bellotti and Crook (2012); Loterman et al. (2012); Qi and Zhao (2011); Tong et al. (2013); Krüger and Rösch (2017); Tomarchio and Punzo (2019). Concerning the increased computational power and methodical progress in academia, machine learning models have become more and more frequently applied concerning LGDs.[1] Early studies by Matuszyk et al. (2010) and Bastos (2010) employ tree-based methods. Moreover, several studies provide benchmark exercises using various machine learning methods, see, e.g., Bellotti and Crook (2012); Loterman et al. (2012); Qi and Zhao (2011). Bellotti et al. (2021) and Kaposty et al. (2020) update previous benchmark studies with new data and algorithms. Nazemi et al. (2021) find text-based variables to be important drivers for marked-based LGDs. Furthermore, evidence that spatial dependence plays a key role in peer-to-peer lending LGD estimation can be found in Calabrese and Zanin (2022). By combining statistical and machine learning models, Sigrist and Hirnschall (2019) and Kellner et al. (2022) show that benefits from both worlds can be captured.

An important aspect, to which the machine learning LGD literature has not yet paid attention, is the associated uncertainty around estimates and predictions[2]. Commonly, we can define two types of uncertainty, aleatoric and epistemic (Der Kiureghian and Ditlevsen, 2009). Following Gawlikowski et al. (2022), aleatoric uncertainty is the uncertainty in the data itself that can not be reduced and is therefore also known as irreducible or data uncertainty. In classical statistics, this type of uncertainty is for example represented by ϵ in the linear regression framework. Epistemic uncertainty refers to the uncertainty of a model due to the (limited) sample size. This uncertainty can be reduced by increasing the sample size on which the model is trained and is therefore also known as reducible or model uncertainty (Gawlikowski et al., 2022). In a linear regression setting, epistemic uncertainty is, accounted for by the standard errors of the beta coefficients. Given a larger sample size, the standard errors should decrease. Recently, the literature on uncertainty estimation has grown rapidly as outlined in a survey article by Gawlikowski et al. (2022).

A first intuitive way to quantify uncertainty is the Bayesian approach, which is also common in classical statistics. However, Bayesian neural networks are computationally expensive and do

[1] Furthermore, several studies use machine learning to estimate PDs, see, e.g., Li and Chen (2021); Petropoulos et al. (2020); Luo et al. (2020); Gunnarsson et al. (2021); Dumitrescu et al. (2022). Concerning mortgage probability of default, see, e.g., Kvamme et al. (2018), Barbaglia et al. (2021), Sadhwani et al. (2021) and Chen et al. (2021). Overall, there is a consensus that machine learning methods outperform linear logit regression.

[2] Gambetti et al. (2019) uses an extended version of the beta regression to model the mean and precision of market-based LGDs. This can be interpreted as focusing on the aleatoric uncertainty. However, the literature using machine learning algorithms lacks uncertainty estimation concerning LGD estimates.

not scale easily to complex neural network architectures containing many parameters. Therefore, other researchers aim at approximating Bayesian inference/prediction for neural networks. Blundell et al. (2015) introduce a backpropagation-compatible algorithm to learn probability distributions of weights instead of only point estimates. They call their approach "Bayes by Backprop." Rather than apply Bayesian principles at the time of training, another strand of literature tries to approximate the posterior distribution only at the time of prediction. Gal and Ghahramani (2016) introduce a concept called Monte Carlo Dropout, which applies a random dropout layer at the time of prediction to estimate uncertainty. Another variant of this framework is called Monte Carlo DropConnect by Mobiny et al. (2021). This variant uses the generalization of Dropout Layers, called DropConnect Layers, where the dropping is applied directly to each weight, rather than to each output unit. The DropConnect approach has outperformed Dropout in many applications and data sets, see, e.g., Mobiny et al. (2021). Another strategy is to use so-called hypernetworks (Krueger et al., 2017). This type of network is a neural network that produces parameters of another neural network (so-called primary network) with random noise input. Finally, the hyper and primary neural networks together form a single model that can easily be trained by backpropagation. Another strand of literature applies an ensemble of methods and uses their information to approximate uncertainty, see, e.g., Lakshminarayanan et al. (2017); Valdenegro-Toro (2019); Wen et al. (2020). However, these approaches are computationally more expensive than Dropout or DropConnect-related approaches. A further strand of literature aims at predicting the types of uncertainty directly within the neural network structure. One of these approaches is called deep evidential regression by Amini et al. (2020) and extensions by Meinert et al. (2022), which learn the parameters of a so-called evidential distribution. This method quantifies uncertainty without extra computations after training. Additionally, the estimated parameters of the evidential distribution can be plugged into analytical formulas for epistemic and aleatoric uncertainty. This approach quantifies uncertainty in a fast and traceable way without any additional computational burden. Because it has many advantages, this article relies on the deep evidential regression framework.

We contribute to the literature in two important ways. First, this article applies an uncertainty estimation framework in machine learning LGD estimation and prediction. We observe that deep evidential regression provides a sound and fast framework to quantify both, aleatoric and epistemic uncertainty. This is important with respect to regulatory concerns. Not only is explainability required by regulators, the quantification of uncertainty surrounding their predictions may be a fruitful step toward the acceptance of machine learning algorithms in regulatory contexts. Second, this article analyzes the ratio between aleatoric and epistemic uncertainty and finds that aleatoric uncertainty is much larger than epistemic uncertainty. This implies that the largest share of uncertainty comes from the data itself and, thus, cannot be reduced. Epistemic uncertainty, i.e., model uncertainty, plays only a minor role. This may explain why advanced methods may outperform simpler ones, but still, the estimation and prediction of LGD remain a very challenging task.

The remainder of this article is structured as follows. Data is presented in Section 1.2, while the methodology is described in Section 1.3. Our empirical results are discussed in Sections 1.4, 1.5 concludes.

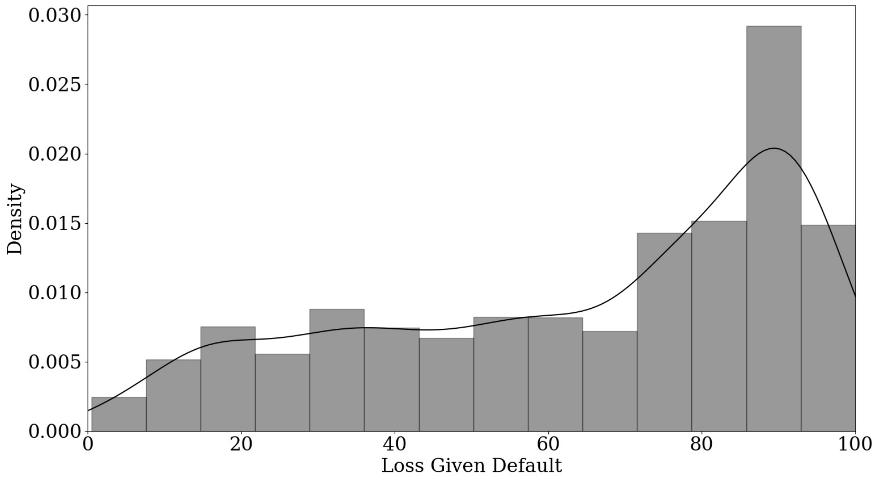
1.2 Data

To analyze bond loss given defaults, we use Moody's Default and Recovery Database (Moody's DRD). This data has information regarding the market-based LGD, default type, and various other characteristics of 1,999 US bonds from January 1990 until December 2019[3]. We use bond characteristics such as the coupon rate, the maturity, the seniority of the bond, and an additional variable, which indicates whether the bond is backed by guarantees beyond the bond issuer's assets. Furthermore, we include a binary variable, which determines if the issuer's industrial sector belongs to the FIRE (finance, insurance, or real estate) sector. To control for differences due to the reason of default, we also include the default type in our analysis. In addition to that, we add the S&P 500 return to control for the macroeconomic surrounding. Consistent with Gambetti et al. (2019), we calculate the US default rates directly from Moody's DRD. To control for withdrawal effects, we use the number of defaults occurring in a given month divided by the number of firms followed in the same period. Since we are interested in the uncertainty in the LGD estimation, we include uncertainty variables. To incorporate financial uncertainty, we use the financial uncertainty index by Jurado et al. (2015) and Ludvigson et al. (2021) which is publicly available on their website. Finally, we include the news-based economic policy uncertainty index provided by Baker et al. (2016), which is also accessible on his website. To keep predictive properties, we lag all macroeconomic variables and uncertainty indices by one-quarter similar to Olson et al. (2021).

Our dependent variable shows a mode at 90%, illustrated by Figure 1.1. This is consistent with Gambetti et al. (2019), who analyzed the recovery rates. The average LGD is about 64.29%

[3] In the original sample with 2,205 bonds, there are 206 bonds with similar LGDs and the same issuer. Since we want to analyze the uncertainty of bonds and not of issuers, we exclude those observations from the data set. However, including these bonds reveals that the uncertainty around their values is considerably smaller, which might have been expected.

Figure 1.1: Histogram of LGDs



as shown in Table 1.1 with a standard deviation of 27.59%. The sample also covers nearly the whole range of market-based LGDs with a minimum of 0.5% and a maximum of 99.99%.

Table 1.1: Descriptive statistics of LGDs across the whole sample

	N	Min.	Median	Mean	Max	St.Dev.	Skewness
LGD	1999	0.50	73.00	64.29	99.99	27.59	-0.59

All displayed values except the sample size and skewness are expressed as percentages.

Table 1.2 lists the variables and data types. In total, we use six bond-related variables, two macroeconomic, and two uncertainty-related variables. The categorical bond-related variables act as control variables for differences in the bond structure.

Table 1.2: Selected variables for the network

Variable	Variable type	Data type
Coupon rate	Bond	Continuous
Maturity	Bond	Continuous
Seniority	Bond	Categorical
Default type	Bond	Categorical
Backed guarantee	Bond	Binary
Industry type	Bond	Binary
S&P 500	Macroeconomic	Continuous
Default rate	Macroeconomic	Continuous
Financial uncertainty	Uncertainty	Continuous
News-based EPU	Uncertainty	Continuous

Table 1.3 shows the correlations between macroeconomic and uncertainty variables. The correlation is moderate to strong across the variables. This must be taken into account when interpreting the effects of the variables. The only exception is the financial uncertainty index and the default rate, which have a very weak correlation.

Table 1.3: Upper triangle of the correlation matrix of macroeconomic and uncertainty features

	S&P500	Default rate	Fin. unc.	News-based EPU
S&P500	100.00	-65.18	-41.69	-65.21
Default rate		100.00	5.25	43.85
Fin. unc.			100.00	51.88
News-based EPU				100.00

All displayed values are expressed as percentages.

Table 1.4 shows descriptive statistics for the seniority of the bond. Each subcategory captures the whole range of LGDs, while the mean and the median of Senior Secured bonds are comparably low. In addition, the Senior Secured bonds have almost no skewness, while the skewness of Senior Unsecured bonds is moderate. The skewness of Senior Subordinated and Subordinated bonds is more negative and fairly similar. Comparing the descriptive statistics across seniority, we observe that the locations of the distributions are different, but the variation of the distribution is considerably large. This might be the first indication of large (data) uncertainty.

Table 1.4: Descriptive statistics of LGDs according to the seniority of the defaulted bond

	N	Min.	Median	Mean	Max	St.Dev.	Skewness
Senior secured	180	0.50	49.75	50.48	99.25	28.87	-0.02
Senior unsecured	1,305	0.50	72.50	63.37	99.97	27.93	-0.53
Senior subordinated	353	0.50	79.0	72.07	99.99	23.97	-0.99
Subordinated	161	0.87	74.0	70.17	99.87	23.74	-0.90

Table 1.5 categorizes the LGDs by their default type, which alters some aspects of the overall picture. Compared to Table 1.1 the categories Distressed Exchange and Others have lower mean and median LGD and positive skewness. The biggest difference between these two categories is that Distressed Exchange has a lower standard deviation. Missed Interest Payment and Prepackaged Chapter 11 show similar descriptive statistics compared to the whole sample in Table 1.1. The last category Chapter 11 has even higher mean and median LGD and the skewness is fairly low.

Table 1.5: Descriptive statistics of LGDs according to the default type

	N	Min.	Median	Mean	Max	St.Dev.	Skewness
Chapter 11	705	0.75	85.00	73.48	99.99	25.46	-1.25
Distressed exchange	322	0.50	40.25	44.51	94.87	24.43	0.18
Missed interest payment	677	1.00	73.50	66.79	99.99	23.98	-0.69
Others	161	1.00	47.00	51.22	99.75	31.38	0.20
Prepackaged chapter 11	134	0.50	76.88	66.58	99.64	28.64	-0.64

1.3 Methods

To model the uncertainty of LGDs, we use a framework called deep evidential regression by Amini et al. (2020). This method is capable of determining the uncertainty of regression tasks and estimating the epistemic and the aleatoric uncertainty. One way to model aleatoric uncertainty in the regression case is to train a neural network with weights w based on the negative log-likelihood of the normal distribution, and thus perform a maximum likelihood optimization. The objective function for each observation is Amini et al. (2020):

$$L_i^L(w) = \frac{1}{2} log(2\pi\sigma_i^2) + \frac{(y_i - \mu_i)^2}{2\sigma_i^2} \tag{1.1}$$

Where y_i is the i-th LGD observation of the sample with size N and μ_i and σ_i^2 the mean and the variance of the assumed normal distribution for observation i. Since μ_i and σ_i^2 are unknown, they can be modeled in a probabilistic manner by assuming they follow prior distributions $q(\mu_i)$ and $q(\sigma_i^2)$. Following Amini et al. (2020), for μ_i a normal distribution and for σ_i^2 a inverse gamma distribution is chosen:

$$\mu_i \sim N(\gamma_i, \sigma_i^2 \nu_i^{-1}) \tag{1.2}$$

$$\sigma_i^2 \sim \Gamma^{-1}(\alpha_i, \beta_i) \tag{1.3}$$

With $\gamma_i \in \mathbb{R}$, $\nu_i > 0$, $\alpha_i > 1$ and $\beta_i > 0$. Factorizing the joint prior distribution $q(\mu_i, \sigma_i^2) = q(\mu_i)q(\sigma_i^2)$ results in a normal inverse gamma distribution:

$$p(\mu_i, \sigma_i^2 | \gamma_i, \nu_i, \alpha_i, \beta_i) = \frac{\beta_i^{\alpha_i}\sqrt{\nu_i}}{\Gamma(\alpha_i)\sqrt{2\pi\sigma_i^2}}(\frac{1}{\sigma_i^2})^{\alpha_i+1} exp\{-\frac{2\beta_i + \nu_i(\gamma_i - \mu_i)^2}{2\sigma_i^2}\} \tag{1.4}$$

This normal inverse gamma distribution can be viewed in terms of virtual observations, which can describe the total evidence Φ_i. Contrary to Amini et al. (2020), we take the suggested definition of the total evidence in Meinert et al. (2022) as $\Phi_i = \nu_i + 2\alpha_i$, because as derived in Meinert and Lavin (2022), the parameters ν_i and $2\alpha_i$ of the conjugated prior normal inverse gamma distribution can be interpreted as virtual observations of the prior distribution, where μ_i and σ_i^2 are estimated from. As a result, the total evidence is the sum of those two expressions. By choosing the negative inverse gamma distribution as the prior distribution, there exists an analytical solution for computing the marginal likelihood or model evidence if the data follows a normal distribution (Amini et al., 2020; Meinert et al., 2022). The marginal likelihood, therefore, follows a student-t distribution:

$$p(y_i|\gamma_i, \nu_i, \alpha_i, \beta_i) = St(y_i; \gamma_i, \frac{\beta_i(1+\nu_i)}{\nu_i \alpha_i}, 2\alpha_i) \tag{1.5}$$

The marginal likelihood represents the likelihood of obtaining observation y_i given the parameter of the prior distribution, in this case, $\gamma_i, \nu_i, \alpha_i$, and β_i. Therefore, maximizing the marginal likelihood maximizes the model fit. This can be achieved by minimizing the negative log likelihood of $p(y_i|\gamma_i, \nu_i, \alpha_i, \beta_i)$. Due to the special conjugated setting with normally distributed data and normal inverse prior distributions, the marginal likelihood can be calculated in a closed form (Amini et al., 2020):

$$L_i^{NLL}(w) = \frac{1}{2} log(\frac{\pi}{\nu_i}) - \alpha_i log(\Omega_i) + (\alpha_i + \frac{1}{2}) log((y_i - \gamma_i)^2 \nu_i + \Omega_i) + log(\frac{\Gamma(\alpha_i)}{\Gamma(\alpha_i + \frac{1}{2})}) \tag{1.6}$$

Such that $\Omega_i = 2\beta_i(1+\nu_i)$ and $\Gamma(.)$ represents the gamma function. This closed-form expression makes deep evidential regression networks fast to compute. To get an accurate estimate of the aleatoric and the epistemic uncertainty the loss function has to be regularized. Contrary to the original formulation of Amini et al. (2020), Meinert et al. (2022) suggest a different regularization term because when using the original formulation the regularized likelihood is insufficient to find the parameters of the marginal likelihood. Therefore, we follow the approach of Meinert et al. (2022) and use the adjusted regularization term. This adjustment scales the residuals by the width of the student-t distribution in Equation (1.5), w_{St_i}, such that the gradients of Φ_i and therefore, of ν_i do not tend to get very large in noisy regions:

$$L_i^R(w) = \left| \frac{y_i - \gamma_i}{w_{St_i}} \right|^p \Phi_i \tag{1.7}$$

With p being the strength of the residuals on the regularization. The loss function for the neural network can therefore be calculated as:

$$L_i(w) = L_i^{NLL}(w) + \lambda L_i^R(w) \tag{1.8}$$

Where λ is a hyperparameter to determine the strength of the regularization in Equation (1.7). Since λ and p have to be determined in advance the network has four output neurons, corresponding to each parameter of the marginal likelihood in Equation (1.5). These parameters can be used to quantify uncertainty. Due to the close connection between the student-t and the normal distribution, w_{St_i} can be used as an approximation for the aleatoric uncertainty (Meinert et al., 2022). Following Meinert et al. (2022), the epistemic and aleatoric uncertainty can be derived as follows:

$$u_{al_i} \equiv w_{St_i} = \sqrt{\frac{\beta_i(1+\nu_i)}{\alpha_i \nu_i}} \tag{1.9}$$

$$u_{ep_i} \equiv \frac{1}{\sqrt{\nu_i}} \tag{1.10}$$

By employing this approach, we assume that our dependent variable, y, follows a normal distribution. LGDs are commonly bound in the interval between zero to one, which is only a part of the space of the normal distribution. Hence, there is the possibility that we obtain predicted values outside this range. However, using the normality assumption is very common in LGD research as the OLS regression is frequently used as the main method or at least as a benchmark to other methods, see, e.g., Bellotti et al. (2021); Qi and Yang (2009); Bellotti and Crook (2012); Loterman et al. (2012); Qi and Zhao (2011); Krüger and Rösch (2017); Kaposty et al. (2020); Jankowitsch et al. (2014); Tobback et al. (2014); Nazemi et al. (2017); Miller and Töws (2018); Nazemi et al. (2018); Starosta (2021). Anticipating the results in Section 1.4, we will see that the predicted values for almost all bonds lie in the interval between zero to one and, thus, our approach produces reasonable estimates. Furthermore, the deep evidential regression approach requires some assumptions to obtain a closed-form solution. For other distributional assumptions, e.g., a beta distribution for the LGD, there is no closed-form marginal likelihood known, which, if used, would eliminate the advantages of this approach.

To unveil the relationships modeled by the neural network, we use Accumulated Local Effect (ALE) plots by Apley and Zhu (2020). ALE plots visualize the average effect of the independent variables on the prediction. Another advantage of ALE plots over other explainable artificial intelligence (XAI) methods is that they are unbiased and fast to compute. As mentioned in Section 1.2, there is a moderate to high correlation between macroeconomic and uncertainty-related variables. Therefore, the XAI method has to be robust to correlations, which is another advantage of ALE plots. For an independent variable $X_j \in \mathbb{R}^{N \times 1}$, the total range of observed values is divided into K buckets. This is accomplished by defining $Z_{j,k}$ as the $\frac{k}{K}$ quantile of the empirical distribution. Therefore $Z_{j,0}$ is the minimum and $Z_{j,K}$ the maximum value of Z_j. Following this approach, $S_{j,k}$ can be defined as the set of values within the left open interval from $Z_{j,k-1}$ to $Z_{j,k}$ with $n_{j,k}$ as the number of observations in $S_{j,k}$. Let $k(X_j)$ be an index that returns the bucket for a value of X_j, then the (uncentered) accumulated local effect can be formalized as:

$$g_{ALE}(X_j) = \sum_{k=1}^{k(X_j)} {n_{j,k}}^{-1} \sum_{i \in S_{j,k}} \left[f(Z_{j,k}, X_{\setminus j,i}) - f(Z_{j,k-1}, X_{\setminus j,i}) \right]. \tag{1.11}$$

$X_{\setminus j} \in \mathbb{R}^{N\times P-1}$ denotes the set of variables without the variable j of P variables and $f(.)$ describes the neural network's output before the last transformation. The minuend in the square brackets denotes the prediction of $f(.)$ if the observation i is replaced with $Z_{j,k}$ and the subtrahend represents the prediction with $Z_{j,k-1}$ instead of observation i. The differences are summed over every observation in $S_{j,k}$. This is done for each bucket k and therefore $g_{ALE}(X_j)$ is the sum of the inner sums weighted by the number of observations in each bucket. In order to get the centered accumulated local effect with mean effect of zero for X_j the $g_{ALE}(X_j)$ is centered as follows:

$$\Theta_{ALE}(X_j) = g_{ALE}(X_j) - N^{-1}\sum_{i=1}^{N} g_{ALE}(X_{j,i}) \tag{1.12}$$

Because of the centering of the ALE plot, the y-axis describes the main effect of Z_j at a certain point in comparison to the average predicted value.

There exist several other XAI methods to open up the black box of machine learning methods. The aim in our article is to investigate non-linear relationships between features and LGD estimates. We therefore decide to use graphical methods. They include partial dependence plots (PDP) by Friedman (2001) for global explanations and individual conditional expectation (ICE) plots by Goldstein et al. (2015) for local explanations. However, the first method especially can suffer from biased results if features are correlated. This is frequently the case for the macroeconomic variables used in our article. We therefore use ALE plots by Apley and Zhu (2020) because they are fast to compute and resolve the problem of correlated features as in our article. Moving beyond graphical methods, there are several other alternatives, such as LIME by Ribeiro et al. (2016) or SHAP by Lundberg and Lee (Lundberg and Lee). However, these methods cannot visualize the potential non-linear relationship between features and LGD estimates. Furthermore, both approaches are known to be problematic if features are correlated and are in some cases unstable, see, e.g., Alvarez-Melis and Jaakkola (2018); Visani et al. (2022). Thus, we use ALE plots by Apley and Zhu (2020) as they are well suited for correlated features. Concerning credit risk, these methods are frequently applied in recent literature. For example, Bellotti et al. (2021) use ALE plots focusing on workout LGDs. Bastos and Matos (2022) compare several XAI methods, including ALE plots as well as Shapley values. Similarly, Bussmann et al. (2020) use SHAP to explain the predictions of the probability of default in fintech markets. Barbaglia et al. (2021) use ALE Plots to determine the drivers of mortgage probability of defaults in Europe. In related fields, such as cyber risk management or financial risk management in general, the application of XAI methods becomes more widespread as well, see, e.g., Giudici and Raffinetti (2022); Babaei et al. (2022); Bussmann et al. (2021); Giudici and Raffinetti (2021).

1.4 Results

1.4.1 Learning strategy

We use the deep evidential regression framework for LGD estimation to analyze predictions as well as aleatoric and epistemic uncertainty. Our data set contains 1,999 observations from 1990 to 2019. To evaluate the neural network on unseen data, which are from different years than the training data, we split the data such that the observations from 2018 to 2019 are reserved as out-of-time data. The remaining data from 1990 to 2017 are split randomly into an 80:20 ratio. A 20% fraction of this data is preserved as out-of-sample data to compare model performance on unseen data which has the same structure. The 80% fraction of this split is the training data. This training data is used to train the model and validate the hyperparameters. Next, the continuous variables of the training data are standardized to adjust the mean of these variables to zero and the variance to one. This scaling is applied to the out-of-sample as well as the out-of-time data with the scaling parameter of the training data. The categorical variables are one hot encoded and one category is dropped. For seniority, Senior Unsecured, and, for the default type, Chapter 11 is dropped and thus act as reference categories. For the guarantee variable and Industry type, we use the positive category as reference. The last preprocessing step includes scaling the LGD values by a factor of 100, such that the LGDs can be interpreted in percentages and enhance computational stability.

After the preprocessing, hyperparameters for the neural network and the loss function have to be chosen[4]. The parameter p of Equation (1.7) is set to 2 to strengthen the effect of the residuals on the regularization, see, e.g., Meinert et al. (2022). The parameter λ is set to 0.001. The analysis is also performed with $\lambda = 0.01$ and $\lambda = 0.0001$, but the differences are negligible. The most commonly used hyperparameters in a neural network are the learning rate, the number of layers, and the number of neurons. To avoid overfitting we included dropout layers, with a dropout rate, which must also be tuned. We use random search to obtain 200 different model constellations and validate them using 5-fold cross-validation. For the random search, we assume discrete or continuous distributions for each hyperparameter. Table 1.6 displays the distributions of the hyperparameters of the neural network. The dropout rate for example is a decimal number, which is usually in the interval between 0, no regularization, and 0.5, strongest regularization. Therefore, we use a continuous uniform distribution to draw the dropout rates.

[4] Amini et al. (2020) provide a python implementation for their paper at `https://github.com/aamini/evidential-deep-learning`.

Furthermore, 20% of the data from the iterating training folds are used for early stopping to avoid overfitting. Each of the five iterations is repeated five times, to reduce the effect of random weight initialization, and averaged. The best model is chosen such that the mean RMSE of the five hold-out fold of cross-validation is the smallest. To determine the number of neurons we use an approach similar to Kellner et al. (2022). As baseline neurons, we use (32, 16) with a maximum of two hidden layers. In this procedure, the multiplier is the factor that scales the baseline number of neurons.[5] As an activation function ReLU is chosen for all hidden units and the network is optimized via Adam. To ensure that ν_i, α_i, and β_i stay within the desired interval, their output neurons are activated by the softplus function, whereby 1 is added to the activated neuron of α_i.

Table 1.6: Setup and final values of the hyperparameter search

Parameter	Distribution	Final parameter
Learning rate	$U^c \sim [0.0001, 0.01]$	0.0029
Dropout rate	$U^c \sim [0.0, 0.50]$	0.4309
Hidden layer	$U^d \sim [1, 2]$	2
Multiple	$U^d \sim [1, 4]$	4

The table shows the ranges for the hyperparameter search. U^c corresponds to the continuous uniform distribution, U^d corresponds to the discrete uniform distribution

The constellation of column three (final parameter) in Table 1.6 is used to form the final network. For that, the network is trained on the training data, 20% of which is used for early stopping. Afterward the trained network is evaluated on the out-of-sample and on the out-of-time data. This procedure is repeated 25 times. Table 1.7 provides the average values and summarizes the evaluation of the different data sets and compares it across different models. Since the loss function in Equation (1.8) depends on λ and p, changes in those parameters result in a loss of comparability.

Table 1.7 compares the neural network from the deep evidential framework to a neural network trained on the mean squared error and to common methods in the literature. These include the linear regression, the transformed linear regression, the beta regression, and the fractional response regression, see, e.g., Bellotti et al. (2021); Loterman et al. (2012). For the transformed linear regression the LGDs are transformed by a logit transformation, which is then used to fit a linear regression. The predictions of this regression are transformed back to their original scale using the sigmoid function. Each model is trained on the same training data. For the neural network trained with mean squared error, the same grid search and cross-validation

[5] For example, if we sample a multiplier of 4 in a two hidden layer network, we have (128, 64).

Table 1.7: Evaluation metrics

Data set	Method	Evidential Loss	RMSE
Training	Evidential neural network	4.1879	0.1813
	Neural network	-	**0.1427**
	Linear regression	-	0.2088
	Transformed linear regression	-	0.2142
	Fractional response regression	-	0.2231
	Beta regression	-	0.2306
Out of Sample	Evidential neural network	4.2574	0.1964
	Neural network	-	**0.1742**
	Linear regression	-	0.2091
	Transformed linear regression	-	0.2100
	Fractional response regression	-	0.2183
	Beta regression	-	0.2328
Out of Time	Evidential neural network	6.1888	0.4241
	Neural network	-	**0.3695**
	Linear regression	-	0.4499
	Transformed linear regression	-	0.4674
	Fractional response regression	-	0.4488
	Beta regression	-	0.3961

For the calculation of the RMSE, the observed LGDs and the predicted LGDs, γ , are rescaled to the original interval from zero to one by dividing the LGDs by 100 to make the RMSE comparable in the literature. The smallest RMSE per data set is printed in bold and the second best is underlined.

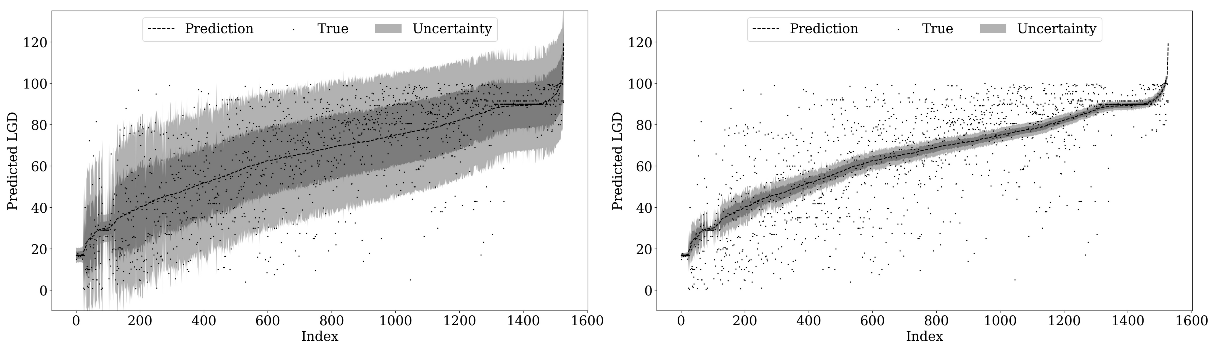
approach with early stopping is used[6]. Since the evidential neural network is the only model with the marginal likelihood as an objective function the evidential loss can only be computed for this model. To compare the evidential neural network with different models, we evaluated the models using the root mean squared error. Note that for computing the root mean squared error only one parameter, γ, is needed since this parameter represents the prediction in terms of the LGD. From Table 1.7, we can see that the neural networks perform best on the training and the out-of-sample data. For the out-of-time data, the beta regression scores second best after the neural network trained with mean squared error, but the difference to the evidential neural networks is on the third digit.

[6] The final parameters of the neural network trained with a mean squared error are very similar in terms of dropout rate (0.4397) and identical for the multiple and the number of hidden layers. The final learning rate (0.0004) is lower than that of the evidential neural network.

1.4.2 Aleatoric and epistemic uncertainty in predictions

The deep evidential regression framework allows us to directly calculate the aleatoric and epistemic uncertainty for every prediction of our neural network. Figure 1.2 shows both types for our estimation sample. The x-axis shows the observation number for the predictions sorted in ascending order. The ordered LGDs are on the y-axis. The dark gray band around the ordered prediction is calculated by adding/subtracting the values of Equations (1.9) and (1.10) on our predictions. The light gray band is obtained by adding/subtracting two times the value of these equations. In the following, we call this "applying one or two standard errors of uncertainty" onto our predictions. The gray dots show the actual observed, i.e., true LGD realizations.

Figure 1.2: Uncertainty estimation in sample



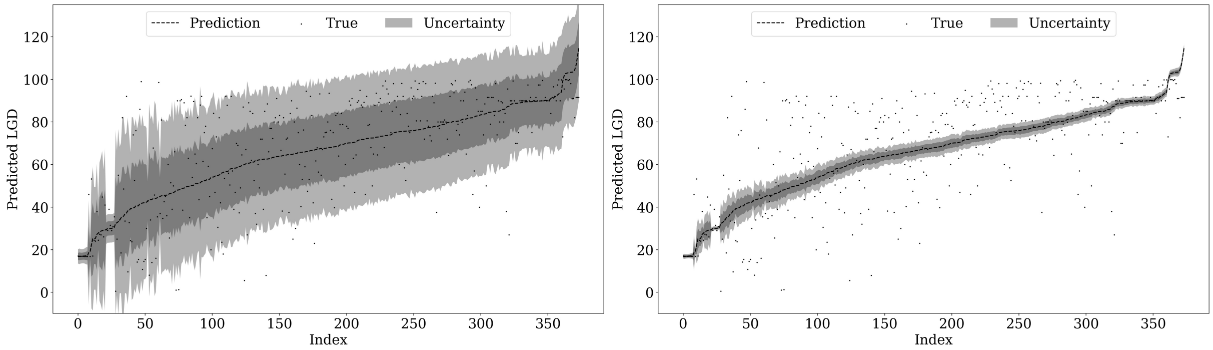
(a) Aleatoric uncertainty

(b) Epistemic uncertainty

Comparing the two plots of Figure 1.2, we observe that the aleatoric uncertainty covers a much larger range around our predictions than the epistemic uncertainty. Almost all true LGD realizations lie within the two standard errors of aleatoric uncertainty. Hence, the irreducible error or data uncertainty has the largest share of the total uncertainty. Recall that market-based LGDs are based on market expectations as they are calculated as 1 minus the traded market price 30 days after default. Therefore, the variation of the data also depends on market expectations which are notoriously difficult to estimate and to a large extent not predictable. Thus, it is reasonable that the aleatoric uncertainty is the main driver of the overall uncertainty. In contrast, the epistemic uncertainty, i.e., the model uncertainty, is considerably lower. This may be attributed to our database. This article covers nearly three decades including several recessions and upturns. Hence, we cover LGDs in many different points of the business cycle and across many industries and default reasons. Therefore, the data might be representative for the data generating process of market-based LGDs. Hence, the uncertainty due to limited sample size is relatively small in our application.

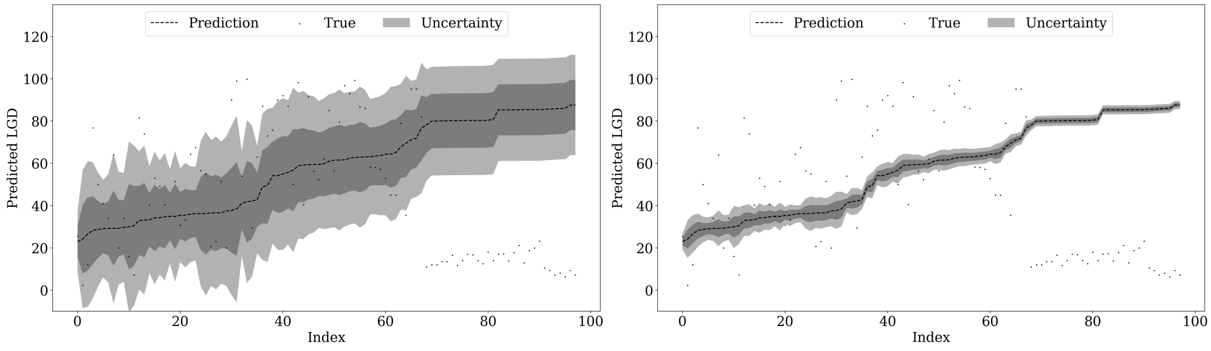
As we model all parameters of the evidential distribution dependent on the input features, we can also predict uncertainty for predictions in out-of-sample and out-of-time samples. Comparing Figures 1.2, 1.3 one might have expected that the epistemic uncertainty is increasing due to the lower sample size and the usage of unseen data. However, the functional relation of the epistemic uncertainty is calibrated on the estimation sample and transferred via prediction onto the out-of-sample data. Hence, if the feature values do not differ dramatically, the predicted uncertainty is similar. Only if we observe new realizations of our features in unexpected (untrained) value ranges, the uncertainty prediction should deviate strongly. Thus, we may use the prediction of the uncertainty also as a qualitative check of structural changes.

Figure 1.3: Uncertainty estimation out-of-sample


(a) Aleatoric uncertainty

(b) Epistemic uncertainty

Figure 1.4: Uncertainty estimation - out-of-time


(a) Aleatoric uncertainty

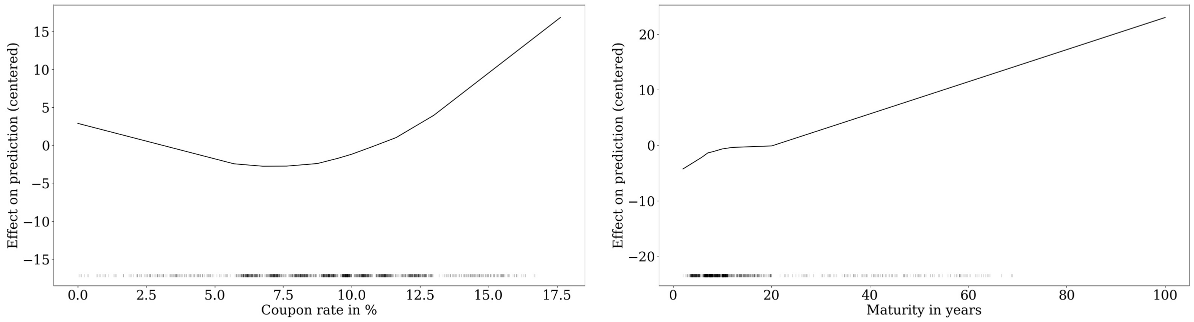
(b) Epistemic uncertainty

Structural changes in LGD estimation are primarily due to changes over time. This is one reason why some researchers argue to validate forecasting methods especially on out-of-time data sets, see, e.g., Kalotay and Altman (2017); Olson et al. (2021). In our application, there is no qualitative sign of structural breaks via diverging uncertainty estimates in 2018 and 2019. Comparing Figure 1.4 with the former two, we observe a similar pattern. This might have been expected as the out-of-time-period is not known for specific crises or special circumstances. Comparing the course of the epistemic uncertainty in all three figures, we observe that the uncertainty bands become smaller as the predicted LGD values increase. This implies that the neural network becomes more confident in predicting larger LGDs. Comparing this course with the histogram in Figure 1.1, one explanation for that might be the considerably larger sample size on the right-hand side. As we observe larger LGDs in our sample, the epistemic uncertainty in this area decreases.

1.4.3 Explaining LGD predictions

In this subsection, we take a deep dive into the drivers of the mean LGD predictions. As outlined in Section 1.3, we use ALE Plots to visualize the impact of our continuous features. We choose $K = 10$ buckets for all ALE plots. Overall we have three different sets of drivers. The first one consists of bond specific variables, subsequently we investigate drivers that reflect the overall macroeconomic developments and finally we follow Gambetti et al. (2019) and include uncertainty-related variables. Evaluating the feature effects is important to validate that the inner mechanics of the uncertainty-aware neural network coincide with the economic intuitions. This is of major concern if financial institutions are tempted to use this framework for their capital requirement calculation. The requirement of explaining employed models is documented in many publications of regulatory authorities, see, e.g., Bank of Canada (2018); Bank of England (2019); Basel Committee on Banking Supervision (2019); Deutsche Bundesbank (2020).

Figure 1.5: Bond-related drivers

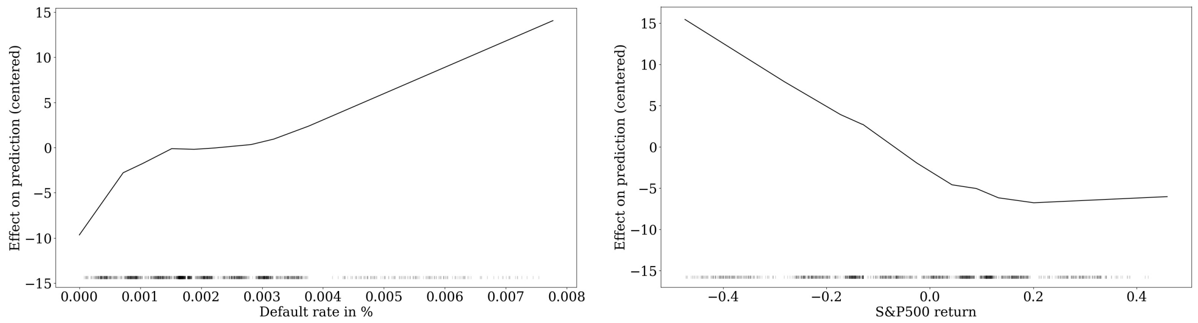


(a) Coupon rate

(b) Maturity

Figure 1.5 shows the feature effect of bond-related drivers. The feature value range including a rugplot to visualize the distribution of the feature is shown on the x-axis. The effect of the driver on the LGD prediction is shown on the y-axis. We observe on the left-hand side of Figure 1.5, a negative effect of the coupon rate up to a value of roughly 8%. This negative relation seems plausible as higher coupon rates may also imply higher reflows during the resolution of the bond and, thus decreases the Loss Given Default. The relation starts to become positive after 8%, which may be explained by the fact that a higher coupon rate also implies higher risk and, thus the potential reflow becomes more uncertain. Maturity has an almost linear and positive relation with the predicted LGD values. In general, the increase in LGD with longer maturity is explained by sell-side pressure from institutional investors which usually hold bonds with a longer maturity, see, e.g., Jankowitsch et al. (2014). These relations were also confirmed by Gambetti et al. (2019), who find that bond-related variables have a significant impact on the mean market-based LGD.

Figure 1.6: Macroeconomy-related drivers



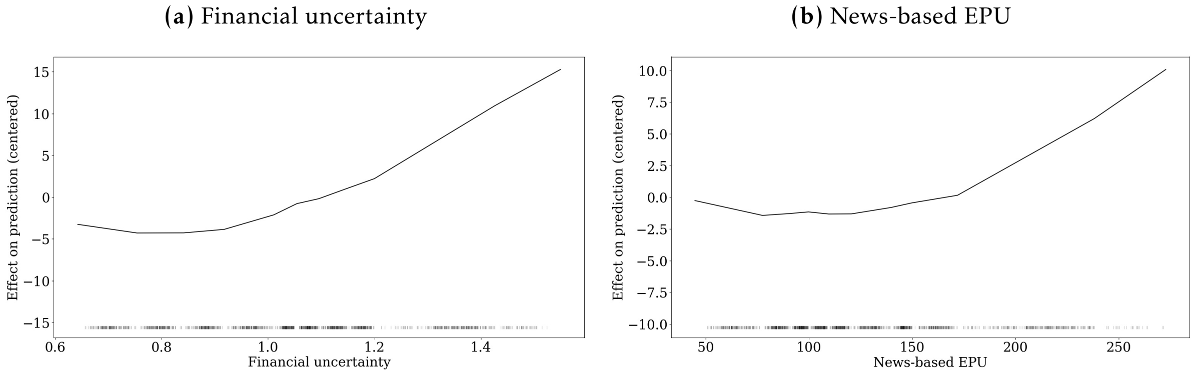
(a) Default rate

(b) S&P 500 return

With regard to features that describe the macroeconomic surrounding, Figure 1.6 shows their effect on the LGD prediction. The default rate is one of the best-known drivers of market-based LGDs and is used in various studies, see, e.g., Kalotay and Altman (2017); Gambetti et al. (2019); Nazemi et al. (2021, 2018). The increasing course reflects the observation that LGDs tend to be higher in recession and crisis periods than in normal periods. This empirical fact also paves the way for generating so-called downturn estimates which should reflect this crisis behavior. These downturn estimates are also included in the calculation of the capital requirements for financial institutions, see, e.g., Calabrese (2014); Betz et al. (2018) or for downturn estimates of EAD see Betz et al. (2022). Similarly, we observe a negative relation of predicted LGDs and S&P 500 returns, implying that LGDs increase if the returns become negative. Interestingly, positive returns have little impact on LGD predictions, which again, reinforces the downturn character of LGDs.

Consistent with Gambetti et al. (2019), who were the first to document the importance of uncertainty-related variables in the estimation of LGDs, we include two frequently used drivers as well, shown in Figure 1.7. Financial uncertainty proposed by Jurado et al. (2015) and the

Figure 1.7: Uncertainty-related drivers


(**a**) Financial uncertainty

(**b**) News-based EPU

News-based EPU index by Baker et al. (2016), which cover uncertainty based on fundamental financial values and news articles. Both show a rather flat course from the low to mid of their feature value range. However, there is a clear positive impact on LGDs when the uncertainty indices reach high levels. Again, this reinforces the crisis behavior of market-based LGDs. The importance of uncertainty-related variables is also confirmed by Sopitpongstorn et al. (2021) who find a significant impact as well. In a similar sense, Nazemi et al. (2021) use news text-based measures for predicting market-based LGDs and underlining their importance. To summarize, recent literature suggests that uncertainty-related variables should be used to include all kinds of expectations of the economics surrounding the model framework.

1.5 Conclusion

Uncertainty estimation has become an active research domain in statistics and machine learning. However, there is a lack of quantification of uncertainty when applying machine learning to credit risk. This article investigates a recently published approach called Deep Evidential Regression by Amini et al. (2020) and its extension by Meinert et al. (2022). This uncertainty framework has several advantages. First, it is easy to implement as one only has to change the loss function of the (deep) neural network and sightly adjust the output layer. Second, the predicted parameters of the adjusted network can easily be turned into mean prediction, aleatoric uncertainty, and epistemic uncertainty predictions. There are virtually no additional computational burdens to calculate predictions and their accompanying uncertainty. Third, the overall computational expense is much lower compared to approaches like Bayesian neural networks, ensemble methods, and bootstrapping. Furthermore, deep evidential regression belongs to a small class of frameworks which allow a direct, analytical disentangling of aleatoric and epistemic uncertainty. With these advantages, this framework may also be suitable for applications in financial institutions to accompany the usage of explainable artificial intelligence methods with quantification of aleatoric and epistemic uncertainty. Moreover, it is possible to include other variables, such as firm-specific financial risk factors, or to focus on non-listed companies. Further applications may also include the prediction of risk premiums in other asset pricing or forecasting the sale prices of real estate. Moreover, in other areas where predictions are critical such as health care, the quantification of prediction uncertainty may allow a broader application of machine learning methods.

This article uses almost 30 years of bond data to investigate the suitability of deep evidential regression on the challenging task of estimating market-based LGDs. The performance of the uncertainty-aware neural network is comparable to earlier literature and, thus, we do not see a large trade-off between accuracy and uncertainty quantification. This paper documents a novel finding regarding the ratio of aleatoric and epistemic uncertainty. Our results suggest that aleatoric uncertainty is the main driver of the overall uncertainty in LGD estimation. As this type is commonly known as the irreducible error, this gives rise to the conjecture that LGD estimation is notoriously difficult due to the high amount of data uncertainty. On the other hand, epistemic uncertainty that can be reduced or even set to zero with enough data plays only a minor role. Hence, the advantage of more complex and advanced methods, like machine learning, may be limited. However, this may not hold for all LGD data sets or if we look at different parts or parameters of the distribution other than the mean. Therefore, we do not argue that our results should be generalized to all aspects of LGDs, but are the first important steps to investigating the relation of aleatoric and epistemic uncertainty. Overall, understanding the determinants of both uncertainties can be key to getting a deeper understanding of the underlying process of market-based LGDs and, thus is certainly a fruitful path of future research.

Acknowledgements
We would like to thank the referees for their comments and suggestions that have substantially improved the article.

Chapter 2

Non-linearity and the distribution of market-based loss rates[*]

This chapter is a joint work with Maximilian Nagl[†] and Daniel Rösch[‡] and corresponds to a working paper with the same name (submitted to *OR Spectrum*, revised and resubmitted).

Abstract
We synthesize the extended linear beta regression with a neural network structure to model and predict the mean and precision of market-based loss rates. We can incorporate non-linearity in mean and precision in a flexible way and resolve the problem of specifying the underlying form in advance. As a novelty, we can show that the proportion of non-linearity for the mean estimates is 14.10% and 80.37% for the precision estimates. This implies that especially the shape of the loss rate distribution entails a large amount of non-linearity and, thus, our approach consistently outperforms its linear counterpart. Furthermore, we derive trainable activation functions to allow a data-driven estimation of their shape. This is important if predictions have to be in a certain interval, e.g., $(0,1)$ or $(0,\infty)$. Conducting a scenario analysis, we observe that our estimated distributions are more refined compared to traditional models, thereby demonstrating their suitability for risk management purposes. These estimated distributions can assist financial institutions in better identifying diverse risk profiles among their creditors and across various macroeconomic states.

Keywords: Loss Given Default; Machine Learning; Explainable Artificial Intelligence (XAI); Distribution

JEL classification: G21, G32, C45, C51

[*] We would like to thank the department editor, the associate editor and two anonymous referees for their comments and suggestions, which have substantially improved the paper.

[†] University Regensburg, Chair of Statistics and Risk Management, 93040 Regensburg, Germany, email: maximilian.nagl@ur.de.

[‡] University Regensburg, Chair of Statistics and Risk Management, 93040 Regensburg, Germany, email: daniel.roesch@ur.de.

2.1 Introduction

The current economic climate is characterized by different challenges. Financial markets have experienced significant turmoils recently due to global economic uncertainty, geopolitical tensions, fluctuations in currency exchange rates and a different interest rate environment. The European Banking Authority (2022) states that these factors also impact financial institutions due to increased funding costs for banks and an overall asset quality deterioration. These circumstances put banks' internal risk management again into the focus of regulators and politics. The largest share of risk a bank faces is determined by its credit risk. Following the latest data, credit risk accounts for 83.3% of risk-weighted assets of 131 major EU banks as of June 2022, underlining its importance (European Banking Authority, 2022). The expected loss (EL) of credit risk related assets can be split into three components. Probability of Default (PD) quantifies the probability that a obligor will not meet his agreed obligations. Exposure at Default (EAD) defines the amount of outstanding obligations. Finally, Loss Given Default (LGD) refers to the percentage share of outstanding debt that is lost, given the obligor defaults. This paper focuses on so-called market-based LGDs, which are relevant for publicly traded instruments such as bonds. They are defined as one minus the ratio of the market price 30 days after default over the outstanding amount. Especially market-based LGDs entail challenging characteristics such as bounded support, skewed distribution, and heteroskedasticity (Gambetti et al., 2019). Moreover, the estimation of market-based LGDs has also macro-economic implications. According to SIFMA Research (2022) the US corporate bond market has a volume of 10 trillion USD and therefore the estimation of the LGDs can be essential for the financial stability of the economy. The Basel Accord allows banks to use their own models to provide estimates for LGDs as well as the other components of the expected loss (Basel Committee on Banking Supervision, 2017). Therefore, academia aims at providing guidance on how to estimate LGDs and which methods to use, see, e.g., Altman and Kalotay (2014); Kalotay and Altman (2017); Bellotti et al. (2021); Kellner et al. (2022) or Gürtler and Zöllner (2023).

In this paper we utilize market-based LGDs sourced from Moody's Default and Recovery Database spanning from January 1990 to March 2021. This dataset stands out as the most comprehensive when focusing on market-based LGDs. It comprises a range of LGD-specific variables, including seniority, industry sector, and default type, among other pertinent information. Additionally, we extend this dataset by incorporating commonly used macroeconomic and uncertainty-related variables described in the literature. Past literature uses classical statistical models, such as fractional response regression, to predict LGDs using this data, see, e.g., Bastos (2010). These methods commonly focus only on the mean LGD predictions, neglecting the challenging characteristics of LGD distributions. Furthermore, most recent literature reinforces the evidence of non-linearity of drivers for mean market-based LGD estimates using machine learning models, see, e.g., Bastos and Matos (2022); Olson et al. (2021); Bellotti et al. (2021) or Sopitpongstorn et al. (2021). Most of these methods have in common that only the conditional *mean* is estimated. Although machine learning models increase the flexibility of the modelling framework and are capable of incorporating some distributional aspects, none of the studies explicitly account for these aspects nor investigate their drivers and potential impact.

Making decisions based only on one location parameter, e.g., by using standard machine learning algorithms, may not be holistic in the sense that further distributional characteristics carry important information. For example the dispersion can enhance the understanding of the underlying mechanics between drivers and the whole distribution and be supportive of the managerial decision process. However, they are frequently neglected in the literature.[1] For example, different parts of the distribution can be interpreted as scenarios for banks and risk managers. In that sense, lower quantiles of the distribution can be interpreted as good scenarios for banks, i.e., a low loss realization. On the contrary, higher quantiles imply higher losses. Hence, risk managers can conduct a scenario analysis to investigate how their loan portfolios face losses in adverse, normal, and good scenarios based on the individual loss distribution of their obligors. This enables managers also to reveal different risk profiles among obligors based on their individual distribution. As future realizations are unknown, comparing (predicted) quantiles can provide risk managers with a good indication of how likely low and high LGD realizations can realize and how individual obligors compare to each other. Against this background, we suggest jointly modeling mean and precision, i.e., a dispersion parameter, to allow for varying shapes of the distribution. This paper doesn't aim for a competitive horse race of various methods but contributes by proposing a novel method for explaining market-based LGDs. It focuses on understanding the factors influencing LGDs mean and precision, an area that has been underexplored so far. Rather than predicting future mean LGDs, it uncovers hidden relationships in the LGD distribution, aiding scenario analysis and deriving implications. Thus, it is designed as a non-parametric tool for exploring these connections.

We contribute to the literature in four important ways. First, we combine beta regression by

[1] There are studies which include distributional characteristics of LGD into their modeling strategy, see, e.g. Calabrese (2014); Altman and Kalotay (2014); Kalotay and Altman (2017); Krüger and Rösch (2017); Betz et al. (2018); Hwang and Chu (2018); Hwang et al. (2020) or Kellner et al. (2022). However, they do not explicitly model the drivers of different parameters of the LGD distribution.

Ferrari and Cribari-Neto (2004) and its extension by Smithson and Verkuilen (2006) and Simas et al. (2010) with artificial neural networks. This contributes to the literature on market-based LGDs, as the combination is the first to allow non-linearity in the mean and precision of the LGD distribution. It extends the work of Gambetti et al. (2019) as we detect a large amount of non-linearity especially in the shape of the distribution. Furthermore, our combination achieves a substantially better performance in and out-of-sample compared to its linear counterpart. Hence, non-linearity plays an important role for the mean and precision of the loss rates.
Second, we use Accumulated Local Effects (ALE) plots by Apley and Zhu (2020) to unveil the non-linear relationships in mean and precision. Furthermore, we quantify the amount of non-linearity in the mean and precision estimation according to Apley and Zhu (2020) and Nagl (2023). To the best of our knowledge, this paper is the first to incorporate, visualize and quantify non-linearity in the precision of market-based loss rates.
Third, we derive trainable activation functions similar to He et al. (2015) to increase the robustness of predictions, especially for unseen data. The trainable activation function offers the neural network a data-driven way of estimating the shape of final predictions, which is new to the finance and credit risk literature. In general, this contribution is not restricted to our discipline but may be beneficial for any other field of research where bounded outputs play an important role, e.g., demand or sales forecasting.
Fourth, we find the that accounting for non-linearity and interactions the modeled distributions differ compared to the beta regression such that they can be better distinguished. This enables risk managers and regulators to better quantify the individual risk of obligors in a straightforward and interpretable way.

The remainder of this paper is structured as follows. In Section 2.2, we give a summary of the relevant literature of LGD estimation. Data is presented in Section 2.3, while the methodology is described in Section 2.4. Our empirical results are discussed in Section 2.5 and Section 2.6 concludes.

2.2 Literature review

Concerning the special characteristics of LGD distributions, mainly advanced statistical methods are applied. These include for example beta regression, mixture regression, and factorial regression among many others, see, e.g., Gambetti et al. (2019) or Sopitpongstorn et al. (2021) as recent examples. Due to the increased computational power and progress in academia, machine learning models become more and more apparent. They were used by academics at the beginning, but become increasingly applied by practitioners and heavily discussed by regulators. However, the vast majority of these studies focus only on mean estimates of LGDs, see Bastos and Matos (2022); Kellner et al. (2022) or Gürtler and Zöllner (2023) for a recent overview. Nagl et al. (2022) focus on the uncertainty quantification of machine learning models for market-based LGDs and finds that the aleatoric uncertainty, i.e., uncertainty in the data itself, is more significant than epistemic uncertainty, i.e., uncertainty due to a limited sample size. This study emphasises the importance of estimating LGD distributions instead of just point estimates. Focussing on the distribution of LGDs is not entirely new to the literature. Closely related to our paper is Gambetti et al. (2019) who use a linear generalized beta regression to model the distribution of market-based LGDs. However, they do not consider non-linearity in mean or precision in their model. Furthermore, related to the contributions of our paper are Krüger and Rösch (2017) and Kellner et al. (2022). They both estimate quantile regression or machine learning-based extensions thereof, focusing on workout LGDs. They emphasize the importance of accounting for distributional aspects in these LGDs. Their approaches require a sizable dataset to reliably estimate multiple quantiles to describe a full distribution. Furthermore, some papers use classical statistical models, such as mixture models for LGDs. Altman and Kalotay (2014), Calabrese (2014), Kalotay and Altman (2017), and Betz et al. (2018) utilize these mixture models, consisting of a combination of different distributions, to disaggregate the estimation of a full conditional distribution into subparts. However, with an increasing number of components, these models become less interpretable, and the drivers of LGDs are linearly connected to the components, reducing flexibility. A novel approach proposed by Gürtler and Zöllner (2023) suggests that the modality type of workout LGDs is crucial in determining the best estimation method. Using cluster analysis, they identify three clusters/modality types suitable for their sample. While this approach outperforms traditional models, it requires a sizable dataset, as the entire dataset is divided into clusters, resulting in less data to fit various models. Overall, the methods proposed to account for distributional aspects necessitate a considerable dataset size, making them especially useful for workout LGDs, where data is typically much larger compared to market-based LGDs as used in our paper.

Recent literature on LGD estimation has seen significant growth, particularly with the widespread application of machine learning models. However, several gaps remain. Firstly, while many studies concentrate on mean LGD predictions, they often overlook distributional characteristics, including aleatoric uncertainty, identified as a primary source of uncertainty in LGD estimation by Nagl et al. (2022). Moreover, approaches addressing distributional aspects are typically only viable for large datasets, such as workout LGDs, rather than market-based LGDs. Hence, this paper aims to bridge this gap by synthesizing existing evidence and proposing a suitable method for market-based LGDs. Secondly, there is a notable absence of studies focusing on potential non-linearity in the shape of LGD distributions. Such non-linearities could significantly impact scenario analysis by allowing for different distributional shapes across categories.

2.3 Data

We use bond loss given defaults from Moody's Default and Recovery Database (Moody's DRD). The examined data contain 2,315 market-based LGDs and related bond characteristics ranging from January 1990 until March 2021. Finding suitable drivers of market-based LGD's mean and precision is a challenging task. Gambetti et al. (2019) synthesizes the evidence of the literature on important predictor variables. We follow Gambetti et al. (2019) and use the same features as a starting point. The variables can be divided into three subgroups, bond characteristics, macroeconomic, and uncertainty related determinants. The bond specific characteristics are coupon rate, maturity, the seniority of the bond as well as the issuer's industrial sector. Furthermore, we include the severity of the default type, the defaulted amount and a dummy variable which indicates whether the bond is backed by guarantees. We use several macroeconomic related variables. These include the industrial production returns computed monthly, the S&P 500 returns[2] as well as the recession indicator[3] provided by the National Bureau of Economic Research. Furthermore, delinquency rates in commercial and industrial loans[4] are included quarterly. Following Gambetti et al. (2019), we gather the American default rates from Moody's database and control for withdrawal effects by using the number of defaults registered in a given month divided by the number of firms followed in the same period. We include both rates because delinquency is commonly used if a borrower misses a single payment. Default is usually triggered when a borrower fails to keep up with the loan repayments agreed upon or in some other way fails to fulfill the terms of the loan. Hence, both indices focus on financial distress, but vary in degree and time dimensionality. The third set of variables reflects different types of uncertainty. This may be of particular interest when the focus is on the uncertainty around the estimated means, modeled via their precision. Therefore, we include the VIX[5], as a proxy for the uncertainty in the stock market. To reflect financial uncertainty

[2] https://fred.stlouisfed.org/series/SP500

[3] https://fred.stlouisfed.org/series/USREC

[4] https://fred.stlouisfed.org/series/DRALACBS

[5] https://fred.stlouisfed.org/series/VIXCLS

the financial uncertainty index[6] derived by Jurado et al. (2015) and Ludvigson et al. (2021) is added. Furthermore, we take the policy uncertainty into account by extending the data set with the news-based economic policy uncertainty[7] provided by Baker et al. (2016). The last two uncertainty measures are uncertainty based on forecast dispersion of the consumer price index[8] to reflect the inflation and the expenditures of federal and state/local purchases[9]. Those rely on the dispersion of forecasts computed from the Federal Reserve Bank of Philadelphia's Survey of Professional Forecasters. For further details on the variables, we refer to Gambetti et al. (2019). Similar to Olson et al. (2021) all macroeconomic variables and uncertainty indices are lagged by one quarter to ensure predictive properties.

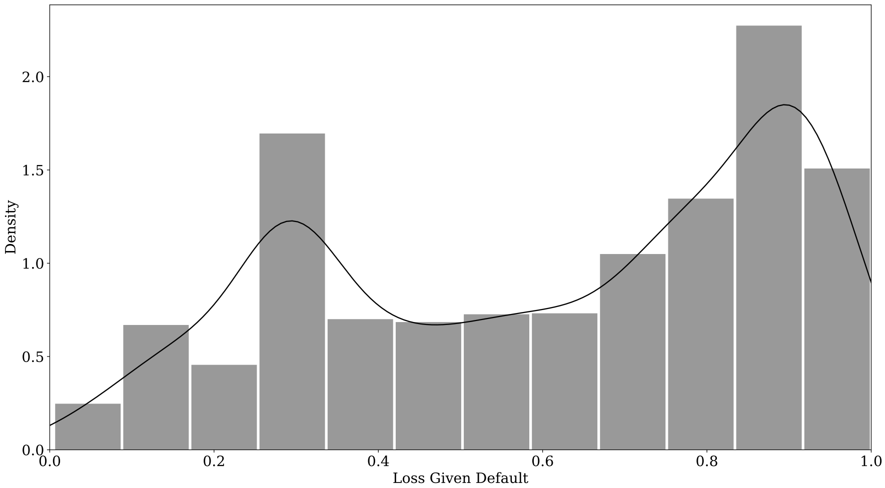


Figure 2.1: Histogram of LGDs

Similar to Görgen et al. (2022) Figure 2.1 shows the slightly negatively skewed bimodal distribution of the realized market-based LGDs in our sample. The average LGD is 61.40%. The lowest LGD is only half a percent, while the highest one is close to 100%. Overall, we recover the stylized empirical features of bond-related LGDs such as bounded support and skewed distribution.

Taking a closer look at the correlations of the uncertainty measures in Table 2.1, one can observe that the highest correlation is between VIX and financial uncertainty with a value of 76.64%. The other correlations are moderate to low.

[6] https://www.sydneyludvigson.com/macro-and-financial-uncertainty-indexes

[7] http://www.policyuncertainty.com/global_monthly.html

[8] https://www.philadelphiafed.org/surveys-and-data/real-time-data-research/dispersion-forecasts

[9] https://www.philadelphiafed.org/surveys-and-data/rfedgov

Table 2.1: Upper triangle of the correlation matrix of uncertainty features in percentages

	VIX	Fin. unc.	News-based EPU	CPI unc.	F.S.L. exp. unc.
VIX	100.00	76.64	40.14	41.24	12.34
Fin. unc.		100.00	27.17	47.54	-8.59
News-based EPU			100.00	29.48	-0.79
CPI unc.				100.00	7.62
F.S.L. exp. unc.					100.00

Notes: All displayed values are expressed as percentages.

As this database is often used for investigating market-based LGDs, different periods are frequently used in the literature, see, e.g. Altman and Kalotay (2014); Kalotay and Altman (2017); Hwang and Chu (2018); Gambetti et al. (2019); Hwang et al. (2020); Sopitpongstorn et al. (2021) just to name a few.[10] Therefore, stylized facts such as the LGD increasing for lower seniority and differences for average LGDs across industries are well known. Hence, we move the discussion of these facts to Appendix 2.A. Table 2.A.1 displays an overview of summarizing statistics for the total dataset. Tables 2.A.2, 2.A.3 and 2.A.4 give an overview of the descriptive statistics according to the seniority, industry sector and the default type.

2.4 Methods

The distribution of LGDs ranges from 0 to 1 and can be skewed and multimodal. As a starting point, we rely on the beta regression because of its flexibility and the fact that the distributional assumption matches the range of the LGDs. We use the alternative definition with two parameters $0 < \mu < 1$ and $\phi > 0$ to model the LGD, Y, with support $0 < Y < 1$. μ corresponds to the mean of Y, whereas ϕ is the precision parameter. Following Ferrari and Cribari-Neto (2004) the density of the beta distribution is:

$$f(y;\mu,\phi) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\cdot\Gamma((1-\mu)\phi)} \cdot y^{\mu\phi-1} \cdot (1-y)^{(1-\mu)\phi-1}, \tag{2.1}$$

where $\Gamma(.)$ denotes the Gamma function. The parameters μ and ϕ can be linked to the first two

[10] The actual number of observation varies between these studies as they do not use the same set of bond characteristics. For example, Hwang and Chu (2018); Hwang et al. (2020) and Sopitpongstorn et al. (2021) report a higher number of observations by not considering the coupon rate, contrary to Gambetti et al. (2019) and our approach. However, the stylized descriptive facts remain comparable.

central moments of Y by:

$$E(Y) = \mu \tag{2.2}$$

$$Var(Y) = \frac{\mu(1-\mu)}{1+\phi} \tag{2.3}$$

By Equation (2.3) we see, that for a fixed μ the variance decreases in ϕ. Especially the extension to a generalized form, where the precision ϕ can be modeled as a dependent variable in addition to the mean μ, is useful to analyze LGD uncertainty. To model the density in Equation (2.1) one can formulate a regression model called the beta regression as stated by Ferrari and Cribari-Neto (2004); Smithson and Verkuilen (2006); Simas et al. (2010). In this regression model, the input variables are weighted by their regression coefficients and transformed to match the desired support of μ and ϕ. Usually for μ the logistic and for ϕ the exponential function is used. The regression coefficients are estimated by maximum likelihood optimization. In this approach, μ and ϕ are modeled as a function of the explanatory variables. We maximize the sum of the log-likelihood over N bonds of Equation (2.1), where y_i represents the LGD of the i-th of N bonds[11]:

$$\begin{aligned}LL(y_1,...,y_N;\mu_1,...,\mu_N,\phi_1,...,\phi_N) = \sum_{i=1}^{N}(log(\Gamma(\phi_i)) - log(\Gamma(\mu_i\phi_i)) - log(\Gamma(1-\mu_i)\phi_i) + \\ (\mu_i\phi_i - 1)y_i + ((1-\mu_i)\phi_i - 1)(1-y_i).\end{aligned} \tag{2.4}$$

Despite the flexibility of the beta regression, it is limited in that the relationships between the predictors and the dependent variable have to be specified beforehand. To resolve this restriction we propose the Beta Regression Artificial Neural Network (BRANN) and its extension, the Generalized Beta Regression Artificial Neural Network (G-BRANN).

Modeling approaches of beta regression neural networks

The potential modeling approaches to connect neural networks with the beta regression can be divided into three cases. First, if the input variables are the same for both outputs, μ and ϕ, a vanilla feed-forward neural network with two output neurons can be used[12]. This allows interactions of the input variables between μ and ϕ. Hence, one would assume, that μ and ϕ can be explained by the exact same variables, which can interact with each other. Second, this can be relaxed by using skip connections, see, e.g., He et al. (2016). Hence, the input variables of one

[11] Because some of the explanatory variables are bond-specific for each observation i, i=1,...,N, we also subscript μ_i and ϕ_i in the log-likelihood.

[12] For sparsity of notation, we skip subscripts for μ and ϕ here.

parameter are a subset of the other and one would assume that μ and ϕ are driven by the same dynamics. Third, the input variables for the two parameters can be different and do not need to share any input variables. So each parameter can be modeled by a separate neural network, a μ-sub-network and a ϕ-sub-network, and then merged together. This is the most flexible approach, and, thus, we follow this path. In BRANN, which is based on a beta regression, ϕ is modeled as a constant. So the ϕ -sub-network can be represented by a neural network with a constant as input and no hidden layers. Similar to the beta regression, ϕ has to be a positive value, thus, we impose a transformation $\zeta(.)$ which is the exponential function following Ferrari and Cribari-Neto (2004).

BRANN

The μ-sub-network can consist of L layers, $l = 1,...,L$ with K_l neurons each. The first layer is called the input layer. This layer takes the input matrix $X \in \mathbb{R}^{N\times p}$, which typically consists of N observations with p exogenous features, and feeds every observation into the network. Each layer l takes as input the output of the previous layer $o_{l-1} \in \mathbb{R}^{K_{l-1}\times 1}$ and weights it by multiplying it with a weights matrix $W_l \in \mathbb{R}^{K_l \times K_{l-1}}$ and adding a bias vector $b_l \in \mathbb{R}^{K_l\times 1}$. The weighted output of the previous layer is activated by a non-linear activation function $\psi(.)$

$$o_l = \psi(W_l o_{l-1} + b_l) \tag{2.5}$$

This produces the output o_l of the current layer l. For the last layer L the weighted output of the previous layer is activated by a function $\iota(.)$. Since the desired range for μ is between 0 and 1, the logistic function is chosen for $\iota(.)$ to ensure the output of the μ-sub-network, o_μ, stays in the desired interval from 0 to 1.

$$o_\mu = \iota(o_L) = \frac{1}{1+e^{-o_L}} \equiv \hat{\mu}_{BRANN} \tag{2.6}$$

Since the ϕ parameter of BRANN is modeled as a constant, the ϕ-sub-network in BRANN can be formalized as follows:

$$o_\phi = \zeta(r) = e^{w_\phi r + b_\phi} \equiv \hat{\phi}_{BRANN} \tag{2.7}$$

with $r \in \mathbb{R}$ as the constant input, usually $r = 1$, $w_\phi \in \mathbb{R}$ and $b_\phi \in \mathbb{R}$ are weight and bias for r. In a final step o_μ and o_ϕ are merged to get predictions of μ and ϕ.

G-BRANN

For G-BRANN we use the same procedure as for BRANN, but increase the flexibility in the ϕ-sub-network by allowing multiple input variables, that can differ from the input variables of the μ-sub-network. Hence, we allow our model to incorporate non-linearity, which can be different in the μ- and the ϕ-sub-network. Since the μ-sub-network is the same as in Equation (2.6), only the ϕ-sub-network changes to a network with Z layers where $z = 1, ..., Z$.

$$o_z = \psi(W_z o_{z-1} + b_z) \tag{2.8}$$

$$o_\phi = \zeta(o_Z) \equiv \hat{\phi}_{G-BRANN} \tag{2.9}$$

Trainable activation functions

Outliers can impact the optimization tremendously and cause problems in the whole estimation process. This is especially true for the ϕ-part and out-of-sample predictions. The usual link function for the ϕ-part of the generalized beta regression is the exponential function to ensure positivity of $\hat{\phi}$, but this link function can be too steep or too flat. Because we do not know the best suiting link function, we choose a data-driven approach and give G-BRANN the flexibility to learn the last activation function for ϕ from the data. We introduce three trainable activation functions in the spirit of the Parametric Rectified Linear Unit by He et al. (2015). Hence, the last activation function for ϕ, labeled as $\zeta(.)$, differs for G-BRANN. The first one, the trainable exponential function (t-exp), can be modeled in terms of the exponential function, with the addition that the steepness of the curve is defined by the parameter a. This trainable parameter makes it possible for the G-BRANN to determine how steep the function should be in a data-driven fashion. If a is equal to Euler's number, the function results in the original link function. The trainable exponential function with a as the trainable parameter can be calculated as follows:

$$\zeta(o_Z) = e^{o_Z \cdot log(a)} = a^{o_Z} \tag{2.10}$$

The second function, trainable softplus (t-soft), is based on the softplus function with a steepness parameter q. Analogous to the first function, the network can learn how steep the activation function should be. The trainable softplus function with q as a trainable steepness parameter can be represented as:

$$\zeta(o_Z) = \frac{log(1 + e^{q \cdot o_Z})}{q} \tag{2.11}$$

The last activation function is called the trainable sigmoid function (t-sig), which is an extension of the adjustable generalized sigmoid, as described in Apicella et al. (2021), as we introduce an additional shifting parameter c. The sigmoid is bounded on the open interval from zero to one. Because the ϕ parameter of a beta distribution can be any positive number a few adaptions have to be made. A multiplicative constant h can be used to stretch the sigmoid function to the open interval from zero to h, which can be any positive number. A common problem of the sigmoid function is saturation. To resolve that, we introduce two additional trainable parameters. The parameter s is the steepness parameter of the sigmoid function. For decreasing negative s the sigmoid function tends toward a step function. The last parameter c is the shifting parameter. This can be helpful if the output of the function tends towards the lower bound. Since zero is the lower bound of the sigmoid function and the ϕ of a beta regression, the log-likelihood can explode for very low ϕ.

The trainable logistic function with h, s, and c as height, scale, and shift parameters are defined as:

$$\zeta(o_Z) = \frac{h}{1 + e^{s \cdot o_Z}} + c \tag{2.12}$$

Graphical illustrations of the activation functions can be found in Figures 2.B.1, 2.B.2 and 2.B.3 in Appendix 2.B. Table 2.2 summarizes the proposed activation functions.

Table 2.2: Trainable activation functions

Function	Formula	Parameters
Trainable exponential function	$\zeta(o_Z) = e^{o_Z \cdot log(a)} = a^{o_Z}$	a: Steepness parameter
Trainable softplus function	$\zeta(o_Z) = \frac{log(1+e^{q \cdot o_Z})}{q}$	q: Steepness parameter
Trainable logistic function	$\zeta(o_Z) = \frac{h}{1+e^{s \cdot o_Z}} + c$	h: Height parameter s: Scale parameter c: Shift parameter

Notes: This table illustrates the different definitions of the novel trainable activation functions. Each of the parameters is trained during the model fit and can be estimated data-driven.

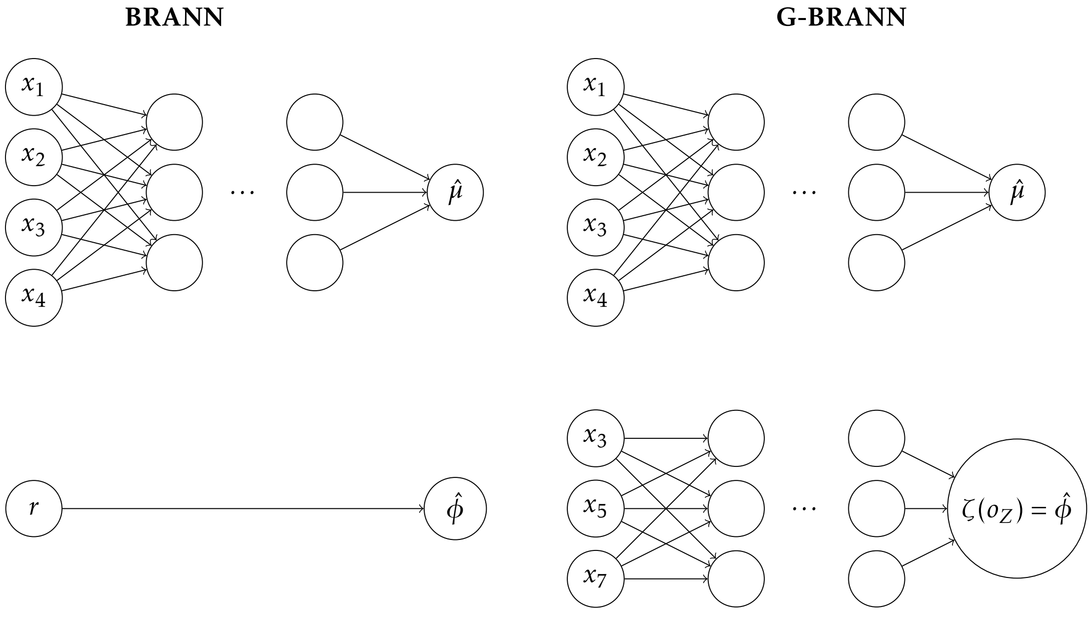


Figure 2.2: Graphical illustration BRANN and G-BRANN

Notes: This figure illustrates a stylized network structure of BRANN and G-BRANN. The advantage of these structures is that we can allow different input variables for each of the distributional parameters of the beta distribution.

A simple graphical illustration of BRANN and G-BRANN with an arbitrary number of hidden layers is provided in Figure 2.2. In this example we take the variables $\{x_1, x_2, x_3, x_4\}$ as input variables for modeling the μ-sub-network. Each input is weighted and non-linearly transformed in the hidden layers. The ϕ-sub-network gets the constant input r, which usually has the value $r = 1$ as the sub-network models ϕ by weighting the constant without a non-linear transformation. This results in a constant ϕ for every observation. G-BRANN can consist of the same μ-part, but this is not mandatory. The ϕ-part shares the input variable x_3 with the μ-sub-network, but has additional variables x_5 and x_7. These input variables are also non-linearly transformed in the hidden layers. The objective function for both network types is the same as for the beta regressions, defined in Equation (2.4). For stability reasons, we minimize the mean of the negative log-likelihood instead of the sum, so the gradients do not tend to explode. For every market-based LGD i, we model individual values of $\hat{\mu}_i$ and $\hat{\phi}_i$. For BRANN and the linear beta regression, $\hat{\phi}_i$ is constant for all observations.

Accumulated Local Effects plots

As BRANN and G-BRANN rely on neural networks, they are black-box by nature. However, the body of literature focusing on explanation methods has grown fast. Bastos and Matos (2022) provide a comprehensive overview of recent XAI techniques for credit risk. They show that financial institutions can use these techniques to (probably) comply with regulatory concerns of recovery rate predictions. Similar conclusions are drawn by Kellner et al. (2022). We use Accumulated Local Effect (ALE) Plots by Apley and Zhu (2020) to analyze BRANN and G-BRANN. This method is a common choice for visualizing feature effects in credit risk. One example is Bellotti et al. (2021), who use ALE plots on workout LGDs or Barbaglia et al. (2023) also using ALE Plots to analyze the drivers of probability of defaults of European mortgage. Multiple XAI methods, including ALE plots as well as Shapley values, are compared by Bastos and Matos (2022).

To compute the ALE plots, we first divide the range Z of one specific predictor $X_j \in \mathbb{R}^{N\times 1}$, where $j = 1,\dots,p$, into a grid with $k \in [0,1,\dots,K]$, where K is the number of total bins. Following Apley and Zhu (2020), Z_k is chosen as the $\frac{k}{K}$ quantile of the empirical distribution of X_j, where Z_0 is the minimum and Z_K is the maximum. Let S_k define a set of observations that lies between the boundaries Z_{k-1} and Z_k. Furthermore, n_k denotes the number of observations in S_k, and $k(X_j)$ is an index that indicates in which bin a given value of X_j falls. The (uncentered) ALE can then be written as:

$$g_{ALE}(X_j) = \sum_{k=1}^{k(X_j)} {n_k}^{-1} \sum_{i\in S_k} \Big[f(Z_k, X_{\backslash j,i}) - f(Z_{k-1}, X_{\backslash j,i}) \Big]. \tag{2.13}$$

$X_{\backslash j} \in \mathbb{R}^{N\times p-1}$ defines the set of variables without the variable j and $f(.)$ denotes the neural network's predictor before the final transformation. For each observation i we obtain a prediction assuming X_j to be the upper and lower limit of the interval, i.e., Z_{k-1} and Z_k, and calculate its difference. These differences are summed over all observations in the bin and weighted by the number of observations in that bin, n_k, to obtain the (uncentered) local effect of X_j. Finally, we accumulate these weighted summed differences up to a given value of X_j using the outer sum. This result is centered such that the mean effect of X_j is zero:

$$\Theta_{ALE}(X_j) = g_{ALE}(X_j) - N^{-1} \sum_{i=1}^{N} g_{ALE}(X_{j,i}) \tag{2.14}$$

The ALE plots have many advantages. Among other things, they are fast to compute and unbiased. Therefore, they can be used even if features are correlated in contrast to many other XAI techniques, such as partial dependence plots, see Apley and Zhu (2020). ALE plots are centered so that the mean effect of the features is zero. Therefore, the y-axis of the ALE can be interpreted as the main effect of the independent variable at a certain point in comparison to the average predicted value. Furthermore ALE plots provide a R^2-like measure, which describes up to which degree the prediction can be explained by main order, second order, etc. effects.

The proposed $R^2_{ALE,m}$ by Apley and Zhu (2020) can be formalized as follows:

$$R^2_{ALE,m} = \frac{var\{\sum_{J\subsetneq\{1,\ldots,d\},|J|\leq m}\Theta_{ALE}(X_J)\}}{var\{f(X)\}} \tag{2.15}$$

where m describes up to which order of effects the R^2_{ALE} is calculated. Therefore, it holds that $R^2_{ALE,d} = 1$. Nagl (2023) extended this approach by introducing $R^2_{ALE,linear}$, which measures how much linearity the prediction of the model contains. The $R^2_{ALE,linear}$ is defined as:

$$R^2_{ALE,linear} = \frac{var\{\sum_{j=1}^{p}\Theta^{linear}_{ALE}(X_j)\}}{var\{f(X)\}} \tag{2.16}$$

where $\Theta^{linear}_{ALE}(X_j)$ can be calculated by fitting a linear regression on $\Theta_{ALE}(X_j)$, which are the first order effects. Therefore, $1 - R^2_{ALE,linear}$ quantifies the degree of non-linearity in the prediction. All ALE plots are generated with a grid size of $K = 10$ and we calculate the ALE Plots for o_L (Equation(2.5)) and o_Z (Equation (2.8)), i.e., before the final transformation in the output layer.

2.5 Results

2.5.1 Feature selection & model estimation

Feature selection

The selection of important drivers for market-based LGDs is not trivial. Therefore, we use the selection of Gambetti et al. (2019) as a starting point of our analysis. However, as our study additionally uses data after 2015 (about 7 years more), we follow an iterative process to select the most relevant to our sample. We divide our data set randomly into an in-sample (80%) and an out-of-sample (20%) groups. For the feature selection process, we further divide the in-sample set into a training (70%) and testing set (30%). During this process, various feature sets are calibrated on this training set and predictions are generated for the testing set. The value of the loss function, i.e. the negative log-likelihood, on the testing set serves as metric for the feature selection process. Therefore, we select our feature to be suitable to predict out-of-sample data. Alternatively, we can use the same approach as Gambetti et al. (2019) and follow a step-GAIC approach. Then, however, we would select our feature only on training data, i.e. in-sample. As we want a model which can also predict out-of-sample/time data, we opt for selecting features by out-of-sample losses. Using AIC instead of the negative log-likelihood, does not change our results.

Table 2.3: Selected variables for the sub-networks

Bond characteristics		
Variable	μ-sub-network	ϕ-sub-network
Coupon rate	✓	✓
Maturity	✓	✓
Industry sector	✓	✓
Seniority	✓	✓
Default type	✓	
Macroeconomic variables		
Variable	μ-sub-network	ϕ-sub-network
Recession indicator	✓	✓
Industry production	✓	✓
S&P 500 Returns	✓	✓
Delinquency rates	✓	
Default rate	✓	✓
Uncertainty measures		
Variable	μ-sub-network	ϕ-sub-network
Financial uncertainty	✓	✓
CPI uncertainty	✓	✓
News-based EPU	✓	
Uncertainty relative to federal/state/local expenditures	✓	

Notes: The NBER-based recession indicators for the United States are retrieved from the Federal Reserve Bank of St. Louis (FRED) website as well as the industrial production and S&P 500 and the delinquency rate on All Loans. Following Gambetti et al. (2019), we gather the American default rates from Moody's database and control for withdrawal effects. The uncertainty measures are retrieved from the author's website. These include financial uncertainty (Jurado et al. (2015) and Ludvigson et al. (2021)), the news-based EPU from Baker et al. (2016). Furthermore, we use survey-based proxies of uncertainty by including the inflation uncertainty measure for the United States (CPI uncertainty) and a proxy of uncertainty relative to both federal and state/local purchases.

Since the μ-part is well researched in terms of drivers, see Gambetti et al. (2019), we choose the same variables for the μ-part, which are best performing in Gambetti et al. (2019).[13] For the ϕ-part we apply the forward selection algorithm using the generalized linear beta regression, assuming the same selection for the μ-part from the previous step, following Gambetti et al. (2019).[14] Please note that the final set of features is robust to different splitting points in the

[13] In contrast to Gambetti et al. (2019) we dropped the Real Gross Domestic Product(GDP) as it worsens the performance considerably. This may partly be traced back to the recent crisis, where we observed large variations in GDP but almost no variation in the resulting LGD in these quarters.

[14] Alternatively, one could also use BRANN and G-BRANN for feature selection, but this would include a hyperparameter search in each step. As the aim of this paper is not to find the ultimate selection of drivers of market-based LGDs, we think that our approach is sufficient. Overall our selection recovers recent findings by Sopitpongstorn et al. (2021), Nazemi et al. (2021) and Bastos and Matos (2022).

training (70%) and testing (30%) set. Table 2.3 shows the final variables for the μ- and ϕ-parts.

Model estimation

To find the optimal parameters for BRANN and G-BRANN, we conduct a random 5-fold cross-validation of the training sample. In summary, we draw 500 different configurations of hyperparameter values. Similar to Kellner et al. (2022) we adopt the multiple approach to find the number of neurons in each subpart of our network. The baseline for the multiple approach is (32, 16), i.e., we use maximum two hidden layers. In this approach we sample a multiplier for the baseline network structure instead of directly sampling the number of neurons in each hidden layer.[15] The descending number of neurons in each hidden layer is inspired by Gu et al. (2020). Furthermore, we use Stochastic Gradient Decent (SGD) as an optimizer and ReLU as an activation function in all hidden layers, which is in line with the literature, see, e.g., Gunnarsson et al. (2021) or Nagl et al. (2022). To increase the robustness of our results, we fit every constellation 10 times and use the average of these repetitions in the hyperparameter search. This eliminates the impact of the random weight initialization in the first step of the training phase.[16] Hyperparameters are the learning rate, the multiple, the dropout rate, the number of hidden layers, and our novel trainable activation functions for the ϕ-part in G-BRANN. In addition to that we include a MaxNorm kernel constraint of 3.0 as recommended by Srivastava et al. (2014) for dropout in neural networks. The search space and the final values are reported in Table 2.4.

Table 2.4: Setup and final values of the hyperparameter search

Parameter	Distribution	BRANN	G-BRANN	
		μ-part	μ-part	ϕ-part
Learning rate	$U^c \sim [0.001, 0.1]$	0.0396	0.0091	
Dropout rate	$U^c \sim [0.05, 0.50]$	0.4385	0.3435	0.3144
Hidden layer	$U^d \sim [1, 2]$	2	2	2
Multiple	$U^d \sim [1, 8]$	1	3	8
Activation function	t-exp, t-sig, t-soft	-	-	t-exp ($\alpha = 3.37$)

Notes: The table shows different values for the hyperparameter search. U^c labels the continuous uniform distribution, whereas U^d labels the discrete uniform distribution. We observe that G-BRANN requires a wider network structure for the μ-part and a wide and deep structure for the ϕ-part.

Interestingly, the estimated coefficient of the t-exp activation function differs to Euler's number, indicating that G-BRANN selects a different shape of this activation function to be optimal.[17]

[15] For example, if we sample a multiplier of 4 in a two hidden layer network, we have (128, 64).

[16] We find that 10 repetitions are enough in our setup. The differences in the means of 10 independent repetitions are negligible, and, thus, we find our results robust and reproducible.

[17] As robustness, we also conduct a hyperparameter search for G-BRANN where we use only the standard activation function in the output layer of ϕ. Overall, the trainable activation function outperforms the standard functions consistently.

Table 2.5: Evaluation Metrics

(**a**) In Sample

			Generalized linear	
	Beta Regression	BRANN	beta Regression	G-BRANN
Σ Log-likelihood	747.470	1211.769	968.669	**2085.335**
$\emptyset$ Log-likelihood	0.404	0.654	0.523	**1.126**
MSE	0.038	**0.022**	0.038	**0.022**
Pseudo-R^2	0.464	**0.688**	0.447	0.636

(**b**) Out of Sample

			Generalized linear	
	Beta Regression	BRANN	beta Regression	G-BRANN
Σ Log-likelihood	165.536	232.737	227.342	**300.546**
$\emptyset$ Log-likelihood	0.358	0.503	0.491	**0.649**
MSE	0.041	**0.031**	0.041	0.030
Pseudo-R^2	0.403	**0.549**	0.398	0.538

Notes: This table shows the average performance metric of BRANN and G-BRANN over 100 repetitions and their linear counterparts. The first row shows the sum of log-likelihood to be comparable to the literature. The second row shows the average log-likelihood, whereas the third row shows the mean squared error. The last row displays the Pseudo-R^2 following Ferrari and Cribari-Neto (2004). We observe that the neural network related methods consistently outperform the linear models in every performance metric. To remain comparability to Gambetti et al. (2019) we report the sum of the log-likelihood. Bold values indicate the best, whereas underlined values the second best performance.

Our main metric for the performance comparison is the log-likelihood, as it measures the performance concerning the distributional fit. However, we additionally include two common metrics from the literature, namely the MSE and the Pseudo-R^2 to assess how well the *mean* estimates perform.[18]

Table 2.5 illustrates that BRANN and G-BRANN outperform their linear counterparts by a large margin in terms of sum and mean log-likelihood. This also holds for MSE, and Pseudo-R^2 in-sample as well as out-of-sample. Bold values indicate the best value, underlined values indicate the second best. Overall, the neural networks are first or second-best choice for every metric. To remain comparable with Gambetti et al. (2019), we report the sum of the log-likelihood. Hence, the values of the out-of-sample data set are smaller due to the smaller sample size. Looking at the mean log-likelihood, we observe that the values are, as expected, slightly smaller but comparable to the in-sample values. Overall, we see substantial improvements in the

[18] The MSE measures the quadratic difference between true and predicted mean LGDs. The Pseudo-R^2 is derived by Ferrari and Cribari-Neto (2004) and quantifies the squared correlation between the linear predictors of the model and the true realization.

log-likelihood by the BRANN and G-BRANN models.

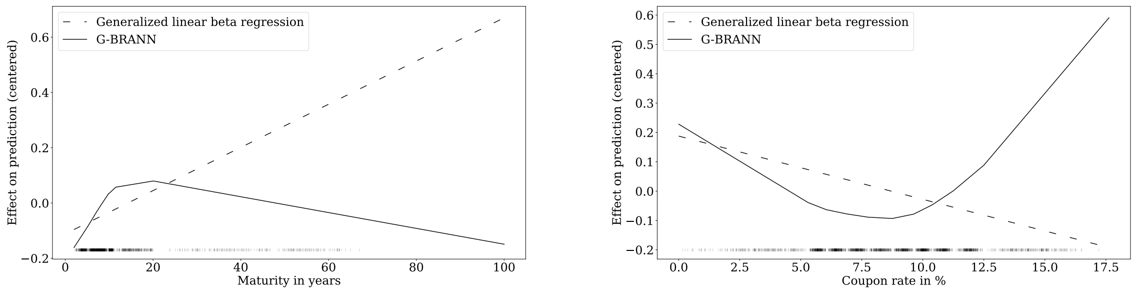
One might argue that the comparison of the "raw" likelihood is not fair as we do not control for the larger number of parameters in the neural networks and we should rely on metrics like the Akaike Information Criterion (AIC) instead. This could be the case for in-sample values due to overfitting resulting in higher likelihood values. However, if we interpret this along with the (mean) likelihood values of the out-of-sample data, we do not see evidence for overfitting as we implemented various regularization techniques and rely on cross-validation. Furthermore, the number of parameters in neural networks does not necessarily coincide with the complexity. Recent literature shows that overparameterized neural networks even generalize better than those with a lower number of parameters, see, e.g. Belkin et al. (2019) or Yang et al. (2020).
To counteract concerns that our results are not robust to an out-of-time exercise and in comparison to other well-known machine learning models, we conduct a robustness check in Appendix 2.C. The search space of possible hyperparameters including the final results can be found in Table 2.C.1. As displayed in the subtables of Table 2.C.2 we observe that the superiority of (G-) BRANN holds also for future predictions, and regarding the mean estimate, we observe similar performance.

2.5.2 Drivers of μ and ϕ

Bond related drivers of μ

The following figures unveil the relationship between the selected variables and the predicted mean of the LGD distribution. To allow a better comparison with the traditional approach, i.e. linear models, we also add their relationship. Overall, the ALE plots in the μ-part from G-BRANN and BRANN point in the same direction. Therefore, only the ALE plots of G-BRANN are presented in the following.[19] As the number of observations varies across the value range of the drivers, areas with a low number of observations should be interpreted with caution. Nonetheless, we have enhanced the robustness of our interpretations by refitting the models and recalculating the ALE Plots 100 times.

[19] The plots for BRANN as well as the plots of the control variables, such as industry sector, seniority, and default type are available upon request.



Maturity

Coupon rate

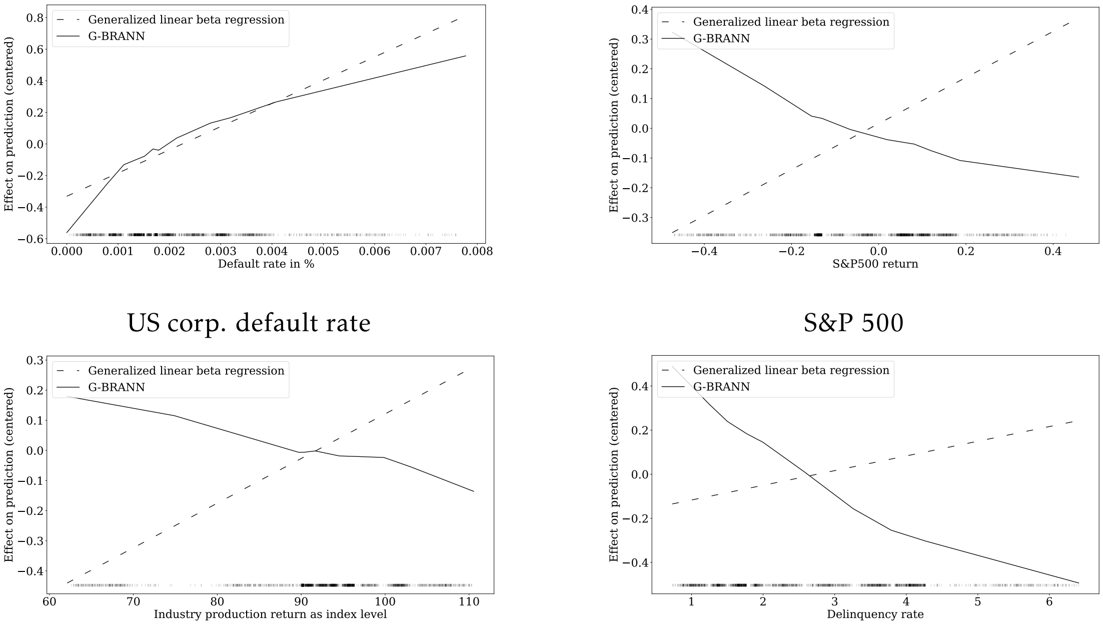
Figure 2.3: ALE plots of bond characteristics | μ

Notes: The figures show the ALE plots for G-BRANN in solid black and the generalized linear beta regression in dashed black. We initialize the G-BRANN 100 times and calculate the final median ALE plots. We also include a rug plot at the bottom to visualize the distribution of the underlying data.

Figure 2.3 shows that the slopes of the linear model (dashed line) coincide with the (linear) evidence from the literature, i.e., LGD increases with maturity and decreases with coupon rate, see, e.g., Gambetti et al. (2019). However, the ALE plots of G-BRANN reveal a more nuanced picture. We observe that a higher maturity results in higher mean LGDs up to a certain point, but this increasing effect vanishes for bonds with a maturity greater than roughly 20 years. Generally, the positive relationship could be explained by sell-side pressure originating from institutional investors who usually hold bonds with longer maturity, see Jankowitsch et al. (2014). This effect decreases for maturities greater than 20 years and even gets negative. But this negative effect could be due to the small number of bonds with very long maturities. For BRANN the effect of the maturity is almost constant after the 10 years mark. For the coupon rate, we observe a u-shaped relationship, as the LGD decreases for coupon rates up to 9%, but increases afterwards. A negative relationship is plausible as bonds with a higher coupon rate could be of higher value given there is a reasonable probability that all cash flows can be collected during the resolution of the bond, see Jankowitsch et al. (2014). However, a higher coupon rate also indicates higher risk, and, thus, for riskier bonds, the market expects higher losses as the probability that all cash flows can be recovered may be lower.

Macroeconomic drivers of μ

Figure 2.4 shows that the default rate has the largest impact of the macroeconomic drivers on market-based LGDs, which appears logarithmic. For S&P 500 returns the linear model finds a (counterintuitive) positive relationship, whereas G-BRANN finds a (intuitive) negative relationship. Similarly, higher industry production is associated with higher losses in the linear model, but has a negative effect in the G-BRANN model. The last macroeconomic variable delinquency rate has an intuitive positive sign in the linear model, but a counterintuitive relation in the G-BRANN model. This is similar to Gambetti et al. (2019), where the delinquency rate has no significant impact in their best model. The counterintuitive sign might be due to correlations in the macroeconomic variables. The problem of counterintuitive signs when incorporating many of them is well-known in the credit risk literature. Figlewski et al. (2012) find that many macroeconomic variables change their signs and have even statistically significant counterintuitive signs if a large selection of them is included.



US corp. default rate

S&P 500

Industry production

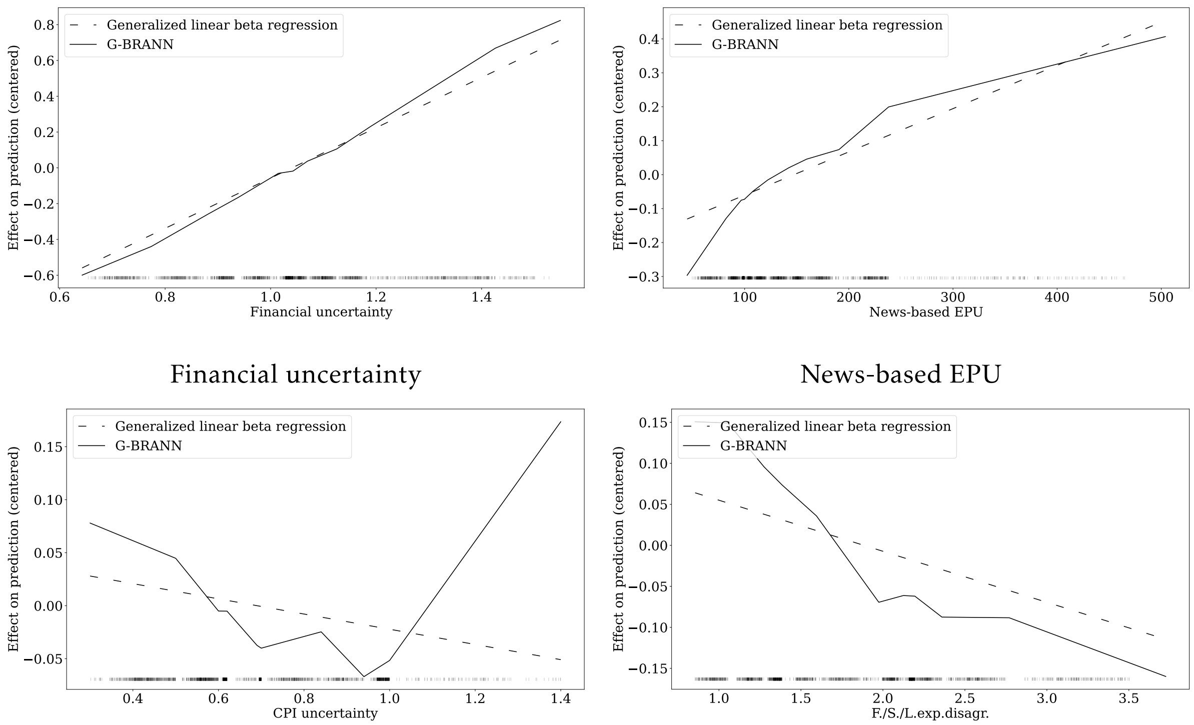
Delinquency rate

Figure 2.4: ALE plots of macroeconomic variables | μ

Notes: The figures show the ALE plots for G-BRANN in solid black and the generalized linear beta regression in dashed black. We initialize the G-BRANN 100 times and calculate the final median ALE plots. We also include a rug plot at the bottom to visualize the distribution of the underlying data.

Uncertainty related drivers of μ

Figure 2.5 shows the ALE plots of uncertainty related variables. We find a substantial positive impact of financial uncertainty, which is similar to Gambetti et al. (2019). The effect is linear in G-BRANN and nearly identical with its linear counterpart. Financial uncertainty has the largest effect of the uncertainty related drivers on market-based LGDs. For the text-related news-based EPU index we find a positive relationship, which is close to the linear model from an index level of 100 onwards. CPI uncertainty shows a U-shaped relation indicating that market-based LGDs decrease for low levels of uncertainty but increase sharply after a certain point. FSL uncertainty exhibits a negative effect, which is in line with the linear model, but in G-BRANN the effect is more extreme up to a value of 2. Then its slope is similar to the linear counterpart. Overall, we find larger non-linearities in uncertainty-related variables than in macroeconomic variables.



Financial uncertainty

News-based EPU

CPI uncertainty

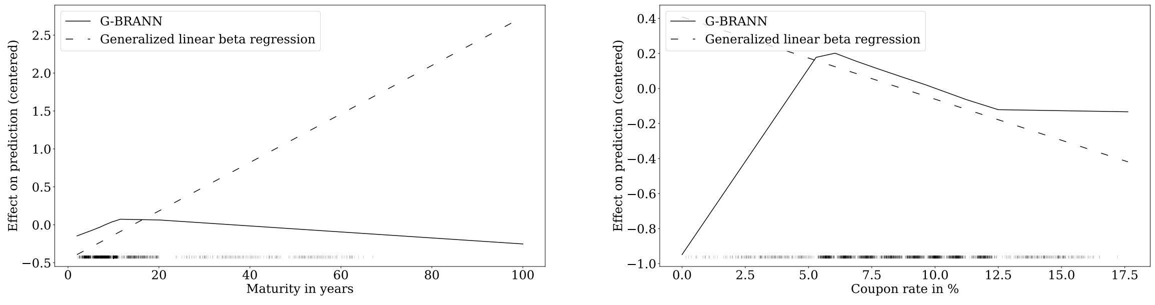
FSL uncertainty

Figure 2.5: ALE plots of uncertainty related variables | μ

Notes: The figures show the ALE plots for G-BRANN in solid black and the generalized linear beta regression in dashed black. We initialize the G-BRANN 100 times and calculate the final median ALE plots. We also include a rug plot at the bottom to visualize the distribution of the underlying data.

Bond related drivers of ϕ

Following the definition of precision ϕ, the estimated sign is inversely connected to the variance of the LGDs. As a consequence of this an estimated negative effect increases the variance of the resulting LGD distribution.



Maturity

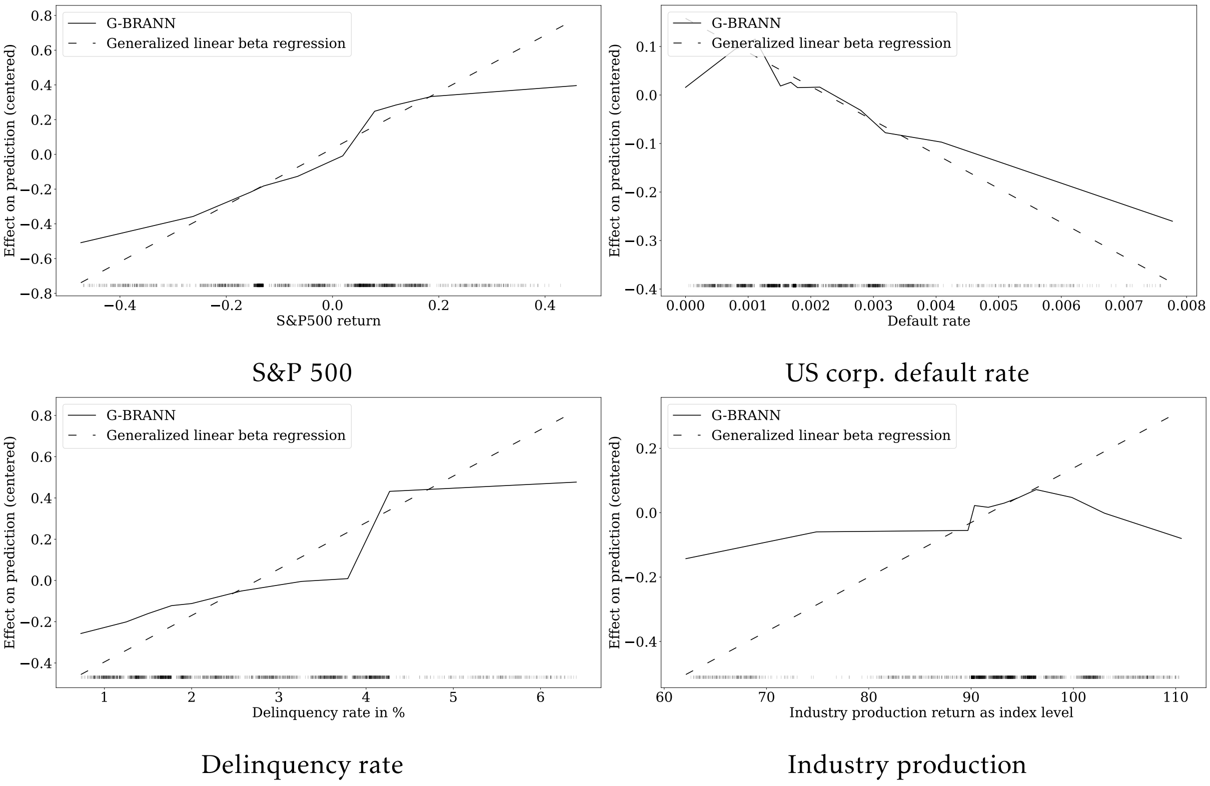
Coupon rate

Figure 2.6: ALE plots of bond characteristics | ϕ

Notes: The figures show the ALE plots for G-BRANN in solid black and the generalized linear beta regression in dashed black. We initialize the G-BRANN 100 times and calculate the final median ALE plots. We also include a rug plot at the bottom to visualize the distribution of the underlying data.

Figure 2.6 shows the impact of the two bond characteristics. We find a small negative effect of maturity on the precision of market-based LGDs. This implies that bonds with longer maturities are associated with less variance in the LGD estimate. G-BRANN recovers an increasing effect up to a certain coupon rate. Therefore for low coupon rates we have a negative effect on the precision, which becomes less negative as the coupon rate approaches 5%. For coupon rates between 5% and 10% we have a decreasing positive effect on the precision, which becomes slightly negative for higher coupon rates. Due to the inverse relationship between precision and variance we expect higher variances of the LGDs for low and high coupon rates.

Macroeconomic drivers of ϕ



S&P 500

US corp. default rate

Delinquency rate

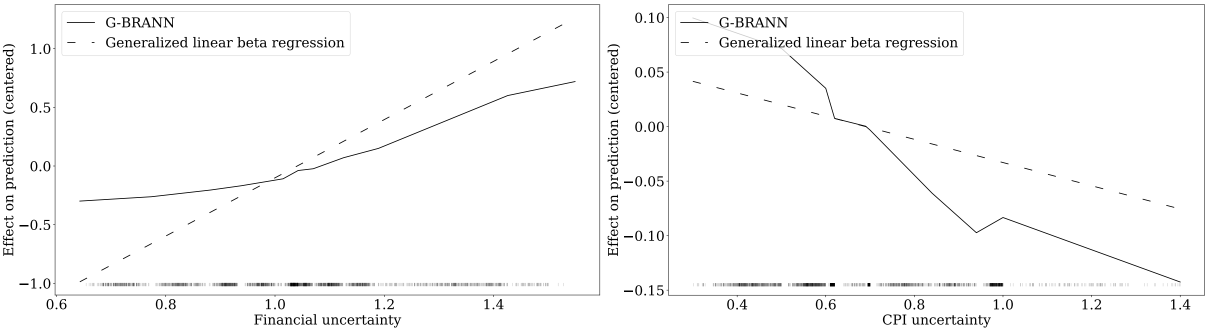
Industry production

Figure 2.7: ALE plots of macroeconomic variables | ϕ

Notes: The figures show the ALE plots for G-BRANN in solid black and the generalized linear beta regression in dashed black. We initialize the G-BRANN 100 times and calculate the final median ALE plots. We also include a rug plot at the bottom to visualize the distribution of the underlying data.

Turning to the influence of the macroeconomy on the precision of market-based LGDs, Figure 2.7 illustrates their impact. We find a positive relationship between S&P 500 returns and precision, implying that the variance decreases for higher returns. This is somewhat contrary to Gambetti et al. (2019), but they used the level of the S&P 500 and not the (stationary) returns. This positive relationship for very high returns can be partly explained as we included the recent Covid-19 pandemic in our sample, where we observe large positive returns, although the LGD realization stagnated. The positive effect is less pronounced than its linear counterpart. For default rates we have an increasing effect on the prediction for very small default rates, turning negative afterwards, which is consistent with the linear model. For delinquency rates, we find a similar picture as for the S&P 500 returns. The impact on the prediction has the same direction as the linear model but is more conservative. The industry production has a small, constant negative effect on average.

Uncertainty related drivers of ϕ



Financial uncertainty

CPI uncertainty

Figure 2.8: ALE plots of uncertainty related variables | ϕ

Notes: The figures show the ALE plots for G-BRANN in solid black and the generalized linear beta regression in dashed black. We initialize the G-BRANN 100 times and calculate the final median ALE plots. We also include a rug plot at the bottom to visualize the distribution of the underlying data.

In our selection, only two uncertainty-related measures were selected for the final model. We find that financial uncertainty has a almost linear effect on the precision, so that high financial uncertainty corresponds to lower variance of the LGD estimation. For this variable the effect is less strong than the effect modeled by the generalized linear beta regression. CPI uncertainty on the other hand shows a negative trend for increasing uncertainty, but the overall effect is comparable small.

Non-linearity in the estimation of μ and ϕ

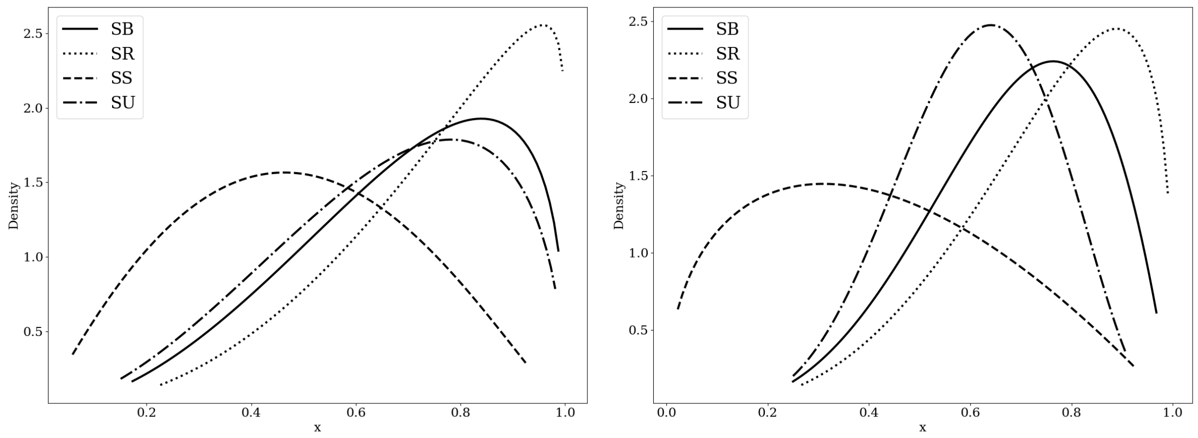
ALE plots are a powerful tool to visualize the modeled effects of features on the prediction. Due to the connection of ALE plots to a functional-ANOVA-like-decomposition ALE plots are capable to quantify the goodness of fit to the prediction due to an arbitrary order of effects according to Apley and Zhu (2020). We calculate the $R^2_{ALE,1}$ for the parameter of the modeled distribution. Therefore, we can measure how well the prediction can be approximated by the first order (main) effects, which are visualized in Figures 2.3 to 2.8. For μ the $R^2_{ALE,1,\mu}$ is 0.9017. This means, that 90.17% of the prediction is due to (non-)linear main effects and only the remaining 9.83%

are a result of (non)-linear higher order effects such as interactions. This picture changes for ϕ. Here the $R^2_{ALE,1,\phi}$ is 0.3531. Therefore, the most part of the prediction is due to (non-)linear higher order effects. Using the $R^2_{ALE,linear}$ derived by Nagl (2023), we can divide the $R^2_{ALE,1}$ further. As $R^2_{ALE,linear}$ measures the how well the prediction can be explained by linear first order effects, the difference between $R^2_{ALE,1}$ and $R^2_{ALE,linear}$ is the improved approximation by non-linearity in the first order effects. For G-BRANN $R^2_{ALE,linear,\mu}$ is 0.8590, which indicates that the improvement in the first order effects by non-linearity is only about 4%. For the precision parameter this becomes more pronounced. $R^2_{ALE,linear,\phi}$ is only 0.1963, which means, that 80.37% of the ϕ predictions is due to non-linearity and higher order effects. More specific the increase due to non-linearity in the first order effects is more than 15%. Therefore, the non-linearity has a tremendous effect for the estimation of ϕ even in the first order effects.

2.5.3 Scenario analysis

The remaining part of this section focuses on the implications of our findings for risk management. As stated in Kellner et al. (2022) only considering the mean or median does not allow to differentiate between risk profiles in a holistic way. Therefore, the whole distribution should be taken into account to derive risk profiles across possible realizations of the LGD.

Assume a bank aims at investigating the implications of favorable and unfavorable scenarios in their credit risk assessment. These scenarios can be easily derived using low and high quantiles of individual LGD distributions, predicted by G-BRANN. To examine this, we compare the trained beta regression and G-BRANN as described in Section 2.5.1 and predict the μ and ϕ for every sample in the training data. Figure 2.9 shows the results for the different types of seniority. To obtain a representative distribution of each of them, we take the the median μ and ϕ for every seniority. The left hand side of Figure 2.9 shows the estimated distributions by the beta regression and on the right side the distributions calculated by G-BRANN are displayed. In the beta regression models all samples have the same value for ϕ, whereas G-BRANN allows individual values of ϕ. Please recall that the overall fit of G-BRANN in terms of likelihood is considerably higher compared to the beta regression. This holds also for every individual category, such as seniority. Therefore, we are confident that the estimated distributions by G-BRANN are superior as well.



Beta Regression

G-BRANN

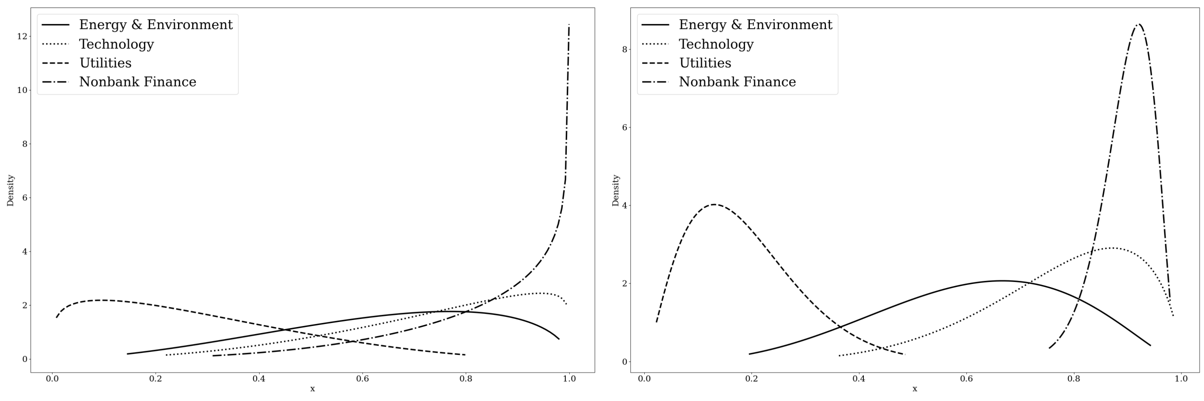
Figure 2.9: Beta distributions per seniority

Notes: The figures show the representative beta distribution modeled by the beta regression and G-BRANN divided into the seniorities. SB, SR, SS and SU refer to Senior Subordinated, Subordinated, Senior Secured and Senior Unsecured.

From a risk manager's perspective a more differentiated picture across categories provides a valuable information to derive individual risk profiles. Therefore, the less these distributions overlap between categories, the more refined can the derived risk profiles be. Overall, using the beta regression the distributions overlap more than using the G-BRANN. Therefore, G-BRANN allows a more refined picture of the different distribution. Again, the fit in terms of likelihood is superior for every category and, thus, the less overlapping distributions suit the data more. To quantify this effect, we calculate the area where the distributions overlap. Since the integral of these distributions is always one, the overlapping area is naturally bounded from zero to one, where one means that one distribution envelops the other distribution. We calculate the overlapping area for every pair of seniority levels. On average the overlapping area of the beta regression is 0.7443 in contrast to 0.6402 for G-BRANN.

We redo the same analysis for the industry types. Typically, the LGDs vary across different industries due to differences in collateralized assets or guarantees. Therefore, a risk manager appreciates a models that allows for a clear distinction between LGDs in different industries. Similar to the seniority, the fit of G-BRANN is superior in every industry compared to the beta regression, which is currently industry standard.

Figure 2.10 shows the estimated distribution for the most common industries in our sample. Overall, we observe a similar picture. G-BRANN produces much more differentiated distributions than the beta regression.



Beta Regression G-BRANN

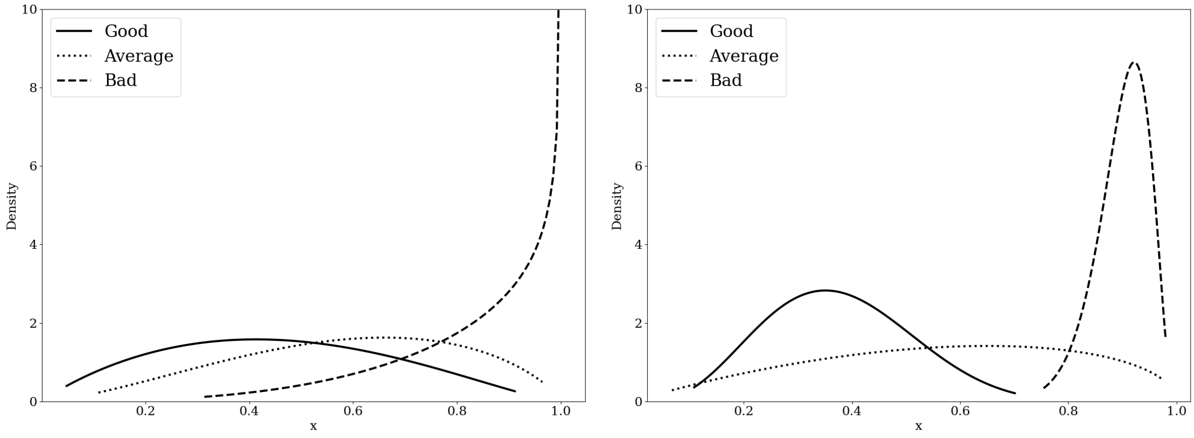
Figure 2.10: Beta distributions per industry type

Notes: The figures show the representative beta distribution modeled by the beta regression and G-BRANN divided into the industry types. Here just a selection of industry types are shown.

Calculating the mean overlapping area the beta regression results in 0.6919 and G-BRANN 0.4444. The individual overlappings are even more different. G-BRANN predicts a distribution for the Utilities sector that does not have any overlapping with the Nonbank Finance sector. On the contrary, the beta regression shows a comparable large overlap. In our data, LGDs from the Utilities sector have the lowest average LGD, whereas the Nonbank Finance sector has the highest according to Table 2.A.3. Again, this shows that G-BRANN reflects the empirical features of our data much better.

Lastly, risk managers do not only want to differentiate between industry types or seniorities, but also between different macroeconomic states. Therefore, we provide a scenario analysis which focuses on the economic surrounding. We choose three quarters with different average realized LGDs. As “good” scenario we rely on the macroeconomic state in Q1 2004 with an average LGD of 0.43, which is comparatively low. The “average” scenario is Q4 2005 with a mean LGD of 0.62 which is very close to the average of our whole dataset. The “bad” case is Q3 2008, which is a quarter of the Global Financial Crisis that is reflected by the very large mean LGD of 0.90. The “good”, “average” and “bad” states are also reflected by the macroeconomic variables, such as the S&P 500 return or the US corp. default rate.

Figure 2.11 illustrates a clear separation between the good and the bad scenario for G-BRANN, whereas the distributions modeled by the beta regression overlap by 0.3502 compared to 0.0095 for G-BRANN. Again, the fit in every macroeconomic state of G-BRANN is considerably better than by the beta regression and, thus, this clear separation is more aligned with the data. Furthermore, the clear separation between good and bad economic states is also economically more plausible.



Beta Regression

G-BRANN

Figure 2.11: Beta distributions per macroeconomic states

Notes: The figures show the representative beta distribution modeled by the beta regression and G-BRANN divided into different macroeconomic sates.

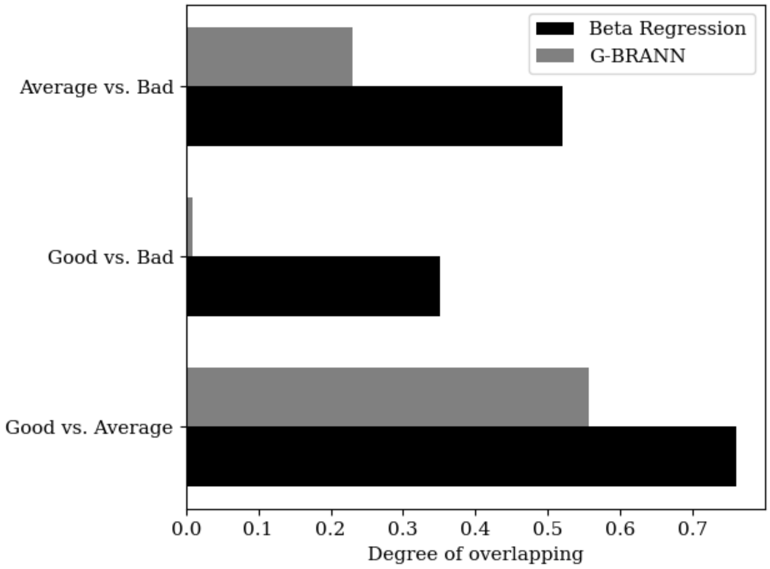


Figure 2.12: Degree of overlapping across the macroeconomic states

Notes: The figure shows the degree of overlapping of the representative beta distribution modeled by the beta regression and G-BRANN divided into macroeconomic states.

Figure 2.12 visualizes the overlapping area of Figure 2.11 to allow for a easy comparison. We observe that the difference between “average” and “bad”, the G-BRANN has less than half of overlapping and for “good ” vs. “bad”, we see overlapping close to zero. Therefore, G-BRANN offers a data-driven and flexible way to derive tailored scenario analysis for risk management tasks and allows for a clear and economic plausible differentiation between macroeconomic states. The detailed results of every pairwise overlapping for every scenario analysis can be found in Appendix 2.D in Figures 2.D.1 and 2.D.2.

2.6 Conclusion

In recent years a broad stream of literature emerged which shows that mean LGD estimates are non-linearly connected to well-known drivers, see, e.g., Bastos and Matos (2022); Bellotti et al. (2021); Nazemi et al. (2021); Olson et al. (2021); Sopitpongstorn et al. (2021) or Xia et al. (2021). The drivers of the LGD distribution's precision (variance) are considerably less investigated as noted by Gambetti et al. (2019). They find that there are several variables with effect on the precision by using a generalized linear beta regression. We extend this approach by allowing non-linearity in mean *and* precision by combining the generalized linear beta regression with a neural network structure. This allows us to incorporate these little-noticed characteristics such as bounded support, skewed distribution, and heteroskedasticity directly into our modeling framework. Furthermore, we derive novel trainable activation functions to address the bounded support problem in the LGD distribution's mean and precision. We implement a data-driven way to characterize the actual shape of the precision predictions which increases the robustness. By accessing Moody's Default and Recovery Database from January 1990 until March 2021, we incorporate the most recent evidence in market-based LGD realizations. We observe strong non-linearity in the prediction of the precision parameter. Therefore, especially this parameter benefits from non-linear modeling.

Modeling the precision and thus, the form of every obligor's LGD distribution enhances the capability of risk managers in several important ways. First, lower and high quantiles can be used to derive good and bad scenarios in a data-driven way. Therefore, the impact of portfolio losses beyond expectation values can be quantified. Hence, our approach provides a flexible and data-driven tool to derive conservative estimates, i.e., higher quantiles, concerning regulator's margin of conservatism. Second, by comparing the individual distributions of obligors, risk managers can reveal differences in the obligor's risk profiles by comparing extreme losses, for example in terms of Value-at-Risk (VaR). This enables banks to better quantify the riskiness of their business in terms of potential losses. Our scenario analysis reveals that the distributions modeled by the beta distribution lack of distinctiveness compared to G-BRANN. Thus, scenario analysis with beta regression could lead to inadequate loss estimation. Furthermore, the application of BRANN and G-BRANN to workout LGDs would be interesting, as they entail similar challenges. As the data on this kind of LGD is considerably larger, one could even consider a multilevel approach by fitting our proposed methods on different seniority levels, industry sectors or default types.[20]

Our approach of combining well-known statistical methods with neural networks and the novel derived activation functions can not only be used for credit risk-related problems but to more general and broader set of problems in business and economics, e.g., demand or retail forecasting.

[20] We thank an anonymous reviewer for suggesting this potential application of BRANN and G-BRANN.

2.A Descriptive statistics

Table 2.A.1: Descriptive statistics of LGDs across the whole sample.

	N	Min.	Median	Mean	Max	St.Dev.	Skewness
LGD	2315	0.50	68.00	61.40	99.99	28.11	-0.35

Notes : All displayed values except the sample size are expressed as percentages.

Table 2.A.1 shows the descriptive statistics of LGDs across the whole sample. Over the 2,315 LGDs we have a slightly negative skewed distribution with a median LGD of 68 %. The following tables should provide an overview of the LGD distribution across the categorical values.

Table 2.A.2: Descriptive statistics of LGDs according to the seniority of the defaulted bond.

	N	Min.	Median	Mean	Max	St.Dev.	Skewness
Senior Secured	195	0.50	47.5	49.42	99.25	28.91	0.0594
Senior Unsecured	1599	0.50	65.0	59.59	99.97	28.36	-0.2206
Senior Subordinated	360	0.50	79.0	72.02	99.99	23.87	-0.9923
Subordinated	161	0.87	74.0	70.17	99.87	23.74	-0.9030

Dividing the LGDs in their seniority the picture changes in a few regards. While the skewness for senior secured and senior unsecured remains relatively close to zero, the skewness for senior subordinated and subordinated decreases close to -1, which indicates moderate skewness. Furthermore, the average LGD per category is different. Higher levels of seniority tend to have lower LGDs on average and at median. For the industry sector there are differences as well. High LGDs in particular can be observed for technology and for nonbank finance companies. By far the lowest mean LGD with low standard deviations are located in the utility sector followed by the banking companies. Those two sectors are the only sectors with highly positive skewness. This low mean LGD in the utilities sector corresponds to the high recovery rates in Gambetti et al. (2019). On the contrary low LGDs in the banking sector are quite different from the observed ones by Gambetti et al. (2019), but one must take into account that the used sample size in the banking sector in this paper is more than six times the sample size used in Gambetti et al. (2019). The remaining industry sectors show slightly higher or similar LGDs to the average LGD over the whole sample.

Table 2.A.3: Descriptive statistics of LGDs according to the industry sector of the defaulted bond.

	N	Min.	Median	Mean	Max	St.Dev.	Skewness
Banking	276.0	7.92	28.96	35.52	99.75	19.13	2.6608
Capital Industries	471.0	0.75	72.50	66.58	99.87	25.51	-0.6049
Consumer Industries	307.0	0.50	71.50	63.78	99.99	26.07	-0.6349
Energy & Environment	296.0	0.50	66.25	63.37	99.97	24.12	-0.5381
Media & Publishing	164.0	1.00	56.50	55.70	99.99	28.96	-0.0344
Nonbank Finance	261.0	14.00	90.00	74.06	99.87	29.63	-1.3282
REIT	17.0	36.65	76.48	66.03	98.12	21.37	-0.0421
Retail & Distribution	164.0	0.50	68.25	63.85	99.50	25.45	-0.6727
Technology	224.0	1.00	79.75	71.46	99.62	25.34	-1.2007
Transportation	84.0	4.75	77.75	66.94	98.25	23.72	-0.8157
Utilities	51.0	6.25	16.00	18.84	80.00	12.65	2.7352

Conditioning the LGDs on the default type there are major differences compared to Table 2.A.1 noticeable. First of all, there are some default types, that barely occur. Some of them occur only once or twice in the observed period of more than three decades. For the slightly bigger categories it is visible that payment moratorium has the lowest average LGD and the smallest standard division by far. The biggest category Chapter 11 shows the second highest average LGD. Only Chapter 7 provides higher average LGD, but also a very small sample size. Most of the conditional distributions are negatively skewed except the category distressed exchange, which is moderately positive skewed and show low average LGDs.

Table 2.A.4: Descriptive statistics of LGDs according to the default type

	N	Min.	Median	Mean	Max	St.Dev.	Skewness
Chapter 11	749	0.75	85.00	73.84	99.99	25.14	-1.2891
Chapter 7	7	54.00	96.00	89.32	99.47	16.19	-2.2681
Distressed exchange	554	0.50	29.00	39.82	98.00	21.33	0.8111
Grace period default	26	2.00	49.94	46.17	92.00	22.22	-0.0041
Missed interest payment	700	1.00	73.50	66.70	99.99	24.19	-0.6654
Others	94	1.00	67.00	62.36	99.75	29.36	-0.3277
Payment moratorium	35	14.87	16.83	16.79	17.95	0.55	-1.8553
Prepackaged Chapter 11	150	0.50	76.75	65.41	99.64	28.81	-0.5668

Notes: For comparability some categories are displayed consolidated, but feed to the network separately.

2.B Trainable activation functions

In the following the trainable activation functions from Section 2.4 are represented graphically. Figure 2.B.1 illustrates the impact of the steepness parameter a. For a larger value for a, $a = 3.37$, the curve is even steeper than the original exponential function. The value for Figure 2.B.1 is chosen according to the trained G-BRANN in Table 2.4. The trainable parameter q of the trainable softplus function in Figure 2.B.2 controls for the curvature of the function. For increasing q the trainable softplus function tends towards a relu activation function.

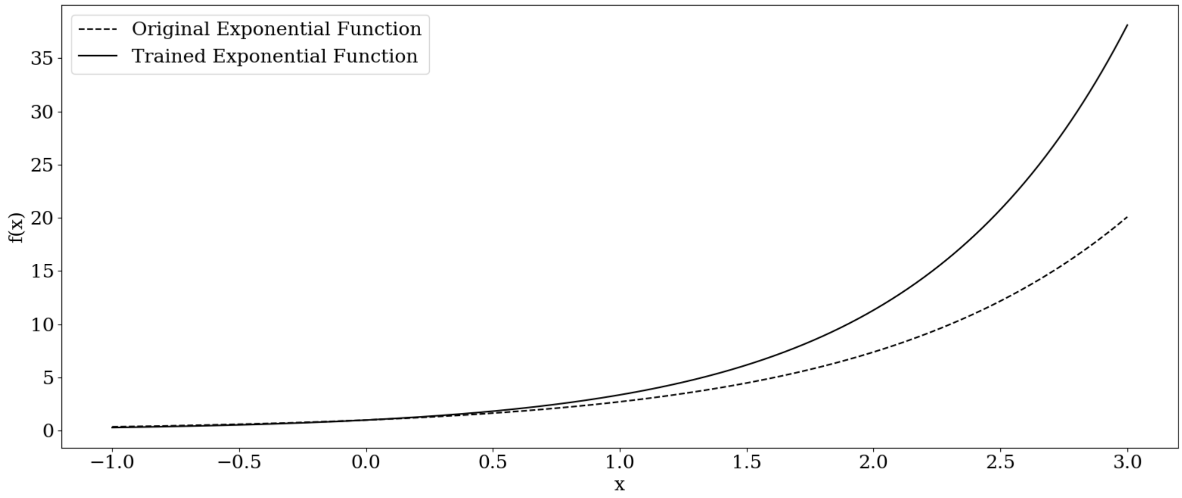


Figure 2.B.1: Trainable exponential function

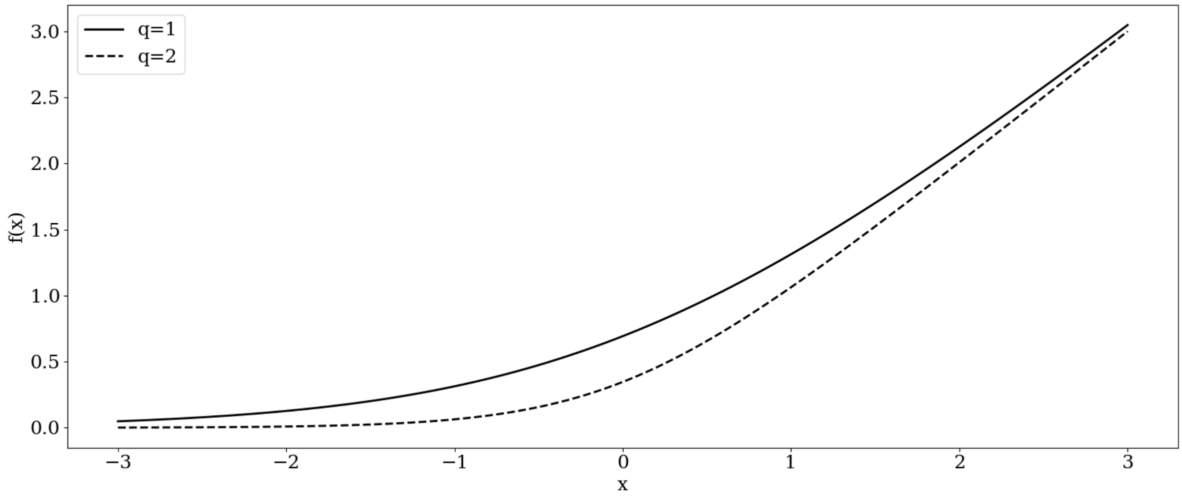


Figure 2.B.2: Trainable softplus function

The last of the proposed trainable activation functions is the trainable sigmoid function. Setting the parameters $c = 0$, $s = -1$ and $h = 1$ it results in the ordinary sigmoid function bounding the output on an interval from 0 to 1. The parameter c shifts the function vertically as displayed by the upper function of Figure 2.B.3. Increasing the scale parameter s towards 0 the output tends to flatten the input so that changes in the input less affect the output. The last parameter h defines the upper bound of the output, so that the trainable sigmoid function can result in higher values than the ordinary sigmoid function. The following figure compares the original sigmoid function with the sigmoid function, which is trained for the robustness section. Here the chosen parameters are $c = 1.88$, $s = -1.45$ and $h = 2.68$.

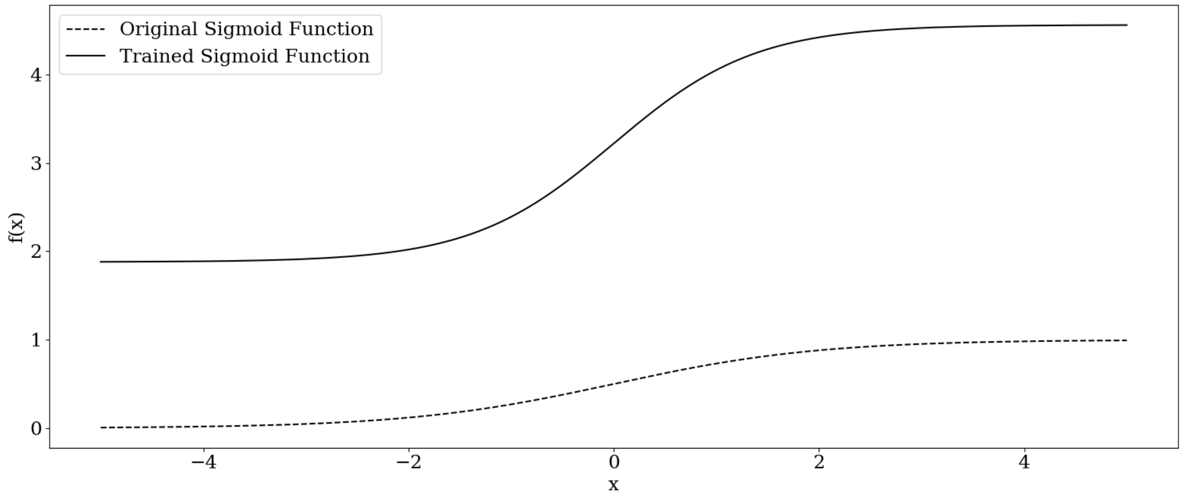


Figure 2.B.3: Trainable sigmoid function

2.C Robustness

Overall, this paper does not seek to conduct a horse race and contributes to the literature by finding a new best method of predicting market-based LGDs. Our aim is at the rationale behind the drivers of *mean* LGDs and the drivers of the *precision* of the LGDs, which is considerably less investigated. Hence, our neural networks can be seen as explanation models to unveil drivers of the mean and variance of the market-based LGDs. We want to use these relations to generate estimates in scenario analysis and derive implications. Therefore, they are not built as a *prediction* model for predicting future *mean* LGDs, but as a non-parametric way to unveil hidden relationships between drivers and the market-based LGD distribution. However, to counteract the concerns that our derived neural networks are not suitable to predict future realizations, we conduct an out-of-time benchmark exercise.

For this purpose we split the the whole sample into a training sample and a test sample. For the training sample only LGDs until end of 2007 are used. The remaining observations act therefore as an out-of-time sample. For BRANN, G-BRANN, Neural Networks and Random Forests we optimized the hyperparameters using 5 fold cross-validation and run each fold 10 times to ensure stable results for every fold. For all models with hyperparameters we draw 500 constellations each by a random search approach. Table 2.C.1 shows the setup and the resulting parameters for the robustness section.

For each model type we choose those hyperparameters, which return the lowest negative log-likelihood or mean squared error (MSE), respectively, averaged over the 5 hold out folds. The (extended) beta regression and (G-)BRANN are optimized by the mean of the negative log-likelihood in Equation (2.4), while the objective of the remaining models is to minimize the MSE. Since the Pseudo-R^2 is based solely on the μ-part of the (extended) beta regression and (G-)BRANN, which represents the predicted LGD, it can be calculated for all models. However, this does not apply to the log-likelihood calculation. For the out-of-time comparison we form a portfolio of 100 randomly drawn bonds and evaluate the MSE, Pseudo-R^2 and, if possible, the log-likelihood. This procedure was repeated 10 times and their average is provided in Table 2.C.2.

Comparing the values for the log-likelihood in- and out-of-time, we observe that in both samples one of our neural network approaches outperform the linear beta regressions. Therefore, we can argue that the good performance illustrated in Table 2.5 can be recovered when we focus

Table 2.C.1: Setup and final values of the hyperparameter search - robustness

Model	Parameter	Distribution	Final parameter
BRANN	Learning rate	$U^c \sim [0.001, 0.1]$	0.0893
	Dropout rate	$U^c \sim [0.05, 0.50]$	0.3289
	Hidden layer	$U^d \sim [1, 2]$	1
	Multiple	$U^d \sim [1, 8]$	1
G-BRANN	Learning rate	$U^c \sim [0.001, 0.1]$	0.0636
	Dropout rate μ	$U^c \sim [0.05, 0.50]$	0.4314
	Dropout rate ϕ	$U^c \sim [0.05, 0.50]$	0.1827
	Hidden layer μ	$U^d \sim [1, 2]$	2
	Hidden layer ϕ	$U^d \sim [1, 2]$	2
	Multiple μ	$U^d \sim [1, 8]$	1
	Multiple ϕ	$U^d \sim [1, 8]$	7
	Activation function	t-exp, t-sig, t-soft	t-sig
Neural Network	Learning rate	$U^c \sim [0.001, 0.1]$	0.0796
	Dropout rate	$U^c \sim [0.05, 0.50]$	0.4951
	Hidden layer	$U^d \sim [1, 2]$	1
	Multiple	$U^d \sim [1, 8]$	6
Random Forest	Number trees	$U^d \sim [10, 250]$	90
	Splitsamples	$U^d \sim [2, 10]$	3
	Leafsamples	$U^d \sim [1, 10]$	1
Regression Tree	Splitsamples	$U^d \sim [2, 10]$	7
	Leafsamples	$U^d \sim [1, 10]$	10
Ridge Regression	Regularizationparameter	$U^c \sim [0.0, 10]$	0.0023
Lasso Regression	Regularizationparameter	$U^c \sim [0.0, 0.02]$	0.0001
Elastic Net	Ratio	$U^c \sim [0.0, 1]$	0.1711
	Regularizationparameter	$U^c \sim [0.0, 0.02]$	0.0129

Notes: The table shows different values for the hyperparameter search. U^c labels the continuous uniform distribution, whereas U^d labels the discrete uniform distribution, in which the upper bound was excluded. For the random forest and the regression tree the splitsamples and the leafsamples refer to the minimum number of samples to split respectively to include in a leaf.

on future predictions. Overall the non-linear models recover the distribution of market-based LGDs best. While focusing only on the μ-part, i.e. only on mean predictions, we observe that the Random Forest performs best in-sample and third best out-of-time. This is similar to findings in the literature, see, e.g., Kaposty et al. (2020); Bellotti et al. (2021); Nazemi et al. (2021). However, as previously noted, the aim of this paper is not to predict the mean of market-based LGDs best as done by various studies, e.g., Bastos (2010); Loterman et al. (2012); Qi and Zhao (2011); Bastos and Matos (2022); Olson et al. (2021); Nazemi et al. (2021) among many others. The contribution of this paper is to model the precision of the market-based LGDs distribution in a straightforward non-linear way, which is a novelty in the literature of LGD modeling.

Table 2.C.2: Evaluation Metrics

(**a**) In Sample

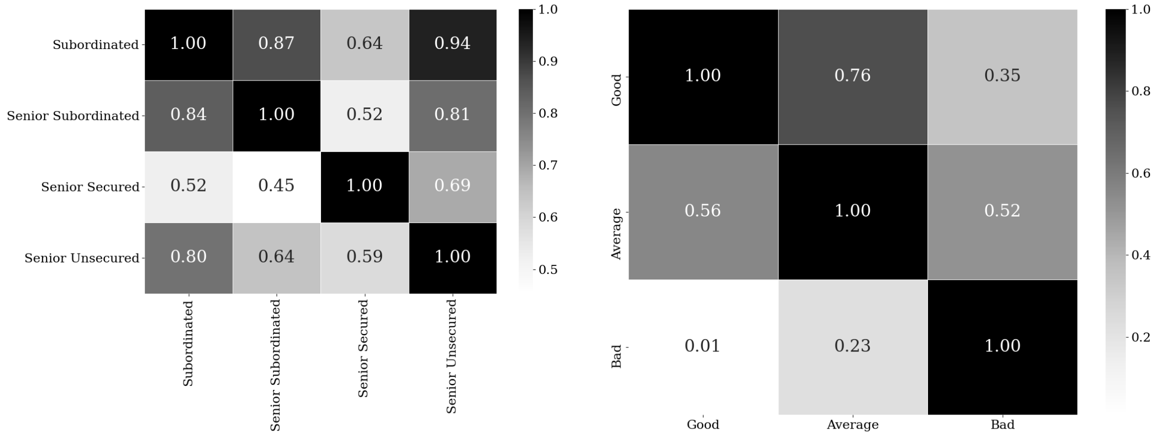
	Log-likelihood	MSE	Pseudo-R^2
Beta Regression	434.178	0.039	0.433
BRANN	441.049	0.039	0.447
Generalized Beta Regression	**487.562**	0.039	0.423
G-BRANN	<u>478.603</u>	0.037	0.485
Neural Network	-	0.045	0.398
Random Forest	-	**0.005**	**0.948**
Regression Tree	-	<u>0.023</u>	<u>0.681</u>
Linear Regression	-	0.039	0.472
Ridge Regression	-	0.040	0.459
Lasso Regression	-	0.039	0.471
Elastic Net	-	0.044	0.417

(**b**) Out of Time

	Log-likelihood	MSE	Pseudo-R^2
Beta Regression	<u>15.723</u>	0.057	0.339
BRANN	8.838	0.061	0.262
Generalized Beta Regression	-21.463	0.056	0.361
G-BRANN	**20.651**	0.058	0.286
Neural Network	-	0.115	0.173
Random Forest	-	0.053	0.426
Regression Tree	-	0.068	0.300
Linear Regression	-	0.062	0.379
Ridge Regression	-	<u>0.052</u>	<u>0.450</u>
Lasso Regression	-	0.061	0.386
Elastic Net	-	**0.047**	**0.461**

2.D Scenario analysis

The following provides the detailed results of our scenario analysis in Section 2.5. The estimated overlappings are illustrated via heatmaps. The lower triangle refers to the results of G-BRANN and the upper triangle to the results of the beta regression. In general, a lower value refers to less overlapping and vice versa.



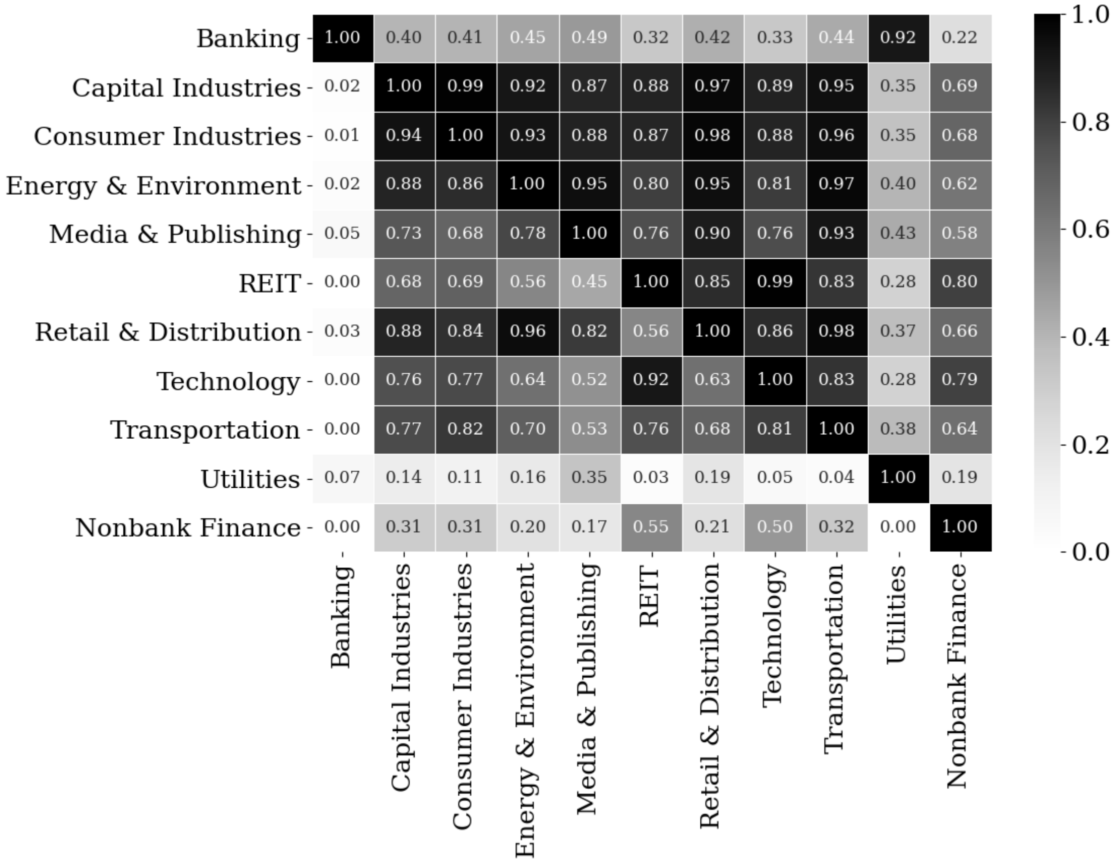
Seniority

Macroeconomic state

Figure 2.D.1: Overlappings | Seniority & Macroeconomic state

Notes: The figure show the overlapping of the estimated distributions of G-BRANN on the lower triangle and the beta regression on the upper triangle. A lower value indicates less overlapping.

Figure 2.D.1 shows the overlapping estimates for the different seniority types and macroeconomic states. Overall, we can see that G-BRANN has in every constellation a lower overlapping, which implies that G-BRANN helps to differentiate between seniority and industry type better than the standard beta regression. Please recall that the fit of G-BRANN is better in any seniority type or macroeconomic state. Therefore, we argue that the less overlapping better represents the underlying data.



	Banking	Capital Industries	Consumer Industries	Energy & Environment	Media & Publishing	REIT	Retail & Distribution	Technology	Transportation	Utilities	Nonbank Finance
Banking	1.00	0.40	0.41	0.45	0.49	0.32	0.42	0.33	0.44	0.92	0.22
Capital Industries	0.02	1.00	0.99	0.92	0.87	0.88	0.97	0.89	0.95	0.35	0.69
Consumer Industries	0.01	0.94	1.00	0.93	0.88	0.87	0.98	0.88	0.96	0.35	0.68
Energy & Environment	0.02	0.88	0.86	1.00	0.95	0.80	0.95	0.81	0.97	0.40	0.62
Media & Publishing	0.05	0.73	0.68	0.78	1.00	0.76	0.90	0.76	0.93	0.43	0.58
REIT	0.00	0.68	0.69	0.56	0.45	1.00	0.85	0.99	0.83	0.28	0.80
Retail & Distribution	0.03	0.88	0.84	0.96	0.82	0.56	1.00	0.86	0.98	0.37	0.66
Technology	0.00	0.76	0.77	0.64	0.52	0.92	0.63	1.00	0.83	0.28	0.79
Transportation	0.00	0.77	0.82	0.70	0.53	0.76	0.68	0.81	1.00	0.38	0.64
Utilities	0.07	0.14	0.11	0.16	0.35	0.03	0.19	0.05	0.04	1.00	0.19
Nonbank Finance	0.00	0.31	0.31	0.20	0.17	0.55	0.21	0.50	0.32	0.00	1.00

Figure 2.D.2: Overlappings | Industry

Notes: The figure show the overlapping of the estimated distributions of GBRANN on the lower triangle and the Beta Regression on the upper triangle. A lower value indicates less overlapping.

Figure 2.D.2 shows the overlapping estimates for the different industry types. Interestingly, G-BRANN shows very less overlapping for the banking industry with any other industry type. On the contrary, the beta regression shows medium overlapping. It is well known that the banking industry differs from other industry types due to their special business model and their impact on financial stability. It seems that the difference is also visible in the LGD estimates in our sample. Similar to Figure 2.D.1 G-BRANN shows considerable less overlappings and, thus, allows for a better differentiation between industry types.

Chapter 3

GAMME - Advances in Predictive Mean Matching

This chapter corresponds to a working paper with the same name (submitted to *The Scandinavian Journal of Statistics*, revised and resubmitted).

Abstract
Missing data is a widespread problem in almost any scientific and practical application. Incorrect imputation can bias results and make statistical inference invalid. In this study, we propose a new imputation method the **G**eneral **A**daptive **M**ean **M**atching **E**stimator (**GAMME**) to incorporate non-linearites and interactions in the well-known predictive mean matching (PMM) method. We use a neural network to reveal non-linear structures and incorporate this information into predictive mean matching utilizing an explainable artificial intelligence (XAI) method namely Accumlated Local Effects (ALE) plots. This reduces bias in regression coefficients and corrects confidence intervals to allow valid statistical inference. The capabilities of GAMME are demonstrated by benchmarking various state-of-the-art imputation methods in a simulation study. To the best of our knowledge, we are the first to incorporate non-linearities in the predictive mean matching framework via ALE plots.

Keywords: Missing Data, Multiple Imputation, Machine Learning, Explainable Artificial Intelligence (XAI)

JEL classification: C15, C45

3.1 Introduction

Missing data is a prominent problem in many scientific fields and in practice. Missing values can contain potential important information which can not be taken into account. Many statistical models or machine learning methods require complete data. Ignoring observations can reduce the sample size to a large extent, which can introduce bias or worsen performance, especially when models rely on large datasets like, e.g., neural networks (Alwosheel et al., 2018). One common field of research for missing data are surveys, see, e.g., Rubin (2003) or official statistics like in Boeschoten et al. (2019). Especially in these areas it is often needed to combine different datasets, which can lead to missing values, and is handled via data fusion as in Rässler (2003). King et al. (2001) report that in political science the data often suffers from high missing data rates too. Missing data is also a problem in clinical trials, see Little et al. (2012). Bryzgalova et al. (2024) analyze missing values in the context of firm characteristics and their impacts on asset pricing. They find that missing values are present in more than 70% of the observations. Therefore, deleting observations with missing values shrinks the whole dataset and lead to ignoring potential useful observations. Baesens and Smedts (2023) emphasize the importance of preprocessing the data including the handling of missing values for credit risk models. Improper imputation can affect the final results which can from a scientific view lead to retraction of a published paper, see, e.g., Su et al. (2023). The imputation of missing values results in a complete dataset without deleting information. There are many approaches to impute missing values. One common approach for continuous data is to use predictive mean matching (PMM) with the drawback of considering only linear terms. This paper contributes to the literature by extending PMM to account for non-linearity in the data without specifying the relationships explicitly in advance. Furthermore, a heuristic is proposed to include interactions as well. Our approach differs and outperforms current extension of PMM like Multiple Imputation through XGBoost (MIXGB) by Deng and Lumley (2023). We use a neural network and the functional decomposition property of Accumulated Local Effect (ALE) plots to create a predictive mean matching approach, which has very low bias and high coverage rates for a non-linear data generating process under missing completely at random and under missing at random. The structure of this paper is as follows. In Section 3.2, we give an introduction to the terminology of missing data and summarize the most relevant and state-of-the-art methods. Section 3.3 describes in detail our new method. The conducted simulation study can be found in Section 3.4 including the results. Section 3.5 concludes.

3.2 Background

The following section introduces missing data mechanisms, which differ by their complexity. Furthermore, we introduce well-known and state-of-the-art methods to handle missing data to give an overview of the current literature.

3.2.1 Missing data mechanisms

To introduce the field of missing data, we first classify different types of missing data into the missing data mechanisms. This systematization is founded by Rubin (1976) and classifies missing data due to their complexity. Assume a data matrix Z and a binary matrix R, which has the same dimension as Z. The elements of R, $r_{i,q}$, indicates, whether the corresponding i-th value of the q-th feature of Z, $z_{i,q}$, is missing (Van Buuren, 2018, Ch. 2):

$$r_{i,q} = \begin{cases} 0 & \text{if } z_{i,q} \text{ is missing} \\ 1 & \text{if } z_{i,q} \text{ is observed} \end{cases} \tag{3.1}$$

Therefore, Z can be divided into the observed and the missing part, $Z = \{Z^{obs}, Z^{miss}\}$. Z^{miss} contains every observation with missing values, the remaining observations are covered by Z^{obs}. Furthermore, these datasets can be divided into the dependent variable y and the independent features X such that $Z^{miss} = \{X^{miss}, y^{miss}\}$ and $Z^{obs} = \{X^{obs}, y^{obs}\}$[1]. Following (Van Buuren, 2018, Ch. 2) suppose a missing data model, which defines the probability to be missing for each element of the data. This is notated as $Pr(R = 0|Z^{obs}, Z^{miss}, \theta)$ with θ as the parameters of the missing data model. Then we can define observations as Missing Completely at Random (MCAR) if the probability to be missing is unrelated to any observed or unobserved variable. To illustrate this consider the following example. A survey is conducted in which age and income is collected. Now suppose missing data in income due to technical problems. In this case the probability to be missing is unrelated to age or income and this situation can therefore be classified as MCAR. Under MCAR the missing data model can be written as, see (Van Buuren, 2018, Ch. 2):

$$Pr(R = 0|Z^{obs}, Z^{miss}, \theta) = Pr(R = 0|\theta) \tag{3.2}$$

[1] The terminology of y and X often depends on the context. For an overview we refer to (Wooldridge, 2020, p. 21).

This case is in general very handy, see, e.g., Carpenter and Smuk (2021), but it is often considered as unreasonable, see, e.g, Little (1992); Liang and Wang (2023). To weaken the assumption of MCAR, observations can be classified as Missing at Random (MAR), in which the probability to be missing depends on Z^{obs} and θ[2]. Considering the example above MAR holds if with increasing age people are more likely to refuse to answer the question about their income. As stated in (Van Buuren, 2018, Ch. 2) then for the missing data model it holds, that:

$$Pr(R = 0|Z^{obs}, Z^{miss}, \theta) = Pr(R = 0|Z^{obs}, \theta) \tag{3.3}$$

Under MAR Z^{obs} can be used to impute Z^{miss}, such that valid statistical inference can be conducted. The last case is Missing Not at Random (MNAR), in which the probability to be missing depends either on the missing value itself or on the missing value and some observed values. In the survey example the missing values in income would be classified as MNAR if people with higher income tend to refuse to disclose their income. Following (Van Buuren, 2018, Ch. 2) we can formalize MNAR as:

$$Pr(R = 0|Z^{obs}, Z^{miss}, \theta) = Pr(R = 0|Z^{miss}, \theta) \text{ or } Pr(R = 0|Z^{obs}, Z^{miss}, \theta) \tag{3.4}$$

For MNAR special methods must be applied to overcome the problem of missing data, see, e.g., Hammon and Zinn (2020); Hammon (2022). Prominent approaches to handle MNAR are selection models, see, e.g., Heckman (1976) or pattern-mixture models, see, e.g., Glynn et al. (1986); Little (1993). Since these approaches require additional assumptions and complex models they are usually only used if the application of MAR imputation models is questionable.

3.2.2 Literature

Due to the broad application of missing data handling algorithms we give a overview of the most prominent and well-known algorithms. Imputation methods can roughly be classified into single imputation (SI) and multiple imputation (MI). Furthermore, there are methods which do no imputation and only use observed cases. Most of them are likelihood based approaches. One of the most widely used method in this category is the expectation maximization (EM) algorithm by Dempster et al. (1977). The goal of this method is to find distributional parameters, usually the parameters of a multivariate normal distribution, via iterating between the expectation step

[2] Technically it can further be divide into the ignorable and nonignorable case. This is beyond the scope of this paper. For further details, see, e.g., (Van Buuren, 2018, Ch. 2)

(E-step) and the maximization step (M-step). This likelihood-based method imputes the missing values in the E-step given the current imputation values. The M-step maximizes the likelihood given the current values, which is the basis for the next E-step (Enders, 2022, pp. 112-114). For an more in depth view on the EM algorithm, see, e.g., Dempster et al. (1977) or (Little and Rubin, 2019, pp. 185-212). Another approach which does no imputation is complete case analysis (CCA). It is also known as listwise deletion and it is a fast and easy way to handle missing data. This method deletes every observation in the dataset, which has at least one missing value. This can lead to very small datasets and a huge information loss during this process. As stated in Schafer and Graham (2002) in case of MCAR CCA is a valid option and can outperform single or multiple imputation techniques due to its efficiency, but this holds not for every case. With certain exceptions, if MAR holds CCA leads to biased estimates and incorrect standard errors. To overcome the problem of discarding observations and reducing the sample size single imputation (SI) can be applied. SI replaces each missing value by one single value. One common approach is mean imputation, see, e.g., Lin and Tsai (2020). In this case the missing value of a feature is replaced by the mean of the observed values of the feature. This results in a dataset with the same size of the original dataset, but with a changed distribution of the feature, see (Van Buuren, 2018, Ch. 1). An extension of mean imputation is conditional mean or regression imputation. This replaces the missing values by the prediction of regressing the feature with missing values on the remaining. Therefore, relationships can be taken into account, which can improve the imputation, see (Van Buuren, 2018, Ch. 1). Regression imputation has the drawback of only accounting for linear effects. Therefore, several approaches are invented using tree-based methods to overcome this. Vateekul and Sarinnapakorn (2009) propose a two step strategy, in which a classification tree is used to determine missing data patters and each pattern is imputed with a regression tree. Stekhoven and Bühlmann (2012) propose an imputation strategy which rely on iterative fitted random forests called MissForest. This random forests can incorporate non-linear effects and interactions. More concretely, this method imputes the missing values in the first place by an initial guess. Then a random forest is fitted on the observed data and is used to predict the missing values. This procedure is repeated until a stopping criterion is met. MissForest is a common choice in the context of missing data (Sun et al., 2023). For an overview of tree-based imputation methods we refer to Tang and Ishwaran (2017). Other approaches use neural networks to handle missing data. E.g., Choudhury and Pal (2019) use an autoencoder to impute missing values. One of the most prominent imputation approaches in the field of neural networks is the generative adversarial imputation network (GAIN) by Yoon et al. (2018). This is an imputation technique, which utilizes the capabilities of generative adversarial networks (GANs) to model conditional distributions. GANs are a type of generative models introduced by Goodfellow et al. (2014), in which two networks the generator and the discriminator compete against each other. The goal here is for the generator to produce synthetic observations, which can not be distinguished from real observations by the discriminator. Yoon et al. (2018) adapt this behavior, such that the generator in GAIN imputes the missing values conditioned on observed values and the discriminator should decide, if the imputations are real observations. To accomplish that, an additional parameter is introduced, which is necessary to model the data distribution uniquely. Another example for imputation by GANs is MisGAN by Li et al. (2019). Here a GAN learns the distribution of the data in the presence of missing data. Furthermore, it can be used as an imputation method for e.g. pictures. For a broader overview of machine learning approaches to impute missing data we refer to Emmanuel et al. (2021) or Sun et al. (2023). One advantage that SI methods have in common is that the result is one completed dataset. Then the imputed values are handled as if they are the true observed values.

As pointed out by (Van Buuren, 2018, Ch. 1) this often leads to an underestimation of standard errors and therefore can produce incorrect statistical tests. To overcome the problems of single imputation methods multiple imputation (MI) was proposed in the 1970s (Rubin, 2004). Here the missing values are imputed multiple times to generate several complete datasets. Each dataset is analyzed individually and the results can be pooled using the Rubin's rules (Van Buuren, 2018, Ch. 1). Another approach is to use multiple completed datasets sequentially to train a neural network as in Han and Kang (2022) which improves predictive performance. Many SI methods are extended to a multiple approach. One example is the bootstrap regression imputation method that extends the previous introduced regression imputation. Following (Van Buuren, 2018, Ch. 2) to estimate the regression coefficients a bootstrap sample is used and the the residual variance $\widetilde{\sigma^2}$ is calculated. Using these coefficients the missing values can be predicted. To make this imputation valid, to each imputed value a random draw of a normal distribution with mean of zero and variance of $\widetilde{\sigma^2}$ is added. Another extension is called Predictive Mean Matching (PMM) which uses a linear model to find similar observed values to the corresponding missing values and uses the observed values as imputation values. The major differences to the bootstrap regression imputation is that PMM has no distributional assumption and imputes only observed values whereas bootstrap regression imputation assumes normally distributed imputations and creates new values for imputing missing values. PMM is the basis for the proposed method in this paper and is described in detail in Section 3.3.1. There are different approaches to conduct PMM, see, e.g., (Van Buuren, 2018, Ch. 3). However, this approach has the drawback of not incorporating non-linearites or interactions, if not modeled explicitly. Therefore, Deng and Lumley (2023) use XGBoost instead of a linear regression as the underlying model. The uncertainty induced by the missing values is reflected by fitting the XGBoost on a random subsample. To extend the tree-based approaches to multiple imputation Doove et al. (2014) proposed a method which uses the classification and regression tree (CART) algorithm in a PMM manner to model complex relationships like non-linearities and interactions. To impute a value $z_{i,q}$ of the feature Z_q a CART is trained to predict Z_q conditioned on the remaining variables of the dataset $Z_{\backslash q}$. Instead of imputing $z_{i,q}$ by the prediction Doove et al. (2014) propose to sample from the observations in the terminal leave into which $z_{i,q}$ falls. The authors extend this approach to random forests (RF) as well. Since a RF consist of multiple trees, the terminal leaves of all trees have to be considered. Hence, every observation in the terminal leaves in which $z_{i,q}$ falls for every tree is serves as a possible imputation. From these values one is drawn randomly. For a broad overview of multiple imputation methods we refer to Murray (2018).

3.3 Methodology

In this section we introduce the **G**eneral **A**daptive **M**ean **M**atching **E**stimator (**GAMME**), which uses a neural network to model non-linearities and transfer these to the predictive mean matching by utilizing accumulated local effect (ALE) plots.

3.3.1 Predictive mean matching

Predictive mean matching is a comparable fast imputation method which needs no distributional assumptions. Since PMM is a non-parametric imputation approach it is fairly robust under different distributional assumptions, see, e.g., Kleinke (2017). Furthermore, as shown in Vink et al. (2014) PMM leads to plausible imputations even if the imputed data is semicontinuous. From a practical point of view, it is the default imputation method for continuous data in the popular R package `mice`, see Van Buuren and Groothuis-Oudshoorn (2011). As explained in (Van Buuren, 2018, Ch. 3) this method replaces missing values by observed values of the same feature, which are called donors. In a first step the data is divided in the observations with missing values Z^{miss} and the fully observed ones Z^{obs}. Furthermore, the number of donors d is defined from which in the last step the imputation is sampled from. Subsequently, the observed values of the feature with missing values Z_q^{obs} are regressed on the fully observed variables $Z_{\setminus q}^{obs}$ to obtain their prediction $\widehat{Z_q^{obs}}$. To account for the uncertainty due to missingness the dataset is completed M times. Therefore, the regression of Z_q^{obs} on $Z_{\setminus q}^{obs}$ is repeated M times either based on a bootstrap sample or fitted in a bayesian manner. The calculated coefficients of these regressions $\tilde{\beta}$ are used to predict the missing values Z_q^{miss} resulting in $\widetilde{\widehat{Z_q^{miss}}}$. To determine which observed value serves as a donor for a specific missing value, $\widehat{Z_q^{obs}}$ and $\widetilde{\widehat{Z_q^{miss}}}$ are compared based on a metric, usually the absolute difference[3]. A set of d possible donors are created for every missing value t which correspond to the d smallest distances between $\widehat{Z_q^{obs}}$ and $\widetilde{\widehat{Z_{t,q}^{miss}}}$. From this set one index r^+ is randomly drawn. In a final step the corresponding observed value $Z_{r^+,q}^{obs}$ replaces the missing value $Z_{t,q}^{miss}$. Despite the usage of predictive mean matching in practical applications, see, e.g. Blazek et al. (2021), the literature on theoretical aspects of this approach is sparse. However, Yang and Kim (2020) and Chlebicki et al. (2024) derive asymptotic properties for PMM, that build the foundation for future research on this topic.

3.3.2 Neural network

Contrary to linear regressions as used in PMM, neural networks are capable of modeling interactions and non-linearities. In fact, as shown in Hornik et al. (1989) a neural network with just one hidden layer can approximate a function up to an arbitrary precision. Following Apicella et al. (2021) feedforward neural networks usually consist of one input layer L_0, one or multiple hidden layer $L_1, ..., L_S$ and one output layer L_{S+1}. These layers are fully connected by weights matrices W_s and bias terms b_s, where W_s and b_s are the connections between L_{s-1} and L_s. Each hidden layer consists of a predefined number of neurons, which can differ across the layers. The number of neurons in the output layer depends on the task the neural network should perform. Each neuron performs a usually non-linear activation. To process the data matrix X, it is fed into the input layer, weighted by W_1 and b_1 and non-linear activated by an activation function $a(.)$. Therefore, this can be written as:

$$A_1 = a(XW_1 + b_1) \tag{3.5}$$

with A_1 as the activated weighted input data, which is the output of the first hidden layer and the input for the second hidden layer. This procedure is done repeatedly until the information reaches the output layer. The activation function in the output layer $o(.)$ is used to meet a

[3] In the literature this often referred as Type 1 matching, which is also the default case for PMM in `mice`, see Van Buuren and Groothuis-Oudshoorn (2011).

desired interval of the dependent variable y. The activation functions $a(.)$ and $o(.)$ are essential to model non-linearites. If the activation functions are the identity function, then the whole network consists of chained matrix multiplication and remains linear with respect to the input (Goodfellow et al., 2016, p. 168). The output $f(X)$ of a neural network with S hidden layers can be formalized as:

$$f(X) = o(A_S) = o(a(A_{S-1}W_S + b_S)) \tag{3.6}$$

In the training process the weights and biases are updated via gradient descent.

3.3.3 Accumulated local effect (ALE) plots

Accumulated local effect plots are an XAI method introduced by Apley and Zhu (2020). The goal here is to visualize the effect of each feature on the prediction of a model f. Compared to the partial dependent plots ALE plots are fast to compute and are unbiased in the presence of correlations. Following Apley and Zhu (2020) to calculate ALE plots for each feature X_q the range of the feature is divided into H buckets, also known as grid size, where the upper and the lower bound of the buckets correspond to the quantiles of the feature. For every bucket h each observation $x_{ih,q}$ of X_q, that falls into bucket h, the difference in the prediction is computed with $x_{ih,q}$ replaced by the lower and upper bound of bucket h $u_{h-1,q}$ and $u_{h,q}$. The set of observations in bucket h will be further denoted as S_h. For each bucket these differences are averaged and accumulated to get the (uncentered) ALE, see, e.g., (Molnar, 2022, Ch. 8). As stated in Apley and Zhu (2020) this can be formalized as:

$$g_{q,ALE}(X) = \sum_{h=1}^{h_q(X)} {n_h}^{-1} \sum_{X_i \in S_h} \left[f(u_{h,q}, X_{i,\setminus q}) - f(u_{h-1,q}, X_{i,\setminus q}) \right] \tag{3.7}$$

with $X_{i,\setminus q}$ being the observation i without the value of X_q, n_h the cardinality of S_h and $h_q(X)$ a function, that returns the index of the interval for each $x_{ih,q}$. $g_{q,ALE}(X)$ represents the vector of ALE values containing the ALE values $g_{q,ALE}(X_{i,q})$ for each observation i for feature q. Following Apley and Zhu (2020) $g_{q,ALE}(X)$ is subtracted by the mean over the individual ALE values $g_{q,ALE}(X_{i,q})$. Therefore, the centered version $G_{q,ALE}(X)$ has an average effect of zero:

$$G_{q,ALE}(X) = g_{q,ALE}(X) - N^{-1} \sum_{i=1}^{N} g_{q,ALE}(X_{i,q}) \tag{3.8}$$

According to (Molnar, 2022, Ch. 8) the ALE values can be interpreted as difference to the average prediction by changing the feature on the prediction f conditioned on a given observation.

3.3.4 GAMME

PMM has the advantage of imputing only observed and therefore reasonable values as well as the speed of fitting regressions. On the contrary, this method is based on a linear regression, which is not able to account for non-linearities, if not explicitly modeled. Hence, PMM can be inferior to more flexible methods, see, e.g, Murray (2018). To incorporate non-linearities without explicitly modeling them, one can use neural networks, but to account for the uncertainty they must be either bayesian or fitted on a bootstrap sample, which eliminates the advantage of a fast imputation. To overcome this problem, we propose to use the information modeled by a neural network to incorporate that into the standard PMM procedure. Suppose for illustration a data generating process of the following form:

$$y = \gamma_0 + \gamma_1 X_1 + \gamma_2 \zeta(X_2) + \epsilon \qquad\qquad \epsilon \sim N(0,1) \tag{3.9}$$

with X_1 being a feature with missing values, X_2 a fully observed feature and y the fully observed dependent variable. Let γ_0 be the intercept and γ_1 and γ_2 the corresponding feature effects, ϵ random noise and $\zeta(.)$ a non-linear function. Suppose the probability to be missing for each element of X_1 depends on X_2 and y such that in the notation of Section 3.2.1 $Pr(R_{X_1} = 0|X_2, y)$ holds. Following (Van Buuren, 2018, Ch. 3) in PMM X_1 is regressed on the observed X_2 and y. Then for each to be completed dataset this regression is repeated but with taking into account the uncertainty regarding the missingness, e.g., by fitting it on a bootstrap sample. But since X_2 is modeled linearly the imputation model is misspecified and the imputation will be biased. To correct this the imputation model have to take $\zeta(X_2)$ instead of X_2 into account.

Therefore, we propose GAMME to solve this problem. A detailed pseudo algorithm can be found in Appendix 3.A in Algorithm 1. First a neural network is fitted to capture all relationships between y and X. Since a neural network can only be fitted on fully observed values the neural network is fitted on y^{obs} and X^{obs}. The next step is to decompose the prediction into a additive structure. Following Apley and Zhu (2020) ALE plots have this decomposition property, such that for the prediction $f(X)$ it holds that[4]:

$$f(X) = \sum_{q=1}^{Q} G_{q,ALE}(X) \tag{3.10}$$

with $G_{q,ALE}(X)$ being the ALE values of the q-th out of Q features. As the ALE plots are able to decompose the prediction, the neural network have to converge and be tuned carefully. Furthermore, GAMME assumes that the decomposition as specified in Equation (3.10) is sufficient to decompose the prediction. If interactions should be taken into account, we refer to Section 3.3.5 for an extension of GAMME. As stated in Apley and Zhu (2020) effects of higher order are often considered as less important than main and second order effects.

Since many imputation algorithms are designed to impute just one feature they have to be extended. One common approach is to use fully conditional specification (FCS) as described in Van Buuren et al. (2006). FCS uses a chained approach in a Gibbs Sampler fashion in which the imputations are improved over the iterations. This extension is also used for GAMME to extend PMM. For FCS the features are sorted by the proportion of missing values from high to low. Subsequently, the dataset is imputed featurewise by random draws of the observed values of the corresponding feature to create one initial fully observed dataset.

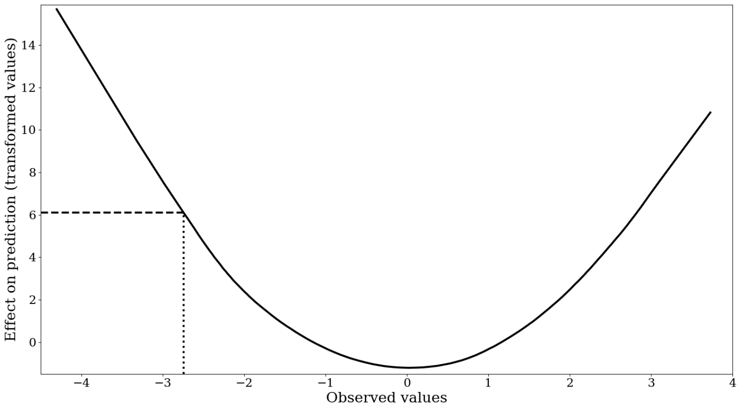


Figure 3.1: Illustrative example of the the ALE transformation

Notes: The figure shows an illustrative example of an ALE plot (solid line) and the ALE transformation. The observed value of −2.75 (vertical dotted line) is replaced with the ALE value of 6.11 (horizontal dashed line).

The next steps have to be done for each to be completed dataset and iterations I. For every feature with missing values k the current imputations are deleted to use PMM. The next steps are the extension which allows GAMME to incorporate non-linearities. Every feature X_q but k and the

[4] The decomposition by Apley and Zhu (2020) takes higher order effects into account to fully reproduce $f(X)$. Since we propose to detect the interactions and model them explicitly as described in Section 3.3.5 those are omitted in Equation (3.10).

dependent variable y are replaced with the corresponding ALE values $G_{q,ALE}(X)$ as illustrated in Figure 3.1 to obtain the transformed dataset $Z^{*(i-1)}$. This allows us to use predictions of the neural network, which is now linearized conditioned on the ALE values. Hence, this corrects the misspecification and the PMM can be used with the standard linear model. Now the algorithm proceeds with steps of PMM as described in, e.g. (Van Buuren, 2018, Ch. 3) but with $Z^{*(i-1)}$ instead of $Z^{(i-1)}$. Therefore, $Z_k^{*(i-1),obs}$ is regressed on $Z_{\setminus k}^{*(i-1),obs}$ and predicts $Z_k^{*(i-1),obs}$ to get $\widehat{Z_k^{*(i-1),obs}}$. To account for the uncertainty due to missingness a bootstrap sample $Z^{*(i-1),obs,b}$ is drawn and used to regress $Z_k^{*(i-1),obs,b}$ on $Z_{\setminus k}^{*(i-1),obs,b}$ to get $\tilde{\beta}$. These regression coefficients are used to predict $Z_k^{*(i-1),miss}$ resulting in $\widetilde{Z_k^{*(i-1),miss}}$. To find suitable donors, we use as distance the absolute difference of every observation r of $\widehat{Z_k^{*(i-1),obs}}$ and t of $\widetilde{Z_k^{*(i-1),miss}}$ in order to find predictions that are similar to those of the missing values. Now we can define a set of donors for every observation t as those observations with the d smallest distances. Subsequently one index r^+ is drawn randomly and the corresponding observed value $Z_{r^+,k}^{(i-1)}$ is used as imputation.
In terms of the example above $\widehat{Z_k^{*(i-1),obs}}$ is estimated by:

$$\widehat{X_1^{obs}} = \widehat{\beta_0} + \widehat{\beta_1} G_{2,ALE}(X^{obs}) + \widehat{\beta_2} y^{obs} \tag{3.11}$$

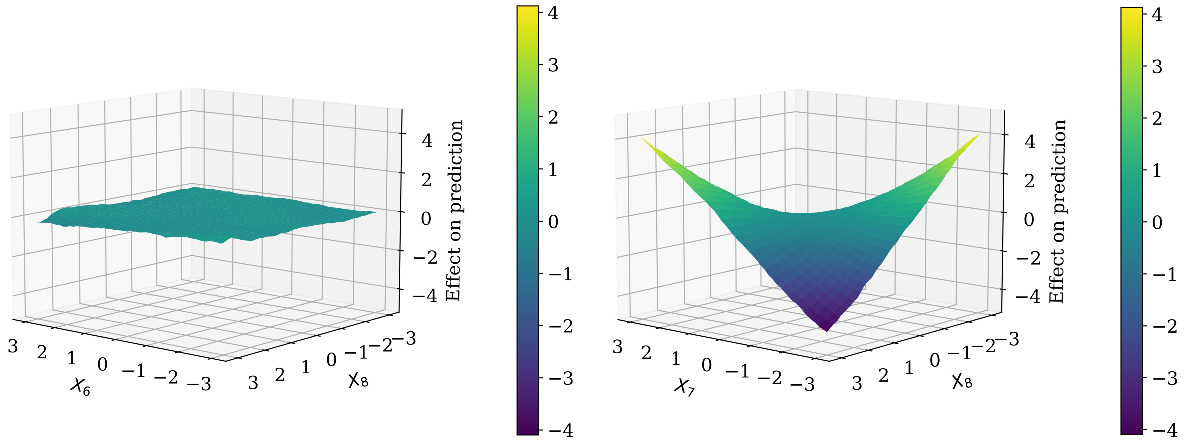
instead of:

$$\widehat{X_1^{obs}} = \widehat{\beta_0} + \widehat{\beta_1} X_2^{obs} + \widehat{\beta_2} y^{obs} \tag{3.12}$$

The adaption for the bootstrap regression is analogous. Hence, $\hat{\beta}$ and $\tilde{\beta}$ with GAMME contain non-linear information by using the transformed values instead of the original features.

3.3.5 GAMME with interactions

In many areas of research and practical applications interaction effects are considered in the modeling process to improve the model and get a deeper understanding of the data generating process, see, e.g. Caprio et al. (2007); Spilimbergo (2009) or Havrylenko and Heger (2024). As pointed out by Havrylenko and Heger (2024) one advantage of neural networks over classical statistical models is the inherent modeling of interactions. One way to visualize these interactions are second order ALE plots (Apley and Zhu, 2020). Similar to the ALE plots for individual features second order ALE plots visualize the the effect of the interaction of two variables on the prediction. These second order ALE plots show the isolated effect of the interaction without the first order effect and are therefore a powerful tool to investigate modeled interactions. However, according to (Molnar, 2022, Ch. 8) for some areas that cover only a few observations, e.g., at tails, these interactions can be unstable. This could lead to unreliable imputations especially when all interactions are included. To overcome this drawback instead of using the second order ALE plots directly and transform every possible pairwise interaction we use them as a heuristic to reveal important interactions. This allows us to only take into account important interactions and omit unimportant ones. For a first step a neural network is fitted on the fully observed data Z^{obs} to predict the dependent variable y^{obs}. Afterwards, every pairwise interaction can be examined by calculating second order ALE plots as presented in Figure 3.2. These plots are based on a grid of ALE values representing the effect on the prediction for the specific combination of the values of the interacting features. To rank the importance of the modeled interactions the ALE values are squared to pronounce high effects and reduce low interactions. For spurious interactions we observe that most ALE values are smaller than one as illustrated in Figure 3.2a. On the contrary real interactions show much higher values as displayed in Figure 3.2b. Therefore, squaring these values pronounce real interactions and reduce the effect of spurious ones. Due to extreme values for example at the tails of features even spurious interactions can have high effects on the prediction according to the second order ALE plots.



(a) Second order ALE plot - spurious interaction.

(b) Second order ALE plot - real interaction.

Figure 3.2: Illustrative example of second order ALE plots

Notes: The figure shows an illustrative example of second order ALE plots. The left plot shows a spurious interaction ($X_6 \cdot X_8$) with only a weak effect on the prediction. The right plot shows an "real" interaction ($X_7 \cdot X_8$) that is part of the data generating process and has great effect on the prediction.

Hence, we summarize the squared ALE values of the 2-D ALE interaction surface by calculating the median instead of the mean to account for such outliers that we denote $\nu_{\{q,j\}}$ for the interaction between feature q and j. We order these medians ν ascending to rank important over unimportant interactions. To determine a cut off value we use the Kneedle algorithm by Satopaa et al. (2011). This method is a general approach to find the "knee" based on the maximum curvature. All $\nu_{\{q,j\}}$ that exceed this cut off values φ are modeled explicitly as the product of feature X_q and X_j and are added to the dataset[5]. Afterwards, GAMME can be applied to the extended dataset as described in Section 3.3.4. We provide a pseudo algorithm, Algorithm 2, as step by step guide in the Appendix 3.A.

3.4 Simulation study

To evaluate the capability of imputing missing values with GAMME and restoring statistical inference a simulation study is conducted and compared to other popular imputation methods including PMM as described in Section 3.3.1.

3.4.1 Simulation setup

In this simulation study we consider two data generating processes (DGPs). The first one uses six independent features $(X_1, X_2, X_3, X_4, X_5, X_6) \overset{\text{iid}}{\sim} U(-3,3)$ to model the dependent variable y as follows:

$$y = -2.5 + X_1 + X_2 + X_3 + X_4 + X_5^2 + exp(X_6) + \epsilon \qquad \epsilon \sim N(0,1) \tag{3.13}$$

The second DGP extends Equation (3.13) by including interactions as well. Therefore, we added two additional features X_7 and X_8 to the data. This results in eight independent features $(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8) \overset{\text{iid}}{\sim} U(-3,3)$. The extended DGP covers in addition to the main effects of Equation (3.13) the interactions $X_3 \cdot X_4$ as well as $X_7 \cdot X_8$. Furthermore, we use different coefficients for the interactions to check the capability of our ALE interaction detection heuristic to find interactions of different strengths. Moreover, X_7 and X_8 have no main effect on y. This allows us to check the capabilities of GAMME for insignificant main effects. Hence, the data generating process can be formalized as follows:

$$y = -2.5 + X_1 + X_2 + X_3 + X_4 + X_5^2 + exp(X_6) + X_3 \cdot X_4 + \frac{1}{2} \cdot X_7 \cdot X_8 + \epsilon \qquad \epsilon \sim N(0,1) \tag{3.14}$$

Another interesting perspective are the interactions between features that contain missing values. These constellations offers a whole strand of missing data literature since the imputation of such

[5] Please note, that this heuristic will always find an interaction due to the kneedle algorithm always finding a turning point. We highly recommend to manually decide if there are interactions based on the second order ALE plots or due to an scientific assumption. If there interactions visible, one can use this heuristic to find the interactions with the biggest effect on the prediction.

missing values are not straight forward. There are some interesting approaches like passive imputation, the just another variable (JAV) technique or the substantive model compatible full conditional specification (SMCFCS). We refer to Seaman et al. (2012); Kim et al. (2015) and (Van Buuren, 2018, Ch. 6) for an overview and leave this topic for future research.

For the feature X_1 and X_2 we induce missing values, the remaining features and the dependent variable y are considered as complete. Following Deng and Lumley (2023) we define ξ_1 with $\xi_{i,1} = y_i + X_{i,3} \forall i$ and create the missing values in X_1 as follows:

$$Pr(R_{X_1} = 0) = \begin{cases} 0.6 & \text{if } \xi_{i,1} \text{ is in the top third of } \xi_1 \\ 0.1 & \text{if } \xi_{i,1} \text{ is in the middle third of } \xi_1 \\ 0.6 & \text{otherwise} \end{cases} \tag{3.15}$$

Therefore the average proportion of missing values in X_1 is roughly 43.33%. For X_2 we use a similar procedure and set ξ_2 as $\xi_{i,2} = y_i + X_{i,4} \forall i$. To study the impact of a smaller proportion of missing values we decrease the probability to be missing such that the average proportion of missing values in X_2 is 20%:

$$Pr(R_{X_2} = 0) = \begin{cases} 0.25 & \text{if } \xi_{i,2} \text{ is in the top third of } \xi_2 \\ 0.1 & \text{if } \xi_{i,2} \text{ is in the middle third of } \xi_2 \\ 0.25 & \text{otherwise} \end{cases} \tag{3.16}$$

To model MCAR we randomly drop a fraction of p_1 observations of X_1 and a fraction of p_2 observations of X_2, which can be formalized as:

$$\begin{aligned} Pr(R_{X_1} = 0) = p_1 \\ Pr(R_{X_2} = 0) = p_2 \end{aligned} \tag{3.17}$$

p_1 and p_2 are set to average proportion of missing values in the MAR case, such that the simulations only differ in the missing data mechanisms. We choose 10,000 observations as sample size as in Deng and Lumley (2023) and run this simulation 1,000 times. In every simulation run each imputation approach replaces the missing values with their imputed values. Based on these completed datasets the inference model is chosen according to the true data generating process in Equation (3.13) or respectively in Equation (3.14). This ensures that the calculated metrics and therefore the evaluation of the imputation approaches are solely based on their imputed values and not on a misspecified inference model. The ALE plots are generated with a grid size of $H = 1,000$ for a refined picture of the effect for this sample size. To investigate the robustness of GAMME we also conducted this simulation with standard normal distributed features. Furthermore, we consider 2,000 observations as sample size to investigate the properties of GAMME on a smaller sample with uniform and standard normal distributed features. In the case of the smaller sample size grid sizes of $H = 200$ and $H = 100$ are used to account for different number of buckets in the ALE plots.

3.4.2 Hyperparameters

Neural networks are very flexible and have multiple hyperparameters, which have to be set before fitting the network. To find the best hyperparameters, a hyperparameter search is conducted for the simulation via random search following Bergstra and Bengio (2012)[6]. For that, we assign each parameter a distribution, from which we draw randomly 100 constellations. Our network consists of four hyperparameters. The learning rate is the fraction of the gradient, which is used to update the weights and biases. We use a continuous uniform distribution with lower bound 0.0001 and 0.01 as upper bound. The bounds are chosen by dividing/ multiplying the default learning rate 0.001 of the used optimizer adam, see Kingma and Ba (2017), by 10. We consider one or two hidden layers, which are equally likely. To determine the number of neurons we use a multiple approach following Kellner et al. (2022) or Nagl et al. (2022). As baseline we use 32 neurons for the first hidden layer and 16 neurons for the second hidden layer if the neural network has two hidden layers. The multiple is the factor by which the baseline is multiplied. Therefore, a multiple of e.g. 3 results in neural network with $32 \cdot 3 = 96$ neurons if the neural network has one hidden layer and in the case of two hidden layers the first hidden layer consists of $32 \cdot 3 = 96$ neurons and the second hidden layer of $16 \cdot 3 = 48$. We use a discrete uniform distribution with whole numbers from 1 to 5. After a hidden layer we include a dropout layer introduced by Srivastava et al. (2014) to regularize the neural network and prevent overfitting. The dropout rate is drawn from a continuous uniform distribution with lower bound 0.0 and upper bound 0.5. Furthermore, early stopping is applied as an additional regularization method. We use three fold cross-validation on the first simulation and choose the network constellation with the smallest average validation mean squared error as our final hyperparameters, which are hold constant for the remaining simulation runs[7]. As activation function we use the ReLU function.

[6] Since the data generating process is the same for every simulation and differs only in the number of observations and the proportion of missingness the results of the hyperparameter searches are quite similar.

[7] To overcome random weight initialization every fold is fitted three times and the results of each fold are averaged.

Besides GAMME GAIN also has parameters, which has to be chosen in advance. For GAIN we follow Yoon et al. (2018) and use the official implementation[8] with their set hyperparameters. One hyperparameter α has to be evaluated via cross-validation. The same cross-validation as for GAMME is used, despite evaluating the root mean squared error and using a grid search as in Yoon et al. (2018). The parameter space for α is set to $\{0.1, 0.5, 1, 2, 10, 100\}$.

3.4.3 Metrics

To compare the approaches we use the relative bias (RB) and the coverage rate (CR) which are according to (Van Buuren, 2018, Ch. 2) appropriate metrics for evaluating imputation methods. As shown in (Van Buuren, 2018, Ch. 2) the root mean squared error is not a useful metric since it does not take into account the uncertainty due to missingness. The relative bias for a feature X_q RB_q is defined as:

$$RB_q = \frac{E(\widehat{\beta_q}) - \beta_q}{\beta_q} \tag{3.18}$$

with $\widehat{\beta_q}$ as the estimated regression coefficients and β_q as the true coefficient in the data generating process.

The coverage rate of feature X_q is the proportion of simulation runs in which β_q lies in the estimated confidence interval. In all simulations a 95% confidence interval is used. To evaluate the metrics for the MI methods a linear regression in the sense of Equation (3.13) respectively Equation (3.14) is fitted on each completed dataset. The estimated regression coefficients and standard errors are pooled according to Rubin's rules. Hence, the pooled regression results are further analyzed identical to a singular regression. For the MI methods $M = 10$ is used to produce M completed datasets. For the methods that rely on PMM (PMM, MIXGB, GAMME) the imputations are drawn from a set of $d = 5$.

3.4.4 Results

We compare eleven methods to the proposed GAMME to determine the capabilities of imputing such that the statistical inference is valid. To cover each class of missing data handling approaches we consider in this paper four single imputation methods, five multiple imputation methods and one likelihood based method. Furthermore, we also consider complete case

[8] https://github.com/jsyoon0823/GAIN

analysis, which does no imputation but is the default case in many studies, see, e.g, King et al. (2001). For the SI methods we use the mean and the regression imputation (Reg) which are a common choice for imputation, see Lin and Tsai (2020). To extend the SI methods by complex imputation methods we also compare MissForest which is a common choice across different scientific areas like medicine or nature, see, e.g., Nusinovici et al. (2020); Knell et al. (2020); Carmona et al. (2021); Gatti et al. (2021). Furthermore, GAIN is also taken into account to cover imputation based on neural networks. As likelihood method the EM algorithm[9] is considered since it is an essential part of the missing data literature, see Lin and Tsai (2020). On the multiple imputation side bootstrap regression imputation (BootReg) and CART as well as RF imputation as the multiple imputation extensions to regression imputation and MissForest are applied. Furthermore, PMM is used which is the linear counterpart and basis for the in this paper proposed GAMME method. In addition to that MIXGB is considered as it is one of the latest extensions to predictive mean matching which also incorporates non-linearities.

As stated in (Van Buuren, 2018, Ch. 1) the assumption of no relationship between the probability of missingness and the features is frequently considered as not realistic. Therefore, we discuss the MAR results in more detail. The first line of every table BD refers to before deletion and represents the result for the case without missing values. The goal for an imputation method is to produce results, which should be as close as possible to the before deletion case. All results are rounded to two digits for the relative bias and one digit for the coverage rate. Hence, column-wise a bold value indicates the best result, which is closest to the BD values, and a underlined value represents the second best result. If multiple methods have the same top ranking all of them are bold/ underlined.

Results without interactions

This subsection covers the results for the DGP as specified in Equation (3.13) that does only cover main effects. Table 3.1 shows the relative bias for 10,000 observations under MAR. As expected without missing values, the BD case, the bias is very close to zero. The complete case analysis in the second line of the table results in comparable low bias except for β_3. Mean imputation works well for β_1 and β_2, but induces substantial bias in the remaining coefficients. We can see that under MAR a few methods act quite similar. There is only a small difference in the results of EM,

[9] The original EM algorithm aims to estimate distributional parameters in the presence of missing data. To compare the EM algorithm with the other methods we use the R package `missMethods` by Rockel (2022), which applies imputation based on the EM algorithm.

BootReg and PMM. The regression imputation performs better for β_1 and β_2 but worse on the remaining coefficients. One reason for the high bias can be that these methods construct their imputation conditioned on the observed values, which gives the potential to perform well under MAR, but only if the imputation model is capable of modeling the data generating process. Since many of the imputation models rely solely on linear terms they potentially induce biased imputations by over- or underestimating effects. The complex imputation methods reduce the bias, especially in β_5 and β_6 which represent the coefficients for the non-linear components of the data generation process in Equation (3.13), compared to the linear approaches but also suffer from substantial bias. GAIN works especially well for β_5 and β_6 but results in high bias in β_1 and β_2. CART, RF and MIXGB underestimate all coefficients by at least 2% except for β_6. MissForest underestimates all coefficients with no missing values and overestimates the coefficients of those features that suffer from missing values. Furthermore, it is clearly visible, that the proposed method GAMME has very low to no bias and dominates the other imputation methods in this simulation since it has the lowest bias for every coefficient.

Table 3.1: Average relative bias in % for 10,000 observations (uniform) under MAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	0.01	0.02	0.01	0.01	0.00	-0.02	0.00
CCA	-2.56	-1.09	-1.10	-2.23	-1.34	-1.19	-0.67
Mean	3.95	-0.39	0.92	2.97	1.97	1.61	0.72
Reg	-15.75	9.08	5.42	1.60	1.09	-10.14	-4.73
PMM	-9.54	-30.13	-13.09	0.91	0.68	-6.38	-2.38
BootReg	-10.30	-31.44	-13.61	0.97	0.70	-6.60	-3.06
CART	-3.48	-20.03	-9.71	-2.97	-2.75	-2.51	-0.42
RF	-4.18	-27.13	-12.65	-2.42	-2.46	-2.62	-0.61
EM	-10.29	-31.45	-13.58	0.94	0.72	-6.60	-3.05
MissForest	-5.74	12.84	3.36	-3.21	-3.30	-3.62	-0.79
MIXGB	-4.07	-2.48	-2.44	-2.56	-2.19	-2.42	-0.65
GAIN	1.78	-12.98	-5.42	3.47	1.31	0.48	-0.38
GAMME	**-0.14**	**-0.29**	**-0.06**	**0.09**	**0.06**	**-0.11**	**-0.08**

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Besides the bias an important property for an imputation method is to correct the standard errors, which directly affects statistical significance. Therefore, Table 3.2 represents the coverage rate across the 1,000 simulation runs. Here the major drawbacks of most imputation methods get clear. BD offers coverage rates around 95%, which severs as benchmark for all other methods. For CCA there is substantial drop in the coverage rate for all features. This also holds for mean imputation which offers higher coverage rate for the features with missing values but there is a further decrease in coverage for the fully observed features. Therefore, significance tests can

Table 3.2: Coverage rate in % for 10,000 observations (uniform) under MAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	93.5	95.0	92.8	95.1	94.3	94.2	94.8
CCA	27.0	73.8	73.7	28.8	65.0	41.6	52.7
Mean	7.4	98.4	89.6	14.6	48.8	29.5	48.0
Reg	0.0	0.0	0.3	52.5	65.3	0.0	0.0
PMM	0.0	0.0	0.0	**95.3**	96.6	0.0	0.0
BootReg	0.0	0.0	0.0	**94.9**	96.2	0.0	0.0
CART	20.9	0.0	0.0	24.2	30.3	7.1	81.1
RF	8.5	0.0	0.0	48.7	46.1	5.4	73.4
EM	0.0	0.0	0.0	85.5	88.4	0.0	0.0
MissForest	0.1	0.0	9.0	8.0	6.3	0.0	29.9
MIXGB	2.9	15.7	24.2	19.3	32.8	3.1	53.5
GAIN	18.9	14.7	16.4	17.4	21.9	46.1	21.3
GAMME	**94.9**	**91.7**	**93.0**	94.1	**95.2**	**94.8**	**93.2**

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

not be trusted and should be avoided. Regression imputation, MissForest and GAIN offer low coverage rates as they are all singe imputation methods. The linear MI methods BootReg and PMM as well as EM have two coefficient whose coverage rates are near the desired result of BD. This could be due to feature which are dominantly modeled by these imputation methods. The complex MI methods CART, RF and MIXGB improves the coverage for β_6 but are problematic for the other coefficients. Only GAMME is able to restore the missing values such that the coverage rate is close to the BD case. GAMME offers the best coverage rate for every feature but one. For this particular variable GAMME ranks second best with only 1% less coverage rate than the BD case. Therefore, GAMME offers the opportunity to have both desired properties: low bias and high coverage rate.

The results of MCAR can be found in Tables 3.B.1 to 3.B.2 in Appendix 3.B. In this case CCA performs best or second best. The reason for that is that according to Carpenter and Smuk (2021) under MCAR CCA is identical to taking a random subsample from the original data, which results in bias free regression coefficients. For mean imputation the regression coefficients are only slightly worse than CCA. The linear methods model X_3 and X_4 well with low bias but remain problematic for the features with a non-linear effect and for the features with missing values. GAIN and MIXGB constantly underestimates the coefficients but less than the linear approaches. The picture for MissForest is very similar to the MAR case. There is an overestimation in the features with and an underestimation in the features without missing values observable. On the other hand GAMME performs equally well with negligible bias. This also holds for the coverage rates. Only CCA, mean imputation and GAMME can produce coverage rates similar to the BD case whereas CCA and mean imputation have a slight advantage. The remaining imputation approaches have very low coverage rates especially for the features with missing values. However, the coverage rates for the linear features X_3 and X_4 increased to a high level near the desired BD case or even further for the linear imputation methods. Taking the results from all simulations into account GAMME is the only method, which works very good under MCAR and MAR for bias and coverage rate.

The detailed results for the normally distributed sample can be found in Tables 3.C.1 to 3.C.4 in the Appendix 3.C. The overall conclusion drawn on the uniform simulations still holds. However, in some cases there are minor performance deteriorations, but GAMME still yields to competitive results. The results for the small sample size can be found in Appendix 3.D and 3.E in which the results for the individual settings are displayed in Tables 3.D.1 to 3.E.4. The reduced sample size decreases the performance of GAMME only slightly. The largest relative bias of GAMME is -1.83% whereas in most cases the absolute relative bias is below 1%. In terms of coverage rates GAMME still performs exceptionally well and provides the best trade of between bias and coverage rates across all settings. The changes in the small sample results due to the decrease of the grid size H are negligible..

Results with interactions

This subsection discusses the results for the second DGP as formalized in Equation (3.14). We use the in Section 3.3.5 proposed heuristic to detect interactions between features. Our simulation shows that both interactions $X_3 \cdot X_4$ and $X_7 \cdot X_8$ are successfully detected in almost every simulation run. Since our approach serves only as a heuristic in a few simulation runs additional spurious interactions is detected as well. In these cases this interaction is modeled and added to the dataset. We decided to use this procedure as one would do it similarly in a practical application in which the true data generating process is unknown. However, in the rare cases that an interaction is detected that features X_1 or X_2 this interaction is not included because as described above interaction with features that contain missing values need special methods to take care of. These cases occurred with a maximum of 1.1.% and only for the small sample size under normally distributed data. We leave this extension for future research.

The following Tables 3.3 and 3.4 describe the results for the second DGP with 10,000 uniformly distributed observations under MAR. Comparing the average relative bias of the imputation methods with the BD case in Table 3.3 we observed that using GAMME as an imputation approach results in a very low bias across all coefficients. For the intercept only GAMME can provide a bias close to zero whereas all other imputation techniques results in a bias of at least 2.60%. β_1 and β_2 are the coefficients for the features with missing values. Here GAMME and mean imputation provide the lowest bias. Here we can observe, that most imputation techniques have their highest biases in these two coefficients. For β_3 and β_4 GAMME offers the best results but many imputation approaches have low biases in these coefficients. β_5 and β_6 are the coefficients for the non-linear features of the data generating process. We can see, that GAMME provides the lowest bias and the imputation techniques that can handle non-linearities also provide good results. $\beta_{3,4}$ and $\beta_{7,8}$ represent the coefficients for the interactions. Here GAMME provides the lowest bias and GAIN the second best results. Since X_7 and X_8 have no main effect the relative bias can not be calculated. The raw bias $E(\widehat{\beta_q}) - \beta_q$ for $q \in \{7, 8\}$ is very close to zero with a maximum of 0.01% for all imputation approaches and simulations and therefore omitted in the bias tables.

Table 3.3: Average relative bias in % for 10,000 observations (uniform) under MAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$
BD	-0.05	0.00	0.00	-0.02	-0.02	-0.01	-0.01	0.00	0.00
CCA	-2.68	-0.79	-0.80	-1.69	-1.36	-1.06	-0.69	-0.61	-0.80
Mean	4.78	**-0.12**	0.56	2.26	2.04	1.51	0.88	0.69	1.01
Reg	-7.98	6.98	3.60	0.48	1.39	-5.83	-2.55	-7.02	-6.66
PMM	-5.09	-35.02	-15.78	0.29	0.87	-3.66	-1.43	-4.67	-4.36
BootReg	-5.19	-35.54	-15.92	0.33	0.84	-3.74	-1.63	-4.50	-4.24
CART	-2.60	-27.30	-12.36	-0.39	-0.17	-1.92	-0.24	-2.80	-9.26
RF	-2.86	-33.87	-15.50	-0.48	-0.22	-1.81	-0.44	-2.27	-4.16
EM	-5.19	-35.51	-15.92	0.34	0.84	-3.73	-1.63	-4.50	-4.29
MissForest	-3.97	9.18	2.51	-0.59	-0.12	-2.71	-0.50	-3.65	-8.06
MIXGB	-3.57	-10.74	-5.95	-1.61	-0.98	-1.71	-0.64	-4.73	-14.60
GAIN	3.68	-13.86	-6.20	5.02	3.60	1.06	0.48	0.29	0.54
GAMME	**-0.18**	-0.38	**-0.09**	**0.09**	**0.03**	**-0.05**	**-0.12**	**0.06**	**0.07**

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Table 3.4 shows the results for the coverage rates in the setting with interactions for 10,000 uniformly distributed observations. For the intercept, β_0, only GAMME is able to impute the missing values such that the coverage is in an reasonable range. The second best result is provided by CART with only 53.5%. Especially interesting are the results for β_1 and β_2. Here we can observe, that GAMME and mean imputation provides good coverage rates exceeding 90%. For the linear features X_3 and X_4 many imputation approaches offer coverage rates near the desired results of the BD case. For these coefficients the coverage rates drop for CCA and mean imputation. GAMME still offers reasonable results with a maximum of 1.3% difference to the BD case. Similar to the case without interactions of the first DGP GAMME provides the best coverage rates for the non-linear features X_5 and X_6. For these coefficients all other imputation approaches provide only low coverage rates. For the interactions many imputation approaches result in low coverage rates. That can be explained by the underlining models that are mostly designed only covering main effects. For the more complex imputation techniques only GAIN provides coverage rates exceeding 55%. CCA provides the second best results. Only GAMME is capable of identifying the interactions in a reasonable way and include them in the imputation process. Here the coverage rates are close to the desired results of the BD case. Taking a look at the coverage rates for the features without a main effect, X_7 and X_8, most imputation approaches offer coverage rates near the BD case. Interestingly GAIN has the lowest coverage rates of about 26%. GAMME is able to lower the too high coverage rates of its linear counterpart, PMM, resulting in a maximum difference to the BD results of only 1.2%.

Table 3.4: Coverage rate in % for 10,000 observations (uniform) under MAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$	β_7	β_8
BD	95.2	95.3	96.3	94.6	95.1	94.6	94.8	94.6	94.9	94.4	94.7
CCA	25.9	83.6	84.9	54.2	65.9	53.7	49.0	77.7	86.7	95.4	95.3
Mean	2.2	**98.9**	94.1	36.6	47.8	33.7	33.9	76.3	85.5	95.7	**94.6**
Reg	0.0	0.0	10.3	81.7	63.2	0.0	0.0	0.0	0.0	85.1	85.1
PMM	4.4	0.0	0.0	98.3	94.5	0.0	12.3	0.0	8.7	98.9	98.5
BootReg	3.8	0.0	0.0	98.7	**94.6**	0.0	5.2	0.0	11.7	99.0	98.8
CART	53.5	0.0	0.0	**93.9**	95.7	28.9	89.6	2.8	0.0	97.0	96.4
RF	46.8	0.0	0.0	97.3	99.0	35.3	88.3	9.6	12.4	99.1	98.7
EM	2.3	0.0	0.0	92.9	87.1	0.0	3.2	0.0	8.0	95.6	95.3
MissForest	6.5	0.0	29.5	76.7	83.5	2.0	55.6	0.0	0.0	87.0	87.3
MIXGB	17.1	0.0	0.1	68.5	84.4	33.1	64.4	0.0	0.0	**94.5**	93.4
GAIN	20.1	7.5	14.4	17.4	20.9	46.8	24.6	55.7	76.1	26.1	26.4
GAMME	**94.0**	90.5	**96.1**	95.9	94.4	**93.9**	**91.7**	**95.5**	**94.6**	95.6	95.4

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Tables 3.F.1 and 3.F.2 in Appendix 3.F show the results for MCAR with interactions. Similar to the DGP without interactions CCA offers the best results in terms of bias and coverage rates due to its equivalence of taking a random subsample as described above. GAMME outperforms all remaining imputation techniques besides mean imputation that is slightly better. This holds for the bias as well as the coverage rates. Analogous to the MAR case all imputation approaches but GAIN offers acceptable coverage rates for β_7 and β_8. In the case of normally distributed observations the results change marginally. As indicated by Tables 3.G.1 and 3.G.2 in Appendix

3.G under MAR GAMME offers the lowest bias for all nine coefficients. For the coverage rates GAMME provides the best or second best values besides for β_7 and β_8. For those coefficients the coverage rates are only 1% and 1.1% above the desired BD case. Changing the missing data mechanism to MCAR GAMME still provides good results in both metrics as displayed in Tables 3.G.3 and 3.G.4 in Appendix 3.G. The bias is with a maximum of -0.65% very low and all coverage rates exceed 91%. Comparing those results to other imputation approaches we see, that besides CCA and mean imputation GAMME is superior for almost every coefficient.
For robustness we challenge GAMME to impute on smaller datasets as well. The detailed results can be found in Tables 3.H.1 to 3.I.4 in Appendix 3.H and 3.I. We observe, that the biases increase and the coverage rates decrease slightly. In most constellations the absolute biases remain below 1% and are always below 2.5%. For the coverage rates GAMME often offers results close to the desired BD case and frequently outperforms most competitors especially for the MAR mechanism. The changes in the small sample results due to the decrease of the grid size H are negligible. Taking everything into account we can conclude that GAMME offers the best trade off between low biases and high coverage rates.

3.5 Conclusion

This paper introduces a new method called GAMME to improve imputation and allow valid statistical inference in the presence of non-linearities. Utilizing the ability of neural networks to approximate well to any function, we can incorporate this knowledge into the well-known predictive mean matching method. This can be achieved by using accumulated local effect plots and their property of functional decomposition. Transforming the feature values reduces the bias of regression coefficients and improves confidence intervals to draw statistical inference. Furthermore, GAMME is also challenged on interacting features. For this we propose a heuristic based on second order ALE plots to disentangle important from spurious interactions. These interactions are modeled explicitly and added to the dataset on which GAMME is applied. GAMME combines the field of machine learning by using neural networks with well established classical statistical approaches the PMM to create a new imputation method. In multiple simulations we show, that GAMME outperforms common imputation methods and advanced machine learning imputation approaches in nearly every covered situation. This also holds for the simulations featuring interactions. The proposed heuristic allows to reveal the real interactions of the simulation study in almost every iteration. Even for the cases where spurious interactions are modeled additionally the performance of GAMME remains exceptionally good.

Taking into account missing completely at random and missing at random scenarios GAMME provides the best trade off between unbiased regression coefficients and correct coverage rates and can therefore be a useful imputation approach in many scientific and practical applications. Future research could provide insights regarding non-linearities in the missing values and mixed type data. Especially the extension to interactions with missing values is of great importance. Forthcoming research could combine GAMME with existing extensions like the just another variable approach by Von Hippel (2009) or passive imputation as it is described in (Van Buuren, 2018, Ch. 6). Current more complex approaches like the substantive model compatible full conditional specification by Bartlett et al. (2015) could provide a fruitful path in further developments of GAMME. Furthermore, theoretical properties of imputation methods are of great interest. Future research could build on the pioneering work of Yang and Kim (2020) who derived asymptotic properties for the PMM imputation technique. Their approach could provide a rewarding path for deriving such properties for methods, that build on PMM like GAMME does.

3.A Pseudocode

This section contains the pseudocode for the proposed method in Section 3.3.4 and its extension in Section 3.3.5.

Algorithm 1 GAMME

```
Require: Dataset $Z = \{Z^{miss}, Z^{obs}\}$; number of to be completed datasets $M$; number of iterations $I$; number of donors $d$; number of features with missing values $K$; total number of independent features $Q$.
Split $Z^{obs}$ into dependent variable $y^{obs}$ and independent features $X^{obs}$
Fit neural network on $y^{obs}$ and $X^{obs}$
for $q$ in 1 to $Q$ do
    $G_{q,ALE}(X) \leftarrow$ Calculate ALE plot of feature $q$ of $X^{obs}$
end for
Sort the features in $Z$ from highest to lowest proportion of missing values
$Z^0 \leftarrow$ Create a copy of $Z$
for $k$ in 1 to $K$ do
    $Z_k^{0,miss} \leftarrow$ Draw a random sample of $Z_k^{0,obs}$ as a initial imputation
end for
for $m$ in 1 to $M$ do
    for $i$ in 1 to $I$ do
        for $k$ in 1 to $K$ do
            $Z^{*(i-1)} \leftarrow$ Create a copy of $Z^{(i-1)}$
            $Z_k^{*(i-1)} \leftarrow$ Delete the current imputation in $Z_k^{*(i-1)}$
            for $q$ in 1 to $Q$ but $k$ do
                $Z_q^{*(i-1)} \leftarrow$ Replace every value in $Z_q^{*(i-1)}$ with the corresponding ALE values
                    from the ALE plot $G_{q,ALE}(X)$[10]
            end for
            $\hat{\beta} \leftarrow$ Regress $Z_k^{*(i-1),obs}$ on $Z_{\setminus k}^{*(i-1),obs}$
            $\widehat{Z_k^{*(i-1),obs}} \leftarrow Z_{\setminus k}^{*(i-1),obs}\hat{\beta}$
            $Z^{*(i-1),obs,b} \leftarrow$ Draw a bootstrap sample of $Z^{*(i-1),obs}$
            $\tilde{\beta} \leftarrow$ Regress $Z_k^{*(i-1),obs,b}$ on $Z_{\setminus k}^{*(i-1),obs,b}$
            $\widetilde{Z_k^{*(i-1),miss}} \leftarrow Z_{\setminus k}^{*(i-1),miss}\tilde{\beta}$
            For every observation $r$ in $\widehat{Z_k^{*(i-1),obs}}$ and for every observation $t$ in $\widetilde{Z_k^{*(i-1),miss}}$
                calculate $\delta_{r,t} = |\widehat{Z_{r,k}^{*(i-1),obs}} - \widetilde{Z_{t,k}^{*(i-1),miss}}|$
            for every $t$ do
                Find the $d$ smallest distances out of $\delta_{r,t}$
                Draw randomly one index $r^+$ from these distances
                $Z^{(i)} \leftarrow Z^{(i-1)}$ with missing value $Z_{t,k}^{(i-1)}$ replaced with $Z_{r^+,k}^{(i-1)}$
            end for
        end for
    end for
    return $Z^{(I)}$ as the $m$-th imputed dataset
end for
```

Algorithm 2 ALE interaction detection

```
Require: Dataset $Z = \{Z^{miss}, Z^{obs}\}$; total number of independent features $Q$.
Split $Z^{obs}$ into dependent variable $y^{obs}$ and independent features $X^{obs}$
Fit neural network on $y^{obs}$ and $X^{obs}$
for $q$ in 1 to $Q-1$ do
    for $j$ in $q+1$ to $Q$ do
        $G_{\{q,j\},ALE}(X) \leftarrow$ Calculate second order ALE plot for feature interaction between $q$ and $j$[11]
        $\nu_{\{q,j\}} \leftarrow$ Calculate the median of the squared ALE values for the second order ALE plot $med((G_{\{q,j\},ALE}(X))^2)$
    end for
end for
Sort $\nu$ ascending
Apply Kneedle to find the cut off value $\varphi$[12]
for $q$ in 1 to $Q-1$ do
    for $j$ in $q+1$ to $Q$ do
        if $\nu_{\{q,j\}} > \varphi$ then
            $Z \leftarrow \{Z, X_q \cdot X_j\}$
        end if
    end for
end for
```

[10] An illustration of this transformation is displayed in Figure 3.1. If an observation lies between the upper and lower bounds of an bucket, the corresponding ALE values are interpolated linearly.

[11] Due to the instability of second order ALE plots for extreme values the grid size is reduced to $K = 50$ to cover more observations in those buckets.

[12] We used the following implementation to apply the Kneedle algorithm: https://github.com/arvkevi/kneed

3.B Simulation results large sample (uniform) without interactions

Table 3.B.1: Average relative bias in % for 10,000 observations (uniform) under MCAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	0.01	0.02	0.01	0.01	0.00	-0.02	0.00
CCA	**0.01**	$\underline{0.05}$	$\underline{-0.02}$	-0.01	$\underline{-0.01}$	**-0.02**	$\underline{0.01}$
Mean	$\underline{-0.06}$	**0.03**	**0.03**	**0.01**	0.03	$\underline{-0.05}$	**0.00**
Reg	-18.63	9.72	5.14	**0.02**	$\underline{0.01}$	-11.40	-3.73
PMM	-11.34	-30.34	-13.51	**0.01**	$\underline{0.01}$	-7.31	-1.93
BootReg	-11.98	-31.47	-14.00	0.02	$\underline{0.01}$	-7.33	-2.40
CART	-3.40	-18.99	-9.74	-2.83	-2.88	-2.44	-0.41
RF	-4.35	-26.15	-12.74	-2.86	-2.86	-2.98	-0.64
EM	-11.99	-31.48	-14.02	0.02	$\underline{0.01}$	-7.33	-2.39
MissForest	-6.01	13.91	3.05	-3.92	-3.95	-4.20	-0.84
MIXGB	-3.26	-2.30	-2.49	-2.15	-2.18	-2.11	-0.51
GAIN	-3.30	-10.54	-4.19	-0.76	-1.75	-1.37	-1.77
GAMME	-0.15	-0.29	-0.10	**0.01**	**0.00**	-0.11	-0.03

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Table 3.B.2: Coverage rate in % for 10,000 observations (uniform) under MCAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	93.5	95.0	92.8	95.1	94.3	94.2	94.8
CCA	$\underline{94.6}$	**94.7**	$\underline{95.2}$	$\underline{95.6}$	95.0	95.7	$\underline{95.4}$
Mean	**92.8**	99.4	96.2	**95.1**	95.5	**94.2**	**94.6**
Reg	0.0	0.0	0.0	85.1	85.1	0.0	0.0
PMM	0.0	0.0	0.0	99.0	98.7	0.0	0.4
BootReg	0.0	0.0	0.0	98.9	98.8	0.0	0.0
CART	18.6	0.0	0.0	24.0	22.5	5.1	80.7
RF	5.1	0.0	0.0	29.0	27.7	0.9	70.1
EM	0.0	0.0	0.0	93.9	**94.4**	0.0	0.0
MissForest	0.1	0.0	9.6	1.5	1.5	0.0	23.5
MIXGB	9.7	20.0	17.7	29.6	29.9	4.6	62.4
GAIN	20.6	15.4	19.6	34.0	31.4	54.3	24.8
GAMME	95.1	$\underline{92.0}$	**94.7**	96.0	$\underline{94.1}$	$\underline{94.8}$	93.8

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

3.C Simulation results large sample (normal) without interactions

Table 3.C.1: Average relative bias in % for 10,000 observations (normal) under MAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	0.00	0.01	-0.02	-0.05	-0.05	0.01	0.03
CCA	-2.82	-4.09	-4.12	-7.42	-4.84	-2.97	-1.85
Mean	1.34	-2.38	0.63	3.14	2.23	1.22	0.68
Reg	-8.95	17.13	9.32	3.26	2.07	-14.22	-7.21
PMM	-3.36	-20.02	-8.00	2.01	1.26	-7.27	-1.40
BootReg	-5.84	-25.44	-10.85	1.92	1.29	-9.21	-4.66
CART	-1.23	-15.51	-7.52	-2.71	-2.27	-2.40	-0.49
RF	-1.60	-22.20	-10.28	-2.62	-2.41	-2.54	-0.81
EM	-5.80	-25.44	-10.84	1.94	1.27	-9.15	-4.64
MissForest	-2.20	20.34	6.29	-3.66	-3.30	-3.62	-1.10
MIXGB	-0.93	**-0.32**	-0.60	-1.24	-1.01	-1.32	-0.49
GAIN	-0.34	-14.69	-4.93	5.09	2.15	0.64	-0.34
GAMME	**-0.01**	-0.79	**-0.47**	**0.13**	**-0.06**	**-0.30**	**0.23**

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Table 3.C.2: Coverage rate in % for 10,000 observations (normal) under MAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	95.5	94.8	95.5	94.5	93.9	95.7	95.3
CCA	9.9	20.1	18.3	0.5	11.0	22.3	35.7
Mean	58.0	77.0	95.1	31.7	61.6	72.5	79.7
Reg	0.0	0.0	0.0	26.6	51.7	0.0	0.0
PMM	2.6	0.0	0.1	77.7	89.9	0.0	48.3
BootReg	0.2	0.0	0.0	79.0	89.8	0.0	0.5
CART	65.0	0.0	0.0	50.9	61.3	33.0	86.3
RF	50.7	0.0	0.0	56.7	62.5	31.6	78.7
EM	0.0	0.0	0.0	65.9	80.4	0.0	0.2
MissForest	12.7	0.0	1.1	16.0	20.2	3.7	41.5
MIXGB	73.0	**93.6**	91.4	78.5	84.4	65.5	80.8
GAIN	33.3	8.3	25.4	16.6	27.7	63.9	33.4
GAMME	**94.3**	88.9	**95.3**	**94.7**	**94.3**	**93.9**	**90.4**

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Table 3.C.3: Average relative bias in % for 10,000 observations (normal) under MCAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	0.00	0.01	-0.02	-0.05	-0.05	0.01	0.03
CCA	0.01	**0.02**	-0.05	-0.02	-0.07	**0.01**	**0.04**
Mean	**0.00**	**0.02**	**-0.02**	-0.08	-0.06	-0.01	0.01
Reg	-8.48	17.78	8.21	**-0.05**	**-0.05**	-12.64	-5.21
PMM	-3.72	-19.40	-8.72	**-0.05**	**-0.05**	-7.51	-1.07
BootReg	-5.45	-26.17	-11.69	-0.08	**-0.05**	-8.12	-3.34
CART	-1.20	-13.99	-7.29	-2.27	-2.33	-2.40	-0.47
RF	-1.58	-20.33	-10.25	-2.76	-2.78	-2.76	-0.84
EM	-5.41	-26.21	-11.68	-0.07	-0.04	-8.09	-3.33
MissForest	-2.13	21.61	6.10	-3.75	-3.79	-3.91	-1.13
MIXGB	-0.78	0.55	-0.38	-0.92	-0.94	-1.20	-0.41
GAIN	-1.84	-12.10	-7.79	-3.11	-2.77	-0.85	-2.08
GAMME	0.07	-0.51	-0.26	**-0.05**	-0.07	-0.19	0.22

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Table 3.C.4: Coverage rate in % for 10,000 observations (normal) under MCAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	95.5	94.8	95.5	94.5	93.9	95.7	95.3
CCA	**95.6**	**94.6**	**94.7**	93.3	**93.8**	96.1	**94.9**
Mean	93.5	97.7	97.5	**94.5**	95.1	**95.6**	94.8
Reg	0.0	0.0	0.1	83.8	85.6	0.0	0.0
PMM	0.8	0.0	0.0	97.2	97.2	0.0	65.3
BootReg	0.2	0.0	0.0	97.3	97.4	0.1	1.7
CART	65.5	0.0	0.0	61.1	58.0	29.1	87.1
RF	49.6	0.0	0.0	51.5	49.9	19.9	77.0
EM	0.0	0.0	0.0	91.6	92.1	0.0	1.1
MissForest	11.9	0.0	0.4	12.1	10.5	0.5	39.0
MIXGB	76.5	91.4	92.7	85.0	85.6	71.5	84.1
GAIN	28.7	18.9	19.6	31.7	30.4	76.8	28.9
GAMME	94.3	92.2	94.0	94.6	95.3	93.9	93.4

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

3.D Simulation results small sample (uniform) without interactions

Table 3.D.1: Average relative bias in % for 2,000 observations (uniform) under MAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	0.01	-0.02	0.01	-0.05	-0.07	-0.01	0.01
CCA	-2.52	-1.12	-1.10	-2.32	-1.41	-1.18	-0.62
Mean	4.07	**-0.40**	0.87	2.97	1.91	1.63	0.77
Reg	-15.63	8.95	5.38	1.61	1.14	-10.12	-4.70
PMM	-9.56	-30.30	-13.10	0.90	0.65	-6.38	-2.35
BootReg	-10.21	-31.55	-13.56	0.99	0.66	-6.59	-3.03
CART	-4.41	-25.93	-11.80	-3.03	-2.89	-3.06	-0.49
RF	-4.17	-32.28	-14.48	-2.01	-2.11	-2.47	-0.66
EM	-10.28	-31.41	-13.55	0.96	0.67	-6.64	-3.02
MissForest	-5.65	8.40	2.23	-2.56	-2.68	-3.38	-0.82
MIXGB	-3.35	0.40	**-0.09**	-1.49	-1.12	-1.87	-0.56
GAIN	2.17	-8.64	-3.37	4.96	2.55	0.50	**-0.33**
GAMME (H=200)	**-0.26**	-0.90	-0.20	**0.43**	**0.23**	**-0.01**	-0.40
GAMME (H=100)	-0.26	-0.89	-0.19	0.43	0.25	0.00	-0.42

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.D.2: Coverage rate in % for 2,000 observations (uniform) under MAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	95.9	94.0	94.9	95.2	94.7	95.2	96.1
CCA	80.3	91.5	91.0	78.5	89.6	85.1	86.6
Mean	63.8	99.4	96.3	72.7	87.1	80.7	82.8
Reg	0.0	3.9	31.5	76.7	78.9	0.0	0.1
PMM	17.0	0.0	0.0	97.5	98.2	3.5	39.6
BootReg	9.1	0.0	0.0	97.3	98.4	2.2	19.6
CART	67.7	0.0	0.3	76.8	79.8	54.5	90.9
RF	79.0	0.0	0.0	92.8	91.9	77.2	**93.1**
EM	4.6	0.0	0.0	91.3	93.5	1.4	9.3
MissForest	29.5	5.1	72.5	63.8	63.7	27.4	64.4
MIXGB	65.4	**93.9**	92.0	82.4	85.8	65.6	79.7
GAIN	50.2	32.5	39.4	38.6	47.4	76.2	50.7
GAMME (H=200)	**94.3**	90.6	**95.0**	**94.2**	**95.4**	**96.5**	88.7
GAMME (H=100)	95.3	91.4	96.2	93.9	94.3	95.4	87.9

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.D.3: Average relative bias in % for 2,000 observations (uniform) under MCAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	0.01	-0.02	0.01	-0.05	-0.07	-0.01	0.01
CCA	-0.04	**0.01**	**0.04**	-0.02	**-0.10**	**-0.03**	-0.03
Mean	**0.00**	0.08	0.07	-0.09	-0.18	**-0.03**	**0.00**
Reg	-18.67	9.69	5.16	-0.14	-0.16	-11.42	-3.79
PMM	-11.46	-30.36	-13.50	-0.11	-0.13	-7.32	-1.95
BootReg	-12.02	-31.45	-13.99	-0.12	-0.15	-7.36	-2.44
CART	-4.44	-24.89	-12.15	-3.31	-3.43	-3.19	-0.51
RF	-4.69	-31.53	-14.79	-2.80	-2.89	-3.11	-0.81
EM	-12.05	-31.37	-13.94	**-0.05**	-0.14	-7.35	-2.45
MissForest	-6.50	9.42	1.67	-3.87	-3.91	-4.39	-1.08
MIXGB	-3.13	1.25	-0.45	-1.74	-1.78	-1.96	-0.59
GAIN	-4.16	-7.95	-5.98	-2.12	-2.32	-1.43	-1.93
GAMME (H=200)	-0.36	-0.82	-0.31	-0.09	-0.13	-0.19	-0.13
GAMME (H=100)	-0.39	-0.85	-0.29	-0.11	-0.09	-0.20	-0.14

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.D.4: Coverage rate in % for 2,000 observations (uniform) under MCAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	95.9	94.0	94.9	95.2	94.7	95.2	96.1
CCA	94.2	**94.5**	95.5	93.6	**94.5**	94.3	95.5
Mean	92.7	99.7	96.9	95.0	95.9	**95.2**	**95.9**
Reg	0.0	1.8	35.5	84.4	85.3	0.0	0.5
PMM	3.9	0.0	0.1	99.1	99.3	0.8	52.6
BootReg	1.9	0.0	0.1	98.2	99.0	0.4	32.4
CART	66.5	0.0	0.1	74.2	72.4	50.2	93.4
RF	72.6	0.0	0.0	88.1	86.0	61.5	92.8
EM	1.6	0.0	0.0	94.5	95.2	0.4	23.1
MissForest	20.6	1.6	78.0	45.4	45.0	10.4	60.2
MIXGB	64.6	90.2	90.9	79.8	79.6	58.5	80.6
GAIN	51.2	39.3	34.3	49.4	50.8	78.6	44.5
GAMME (H=200)	**95.3**	90.8	**95.2**	**95.3**	95.9	93.8	94.4
GAMME (H=100)	94.8	91.4	94.6	95.4	96.3	94.1	94.8

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

3.E Simulation results small sample (normal) without interactions

Table 3.E.1: Average relative bias in % for 2,000 observations (normal) under MAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	0.02	0.06	-0.02	-0.04	-0.09	0.05	0.01
CCA	-2.94	-3.96	-4.08	-7.46	-4.94	-3.15	-1.91
Mean	1.31	-2.23	**0.67**	3.23	2.13	1.25	0.65
Reg	-9.16	17.33	9.43	3.30	1.85	-14.53	-7.37
PMM	-3.53	-20.39	-8.10	2.06	1.10	-7.29	-1.60
BootReg	-6.05	-25.04	-10.68	1.93	1.07	-9.52	-4.88
CART	-1.82	-20.75	-9.46	-3.09	-2.68	-3.14	-0.84
RF	-1.79	-27.12	-12.06	-2.45	-2.44	-2.59	-1.04
EM	-5.95	-25.19	-10.53	2.02	1.11	-9.33	-4.76
MissForest	-2.50	15.85	4.84	-3.36	-3.23	-3.71	-1.43
MIXGB	-1.45	3.31	1.70	-1.38	-1.07	-1.88	-0.80
GAIN	-0.97	-15.24	-7.26	1.99	**-0.22**	**-0.36**	-1.67
GAMME (H=200)	**-0.53**	**-1.83**	-0.77	**0.60**	0.20	-0.94	**-0.28**
GAMME (H=100)	-0.54	-1.80	-0.75	0.56	0.16	-0.95	-0.26

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.E.2: Coverage rate in % for 2,000 observations (normal) under MAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	93.8	94.7	96.6	94.0	95.2	94.5	95.1
CCA	67.2	76.2	76.1	43.6	67.0	74.6	80.2
Mean	85.6	**95.2**	**96.2**	79.3	89.1	90.5	**91.6**
Reg	1.5	0.6	13.7	66.4	78.7	0.1	4.0
PMM	60.1	0.0	39.0	93.2	96.1	12.1	82.6
BootReg	18.8	0.0	14.7	**94.0**	96.1	5.7	28.3
CART	80.4	0.0	17.7	82.4	84.7	68.9	89.5
RF	85.3	0.0	4.0	90.5	89.7	82.1	**91.6**
EM	12.2	0.0	7.7	83.8	88.9	3.5	18.6
MissForest	56.1	0.0	54.1	64.6	66.8	44.2	66.9
MIXGB	76.9	74.7	87.0	85.4	86.7	76.9	81.5
GAIN	53.7	27.7	43.2	39.2	44.3	82.4	50.6
GAMME (H=200)	**91.1**	88.8	96.0	94.8	**94.6**	**90.7**	89.1
GAMME (H=100)	91.7	88.4	96.4	93.9	94.8	89.0	89.3

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.E.3: Average relative bias in % for 2,000 observations (normal) under MCAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	0.02	0.06	-0.02	-0.04	-0.09	0.05	0.01
CCA	0.11	0.18	**-0.01**	-0.11	-0.17	0.15	**0.03**
Mean	**0.02**	**0.13**	0.03	**-0.04**	-0.14	**0.01**	**-0.01**
Reg	-8.64	17.88	8.24	-0.06	**-0.11**	-12.87	-5.43
PMM	-3.82	-19.86	-8.92	-0.09	-0.13	-7.47	-1.23
BootReg	-5.62	-25.70	-11.52	-0.08	-0.12	-8.38	-3.57
CART	-1.67	-19.01	-9.48	-2.82	-2.88	-3.25	-0.74
RF	-1.80	-25.46	-12.29	-2.90	-3.02	-2.92	-1.12
EM	-5.58	-25.83	-11.44	-0.05	-0.14	-8.29	-3.49
MissForest	-2.45	17.12	4.14	-4.00	-4.12	-4.25	-1.53
MIXGB	-1.15	5.12	1.80	-1.02	-1.10	-1.81	-0.67
GAIN	-2.94	-9.27	-7.06	-3.66	-3.62	-1.36	-2.47
GAMME (H=200)	-0.06	-1.63	-0.86	-0.18	-0.23	-0.48	0.04
GAMME (H=100)	-0.06	-1.62	-0.86	-0.14	-0.22	-0.46	0.05

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.E.4: Coverage rate in % for 2,000 observations (normal) under MCAR without interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6
BD	93.8	94.7	96.6	94.0	95.2	94.5	95.1
CCA	94.9	**94.8**	95.1	**94.0**	94.1	94.2	95.9
Mean	92.6	97.7	**96.9**	95.4	**94.7**	**94.6**	**94.9**
Reg	1.9	0.3	23.4	85.6	86.2	0.1	11.0
PMM	57.4	0.0	31.7	96.4	97.0	8.2	87.8
BootReg	19.3	0.0	8.4	98.0	97.3	6.2	45.5
CART	85.1	0.0	15.2	82.7	84.4	67.3	90.5
RF	87.4	0.0	2.8	88.5	87.2	78.3	90.5
EM	15.1	0.0	5.0	93.1	93.3	5.1	37.9
MissForest	55.8	0.0	60.8	58.9	57.1	36.2	66.3
MIXGB	82.2	50.2	83.2	87.1	87.8	77.1	83.6
GAIN	50.8	45.5	37.4	50.6	50.5	85.4	48.7
GAMME (H=200)	**93.4**	88.4	94.7	95.2	94.1	92.9	92.1
GAMME (H=100)	92.9	89.1	94.6	94.7	94.8	92.2	92.0

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

3.F Simulation results large sample (uniform) with interactions

Table 3.F.1: Average relative bias in % for 10,000 observations (uniform) under MCAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$
BD	-0.01	0.00	0.01	0.03	-0.02	-0.01	0.01	-0.02	0.03
CCA	**-0.04**	**0.00**	**0.01**	**0.02**	0.00	**-0.02**	**0.01**	-0.01	-0.01
Mean	0.03	0.01	0.02	**0.04**	**-0.01**	0.01	-0.01	**-0.02**	**0.06**
Reg	-11.30	7.51	3.90	0.06	-0.05	-6.91	-2.28	-6.94	-6.87
PMM	-6.97	-35.33	-15.96	0.06	**-0.03**	-4.35	-1.31	-4.51	-4.44
BootReg	-7.20	-35.52	-16.04	0.05	**-0.03**	-4.38	-1.45	-4.42	-4.33
CART	-2.71	-26.52	-12.44	-0.34	-0.43	-1.97	-0.30	-2.90	-9.94
RF	-3.66	-33.47	-15.61	-0.81	-0.88	-2.36	-0.65	-2.48	-4.52
EM	-7.17	-35.50	-16.02	0.06	**-0.01**	-4.36	-1.45	-4.42	-4.34
MissForest	-4.99	9.92	2.29	-0.94	-1.03	-3.42	-0.73	-3.94	-8.67
MIXGB	-2.49	-10.34	-5.83	-0.72	-0.78	-1.48	-0.44	-4.68	-15.89
GAIN	-3.18	-8.06	-5.74	-1.94	-2.07	-0.71	-1.71	-0.73	-0.65
GAMME	-0.19	-0.38	-0.13	0.01	**-0.01**	-0.11	-0.04	**-0.02**	-0.01

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Table 3.F.2: Coverage rate in % for 10,000 observations (uniform) under MCAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$	β_7	β_8
BD	95.0	94.3	94.7	95.2	94.8	95.2	93.7	95.3	92.8	95.3	95.1
CCA	94.6	**95.1**	93.8	**95.1**	95.4	93.1	94.8	**94.6**	**93.3**	94.5	93.7
Mean	92.5	99.0	97.9	95.6	95.2	**94.9**	95.1	**96.0**	95.0	**94.6**	94.9
Reg	0.0	0.0	5.5	85.9	86.1	0.0	0.0	0.0	0.0	83.8	86.4
PMM	0.0	0.0	0.0	98.5	99.1	0.0	15.0	0.0	8.4	98.8	99.2
BootReg	0.0	0.0	0.0	98.4	98.8	0.0	8.5	0.0	10.3	98.4	99.1
CART	45.6	0.0	0.0	96.6	96.2	21.8	91.6	0.3	0.0	97.9	97.5
RF	22.8	0.0	0.0	94.9	93.9	11.5	75.4	2.8	6.0	98.4	98.5
EM	0.0	0.0	0.0	94.5	**95.1**	0.0	5.9	0.0	7.9	94.5	95.3
MissForest	1.1	0.0	32.8	72.5	69.5	0.1	38.2	0.0	0.0	86.6	89.3
MIXGB	44.2	0.0	0.0	89.6	88.6	42.3	78.4	0.0	0.0	**94.6**	96.3
GAIN	34.2	15.3	20.9	29.8	29.2	75.9	25.5	75.9	88.2	28.9	31.2
GAMME	**95.0**	88.8	**95.0**	95.6	95.7	94.5	**94.4**	96.1	**93.3**	**94.6**	**95.1**

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

3.G Simulation results large sample (normal) with interactions

Table 3.G.1: Average relative bias in % for 10,000 observations (normal) under MAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$
BD	-0.01	0.03	0.02	-0.05	-0.05	-0.01	-0.02	-0.02	-0.08
CCA	-3.34	-3.71	-3.72	-6.74	-4.92	-3.28	-2.02	-1.75	-3.64
Mean	1.84	-2.06	0.66	2.84	2.36	1.37	0.69	0.54	1.53
Reg	-6.35	15.27	8.27	2.01	2.29	-11.04	-5.60	-11.43	-10.10
PMM	-2.29	-25.50	-10.66	0.88	1.26	-5.62	-1.20	-9.12	-8.39
BootReg	-4.11	-28.49	-12.28	1.19	1.38	-7.06	-3.61	-7.30	-6.57
CART	-1.25	-19.14	-8.73	-1.37	-0.99	-2.41	-0.50	-3.91	-12.22
RF	-1.51	-27.00	-12.23	-1.61	-1.18	-2.22	-0.86	-3.27	-6.04
EM	-4.10	-28.46	-12.28	1.22	1.39	-7.03	-3.58	-7.27	-6.51
MissForest	-2.19	18.34	5.71	-2.41	-1.63	-3.54	-1.14	-5.57	-11.96
MIXGB	-1.29	-2.82	-1.75	-1.17	-0.62	-1.54	-0.73	-5.22	-13.24
GAIN	0.71	-13.01	-4.37	4.74	2.83	0.89	-0.41	0.27	1.03
GAMME	**-0.04**	**-0.86**	**-0.35**	**0.17**	**0.05**	**-0.44**	**0.27**	**-0.05**	**-0.01**

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Table 3.G.2: Coverage rate in % for 10,000 observations (normal) under MAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$	β_7	β_8
BD	95.6	94.4	94.7	94.9	94.7	95.6	94.3	94.0	94.0	95.8	94.9
CCA	2.4	29.0	28.1	0.4	9.2	16.6	28.7	79.1	75.1	95.0	96.0
Mean	31.7	82.4	**94.4**	40.3	55.1	68.6	77.3	92.6	90.1	**95.9**	**95.1**
Reg	0.1	0.0	0.0	56.0	48.6	0.0	0.1	0.0	1.4	86.2	86.9
PMM	27.5	0.0	0.0	**94.9**	91.0	0.0	59.8	0.0	14.3	98.1	96.8
BootReg	0.9	0.0	0.0	93.5	89.0	0.1	1.0	0.2	35.9	98.0	97.8
CART	66.0	0.0	0.0	84.6	88.6	32.9	83.5	20.7	0.2	97.6	96.2
RF	58.8	0.0	0.0	83.7	89.5	43.8	77.1	39.3	43.7	98.1	97.3
EM	0.4	0.0	0.0	82.9	78.4	0.0	0.3	0.0	31.6	93.8	93.4
MissForest	13.1	0.0	2.8	40.9	59.4	3.6	42.2	1.3	0.0	88.2	89.5
MIXGB	58.0	44.5	72.0	83.0	89.9	60.7	73.1	4.9	0.2	93.3	94.4
GAIN	39.9	7.0	30.4	17.5	26.2	67.6	33.3	82.4	88.2	38.9	39.7
GAMME	**93.7**	**88.0**	93.3	94.5	**94.5**	**91.7**	**91.3**	**94.6**	**94.7**	96.8	96.0

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Table 3.G.3: Average relative bias in % for 10,000 observations (normal) under MCAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$
BD	0.02	0.02	-0.03	-0.04	0.06	-0.01	0.01	0.02	-0.04
CCA	0.01	**0.01**	**-0.02**	**-0.04**	**0.06**	-0.05	**0.01**	0.00	-0.10
Mean	**0.02**	**0.01**	**-0.04**	-0.05	**0.06**	**0.01**	-0.02	**0.02**	**-0.03**
Reg	-6.87	15.78	7.27	-0.01	**0.06**	-10.23	-4.27	-10.23	-10.29
PMM	-3.04	-24.90	-11.18	-0.03	0.05	-5.97	-1.00	-8.39	-8.82
BootReg	-4.38	-28.85	-13.00	-0.02	0.07	-6.51	-2.74	-6.51	-6.56
CART	-1.19	-17.48	-8.66	-1.24	-1.12	-2.40	-0.47	-3.40	-12.99
RF	-1.59	-25.47	-12.36	-1.90	-1.80	-2.62	-0.93	-3.20	-6.63
EM	-4.38	-28.86	-12.97	-0.07	0.04	-6.51	-2.72	-6.47	-6.50
MissForest	-2.15	19.49	5.35	-2.45	-2.35	-3.96	-1.11	-5.21	-12.78
MIXGB	-0.86	-1.97	-1.71	-0.56	-0.49	-1.32	-0.48	-4.79	-15.13
GAIN	-2.54	-8.60	-7.47	-3.44	-2.93	-0.68	-2.12	-0.68	-0.72
GAMME	0.14	-0.65	-0.36	-0.06	0.04	-0.21	0.27	-0.07	-0.18

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

Table 3.G.4: Coverage rate in % for 10,000 observations (normal) under MCAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$	β_7	β_8
BD	95.5	96.2	94.8	95.8	94.1	95.6	96.0	94.8	95.0	95.5	95.7
CCA	**94.5**	**96.4**	**95.1**	95.3	95.0	95.1	95.4	95.3	93.8	**95.4**	**96.4**
Mean	93.5	98.2	95.9	95.4	94.9	**95.3**	**96.3**	94.4	95.3	95.2	94.7
Reg	0.0	0.0	0.2	86.6	86.4	0.0	0.2	0.0	1.2	87.0	86.0
PMM	4.5	0.0	0.0	98.0	97.3	0.0	72.3	0.0	8.8	96.8	97.3
BootReg	0.1	0.0	0.0	98.6	97.4	0.2	4.6	1.1	37.3	97.9	97.5
CART	68.2	0.0	0.0	84.8	87.4	29.1	86.3	31.4	0.2	97.6	96.6
RF	52.3	0.0	0.0	75.3	78.3	23.7	75.2	39.5	33.4	98.2	97.7
EM	0.1	0.0	0.0	93.5	**93.5**	0.0	4.0	0.7	30.7	93.8	94.2
MissForest	11.8	0.0	3.6	39.3	40.9	0.9	42.6	1.7	0.0	89.9	88.6
MIXGB	75.7	70.2	75.5	91.3	91.7	67.9	83.4	4.6	0.0	94.0	94.3
GAIN	29.4	19.6	23.0	30.6	32.8	84.6	27.3	87.3	93.3	34.3	34.0
GAMME	93.7	91.2	94.3	**95.8**	94.8	94.4	92.4	**95.1**	**94.8**	95.8	96.6

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined.

3.H Simulation results small sample (uniform) with interactions

Table 3.H.1: Average relative bias in % for 2,000 observations (uniform) under MAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$
BD	0.00	0.00	-0.03	0.02	0.01	-0.01	0.02	-0.01	0.01
CCA	-2.65	-0.78	-0.89	-1.73	-1.32	-1.07	-0.67	-0.68	-0.79
Mean	4.66	**-0.11**	0.49	2.23	2.00	1.42	0.89	0.70	0.95
Reg	-8.19	6.96	3.60	0.47	1.53	-5.94	-2.56	-7.09	-6.77
PMM	-5.21	-34.98	-15.75	**0.26**	0.91	-3.71	-1.44	-4.65	-4.37
BootReg	-5.32	-35.51	-15.91	0.32	0.87	-3.82	-1.65	-4.55	-4.34
CART	-3.09	-31.55	-14.20	-0.56	-0.22	-2.15	-0.31	-3.29	-6.54
RF	-2.41	-36.98	-16.78	-0.39	-0.12	-1.49	-0.44	-2.09	-2.82
EM	-5.31	-35.29	-15.86	0.35	0.84	-3.81	-1.64	-4.56	-4.37
MissForest	-3.67	5.02	1.27	-0.53	**0.05**	-2.35	-0.52	-3.58	-5.34
MIXGB	-2.82	-8.54	-3.72	-1.04	-0.17	-1.28	-0.53	-5.11	-8.16
GAIN	2.65	-13.74	-5.51	3.47	3.35	0.72	**0.25**	**0.03**	0.22
GAMME (H=200)	**-0.21**	-1.19	**-0.38**	0.46	0.44	**0.16**	-0.62	0.12	**0.18**
GAMME (H=100)	-0.21	-1.16	-0.34	0.44	0.42	0.17	-0.62	0.12	0.20

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.H.2: Coverage rate in % for 2,000 observations (uniform) under MAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$	β_7	β_8
BD	94.5	95.8	94.6	95.5	92.5	94.4	95.3	95.3	93.9	95.5	94.0
CCA	77.5	**93.2**	92.0	86.3	89.6	86.7	88.0	92.0	**94.0**	94.2	95.2
Mean	54.8	99.2	97.8	82.4	84.0	83.8	78.4	91.2	94.2	94.3	95.4
Reg	12.8	24.0	62.0	82.8	78.3	1.6	16.7	0.1	19.4	84.8	84.6
PMM	70.7	0.0	0.0	98.4	98.2	47.5	77.5	17.6	78.3	98.9	99.2
BootReg	68.7	0.0	0.0	98.3	98.3	43.4	72.8	18.7	79.2	99.3	98.8
CART	84.6	0.0	0.1	**95.4**	96.5	78.9	**93.7**	44.4	39.5	97.5	98.0
RF	92.9	0.0	0.0	98.1	98.9	93.6	97.2	80.3	93.3	98.5	98.9
EM	52.3	0.0	0.0	94.6	**91.9**	33.2	56.5	13.4	68.5	94.3	95.0
MissForest	59.0	48.8	84.7	87.2	86.8	54.5	74.5	18.7	35.2	89.9	90.0
MIXGB	77.6	6.4	64.6	88.8	**91.9**	83.4	85.2	5.7	12.2	**94.8**	**94.1**
GAIN	43.4	22.7	35.4	39.2	38.2	81.6	47.5	82.2	91.8	46.3	45.1
GAMME (H=200)	**94.6**	89.2	**95.2**	94.3	94.9	**94.9**	83.9	**95.0**	95.5	**94.8**	96.4
GAMME (H=100)	94.4	88.2	95.4	94.0	93.1	94.5	85.0	95.1	95.1	95.1	97.0

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.H.3: Average relative bias in % for 2,000 observations (uniform) under MCAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$
BD	0.00	0.00	-0.03	0.02	0.01	-0.01	0.02	-0.01	0.01
CCA	**0.02**	-0.07	**-0.01**	0.04	0.03	**0.00**	**0.01**	**-0.01**	**0.01**
Mean	0.12	**-0.02**	0.01	0.01	0.00	0.02	**0.01**	**-0.01**	0.04
Reg	-11.25	7.48	3.96	0.00	0.03	-6.92	-2.28	-6.95	-6.91
PMM	-6.90	-35.35	-15.92	**0.02**	**0.01**	-4.32	-1.29	-4.47	-4.41
BootReg	-7.13	-35.54	-15.98	0.01	0.00	-4.38	-1.45	-4.42	-4.39
CART	-3.38	-31.07	-14.27	-0.65	-0.62	-2.41	-0.43	-3.34	-7.06
RF	-3.43	-36.75	-16.93	-0.97	-1.00	-2.10	-0.75	-2.24	-3.10
EM	-7.10	-35.38	-16.01	-0.03	-0.01	-4.39	-1.42	-4.42	-4.39
MissForest	-4.99	5.53	1.05	-1.26	-1.20	-3.24	-0.97	-3.82	-5.83
MIXGB	-2.62	-7.84	-3.94	-0.55	-0.52	-1.51	-0.60	-5.22	-9.11
GAIN	-2.82	-8.75	-5.57	-2.09	-2.10	-0.75	-1.63	-0.78	-0.72
GAMME (H=200)	-0.43	-1.14	-0.51	0.01	-0.03	-0.25	-0.13	-0.03	-0.06
GAMME (H=100)	-0.45	-1.16	-0.50	0.00	-0.04	-0.25	-0.14	-0.05	-0.06

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.H.4: Coverage rate in % for 2,000 observations (uniform) under MCAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$	β_7	β_8
BD	94.5	95.8	94.6	95.5	92.5	94.4	95.3	95.3	93.9	95.5	94.0
CCA	94.7	**96.7**	94.9	94.4	94.2	94.8	94.6	96.1	93.4	94.9	94.3
Mean	91.9	99.3	97.2	94.2	95.5	95.3	96.3	96.0	94.8	96.0	**93.8**
Reg	1.0	19.1	53.9	85.3	85.9	0.0	19.6	0.0	17.0	86.8	87.5
PMM	46.4	0.0	0.0	98.5	98.7	30.1	82.3	17.8	78.0	99.3	98.4
BootReg	42.8	0.0	0.0	98.6	99.1	29.3	77.6	19.0	78.1	99.4	98.6
CART	82.6	0.0	0.0	96.4	96.8	72.3	**95.5**	40.5	34.5	98.4	97.2
RF	86.0	0.0	0.0	97.4	98.4	85.9	94.0	77.1	90.8	99.3	98.5
EM	31.4	0.0	0.0	**94.8**	94.7	22.6	65.2	13.7	67.3	95.1	93.5
MissForest	42.3	40.7	86.6	82.6	84.2	31.7	68.1	11.2	30.3	91.0	91.7
MIXGB	81.5	12.5	62.1	92.2	**91.2**	82.0	85.9	3.2	7.4	93.3	94.7
GAIN	63.9	34.4	30.9	43.9	45.4	90.7	44.0	88.2	93.1	44.0	43.0
GAMME (H=200)	**94.4**	88.4	**94.8**	94.6	94.2	**94.5**	94.3	**95.7**	**94.0**	**95.6**	95.4
GAMME (H=100)	94.6	88.5	94.2	94.5	94.3	94.6	94.3	96.3	93.8	96.2	96.2

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

3.I Simulation Results small sample (normal) with interactions

Table 3.I.1: Average relative bias in % for 2,000 observations (normal) under MAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$
BD	-0.01	0.12	-0.01	0.04	0.10	-0.02	0.01	0.04	-0.20
CCA	-3.35	-3.74	-3.66	-6.64	-4.76	-3.20	-2.14	-1.79	-3.83
Mean	1.77	-2.06	**0.69**	3.04	2.40	1.24	0.71	0.65	1.25
Reg	-6.47	15.30	8.35	2.04	2.40	-11.33	-5.64	-11.52	-10.44
PMM	-2.47	-25.85	-10.74	1.08	1.41	-5.72	-1.36	-8.75	-8.36
BootReg	-4.29	-28.26	-12.13	1.28	1.46	-7.42	-3.70	-7.44	-6.99
CART	-1.73	-24.04	-10.73	-1.67	-1.01	-2.99	-0.76	-4.47	-9.52
RF	-1.46	-31.28	-13.82	-1.25	-0.94	-2.08	-0.90	-3.08	-4.58
EM	-4.22	-28.28	-12.12	1.30	1.41	-7.27	-3.65	-7.37	-6.89
MissForest	-2.31	13.58	4.20	-2.26	-1.34	-3.46	-1.31	-5.72	-8.88
MIXGB	-1.63	**0.53**	0.90	-1.14	**-0.13**	-1.94	-0.95	-6.65	-9.66
GAIN	**-0.55**	-12.48	-5.09	1.65	0.64	**-0.23**	-1.68	-0.65	**-0.31**
GAMME (H=200)	-0.62	-2.49	-0.88	**0.76**	0.57	-0.99	**-0.55**	**-0.15**	-0.34
GAMME (H=100)	-0.60	-2.50	-0.91	0.79	0.60	-1.02	-0.50	-0.11	-0.24

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.I.2: Coverage rate in % for 2,000 observations (normal) under MAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$	β_7	β_8
BD	95.7	94.8	94.3	93.8	96.2	95.3	94.8	95.8	95.1	95.6	95.1
CCA	60.1	77.1	80.4	52.9	71.9	76.1	77.1	91.6	90.9	**95.6**	94.5
Mean	82.6	**94.8**	96.0	81.0	87.2	90.5	91.2	94.2	**95.2**	95.3	**95.1**
Reg	7.5	1.6	22.1	76.5	75.7	0.6	10.0	2.3	47.0	87.4	87.9
PMM	81.1	0.0	14.0	95.1	**96.8**	30.0	88.8	23.3	78.7	97.8	97.2
BootReg	44.2	0.0	6.3	95.4	97.1	13.6	46.4	40.4	85.1	97.3	97.6
CART	83.4	0.0	9.5	90.9	94.7	72.3	90.5	71.7	68.8	96.3	96.6
RF	91.2	0.0	0.9	96.2	98.1	89.1	**93.8**	89.1	92.8	97.9	96.9
EM	34.7	0.0	3.5	89.9	92.2	10.8	33.6	31.5	77.7	94.3	94.5
MissForest	62.0	2.1	61.1	75.8	86.2	48.7	71.0	41.8	57.8	91.8	91.9
MIXGB	78.7	92.6	89.5	87.6	91.5	78.3	81.5	38.2	56.9	92.6	93.2
GAIN	66.5	31.5	49.3	40.6	50.5	86.6	52.4	88.8	94.1	60.8	60.0
GAMME (H=200)	**91.6**	83.4	**93.7**	**93.8**	95.3	**91.8**	88.2	**95.8**	95.7	96.0	95.8
GAMME (H=100)	91.5	83.2	94.6	94.5	96.1	91.1	88.4	96.1	95.6	95.8	95.9

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.I.3: Average relative bias in % for 2,000 observations (normal) under MCAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$
BD	-0.01	0.12	-0.01	0.04	0.10	-0.02	0.01	0.04	-0.20
CCA	**0.02**	0.13	-0.11	-0.08	**0.06**	**-0.04**	**0.03**	**0.05**	-0.56
Mean	**-0.04**	**0.12**	**-0.05**	0.18	0.04	-0.07	-0.03	-0.01	**-0.43**
Reg	-6.98	15.84	7.21	0.16	0.05	-10.41	-4.32	-10.35	-10.70
PMM	-3.23	-25.38	-11.38	0.19	0.05	-6.02	-1.14	-8.12	-8.92
BootReg	-4.53	-28.68	-12.89	0.18	0.04	-6.75	-2.81	-6.65	-7.08
CART	-1.66	-22.46	-10.80	-1.41	-1.50	-3.22	-0.72	-4.23	-10.31
RF	-1.70	-29.86	-14.14	-1.79	-1.92	-2.61	-1.10	-3.13	-5.33
EM	-4.50	-28.64	-12.97	0.19	-0.03	-6.70	-2.79	-6.55	-6.88
MissForest	-2.46	14.70	3.39	-2.63	-2.74	-4.12	-1.52	-5.61	-9.83
MIXGB	-1.34	2.48	0.69	-0.50	-0.57	-1.98	-0.75	-6.38	-11.29
GAIN	-2.78	-8.81	-6.96	-3.89	-3.78	-1.26	-2.53	-1.28	-1.72
GAMME (H=200)	-0.12	-2.24	-1.18	**0.00**	0.02	-0.58	0.06	-0.32	-0.88
GAMME (H=100)	-0.13	-2.24	-1.22	0.00	0.01	-0.60	0.06	-0.38	-0.96

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Table 3.I.4: Coverage rate in % for 2,000 observations (normal) under MCAR with interactions

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	$\beta_{3,4}$	$\beta_{7,8}$	β_7	β_8
BD	95.7	94.8	94.3	93.8	96.2	95.3	94.8	95.8	95.1	95.6	95.1
CCA	**94.1**	**95.2**	**94.6**	94.3	95.4	**95.7**	**94.4**	94.5	94.2	**95.7**	94.8
Mean	92.8	97.4	96.5	**93.9**	**96.3**	94.7	95.6	**95.1**	**94.9**	**95.5**	95.5
Reg	3.6	1.1	33.6	85.8	86.4	0.6	20.3	4.7	45.1	88.0	85.3
PMM	66.6	0.0	8.9	97.0	98.3	25.1	90.4	26.7	75.8	97.9	98.1
BootReg	37.4	0.0	3.1	97.1	98.5	16.7	60.0	47.0	84.3	98.1	97.6
CART	86.3	0.0	8.1	92.8	94.7	68.8	91.4	73.8	62.8	97.0	96.6
RF	88.8	0.0	1.0	94.3	**96.3**	81.9	92.0	89.0	90.9	97.9	97.9
EM	28.6	0.0	1.4	93.0	95.3	14.3	49.4	41.1	78.2	94.4	93.5
MissForest	57.6	0.9	72.2	74.6	75.5	40.3	67.8	38.4	49.1	89.2	89.2
MIXGB	81.1	83.3	91.7	88.9	92.1	76.5	84.8	35.6	45.1	91.4	90.4
GAIN	55.7	49.8	42.3	51.4	49.1	87.2	50.4	92.1	93.1	58.5	61.2
GAMME (H=200)	93.9	85.7	93.5	94.9	96.8	93.4	90.6	93.7	**94.9**	97.1	**95.3**
GAMME (H=100)	93.2	85.4	93.4	95.5	97.1	93.7	90.5	95.6	94.4	97.5	95.2

Notes: This table shows the results for 1,000 simulation runs. The best value for each coefficient is in bold and the second best value is underlined. If multiple approaches result in the same value and are best or second best, they are all bold/ underlined. GAMME (H=100) serves as a robustness to GAMME (H=200) and is not considered for the ranking.

Conclusion

Summary

This thesis focuses on various aspects of uncertainty in the application of machine learning. In the first research paper *Quantifying uncertainty of machine learning methods for loss given default* (see Chapter 1) a uncertainty-aware machine learning technique the deep evidential regression is applied to market-based LGDs. This approach divides the uncertainty that is associated with the prediction of the LGDs into aleatoric and epistemic uncertainty. The results document that the proportion of the aleatoric uncertainty is by far larger than the proportion of epistemic uncertainty. This fact is subjected to the second research paper *Non-linearity and the distribution of market-based loss rates* (see Chapter 2) that combines the beta regression and neural networks to model LGDs. The precision parameter of this distributions is closely related to the variance of the modeled distribution. The empirical analysis finds that the vast majority of the feature effects are non-linear for the precision parameter. This flexible approach improves the distributional fit that stresses the importance of adequately modeling the precision parameter.

The third and last research paper *GAMME - Advances in Predictive Mean Matching* (see Chapter 3) focus on a different perspective of uncertainty. Imputation approaches often suffer from two the problems: linearity and single imputation. Despite there are approaches that addresses those problems, most of them lead to invalid statistical inference. The novel approach GAMME that is proposed in the third research paper combines a powerful imputation technique the predictive mean matching with the flexibility of a neural network. In a large simulation study that captures missing completely at random and missing at random GAMME offers the best trade of between unbiased parameters and valid confidence intervals.

Discussion and outlook

The topic of this thesis is of high relevance for risk management and financial institutions. The first research paper as presented in Chapter 1 studies the uncertainty that is associated with modeling LGDs. It is an important step in a deeper understanding of LGDs and the challenges that comes with it. The applied method relies on the deep evidential regression that assumes a normally distributed dependent variable. Despite this is not true for LGDs the deep evidential regression provides competitive results. Future research could extend this framework to different distributional assumptions and still provide the separation of aleatoric and epistemic uncertainty. Furthermore, a comparison regarding different uncertainty estimation techniques like Monte Carlo Dropout by Gal and Ghahramani (2016), ensemble approaches like in Lakshminarayanan et al. (2017) and fully bayesian methods could provide a fruitful area of research to get a deeper understanding of uncertainty. One key result of this paper is that the proportion of aleatoric uncertainty is much larger than the proportion of epistemic uncertainty. These results are based on the dataset that contains market-based LGDs. Another interesting aspect would be the application to workout LGDs.

The market-based LGDs are often assumed to be beta distributed. Therefore, the application of the beta regression is a common choice. The paper in Chapter 2 extends this method by combining the beta regression with a neural network. Future research could apply this flexible approach to workout LGDs. Since those can be negative or greater than one, G-BRANN as presented in Chapter 2 should be extended by using e.g. a four parameter beta distribution (Carnahan, 1989) or using a mixture density network (Bishop, 1994).

Spotting light on uncertainty from different point of view, the last research paper (see Chapter 3) deals with missing data by introducing a novel imputation technique GAMME based on ALE plots and PMM. The conducted simulation study provides initial results for the effectiveness of GAMME under MCAR and MAR. Future research could derive statistical properties of this method and further improve this approach by extending it to mixed type data as well as classification tasks, that could offer a wide range of applications. Another interesting aspect would be the performance under MNAR. Furthermore, extensions regarding the matching can be studied. One possible extension could be to match predicted distributions instead of only means. This could be archived by matching multiple quantiles and use those as a basis to impute missing values.

References

Altman, E. I. and E. A. Kalotay (2014). Ultimate recovery mixtures. *Journal of Banking & Finance 40*, 116–129.

Alvarez-Melis, D. and T. S. Jaakkola (2018). On the robustness of interpretability methods. *arXiv*, 1806.08049v1. Working paper.

Alwosheel, A., S. van Cranenburgh, and C. G. Chorus (2018). Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. *Journal of Choice Modelling 28*, 167–182.

Amini, A., W. Schwarting, A. Soleimany, and D. Rus (2020). Deep evidential regression. In *Advances in Neural Information Processing Systems*, Volume 33, Virtual, pp. 14927–14937. Curran Associates, Inc.

Apicella, A., F. Donnarumma, F. Isgrò, and R. Prevete (2021). A survey on modern trainable activation functions. *Neural Networks 138*, 14–32.

Apley, D. W. and J. Zhu (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology 82*(4), 1059–1086.

Babaei, G., P. Giudici, and E. Raffinetti (2022). Explainable artificial intelligence for crypto asset allocation. *Finance Research Letters 47*(Part B), 102941.

Baesens, B. and K. Smedts (2023). Boosting credit risk models. *The British Accounting Review*, 101241.

Baker, S. R., N. Bloom, and S. J. Davis (2016). Measuring economic policy uncertainty. *The Quarterly Journal of Economics 131*(4), 1593–1636.

Bank of Canada (2018). Financial system survey. Available at `https://www.bankofcanada.ca/2018/11/financial-system-survey-highlights/`. Accessed on 25 July 2024.

Bank of England (2019). Machine learning in uk financial services. Technical report, Bank of England and Financial Conduct Authority. Available at https://www.bankofengland.co.uk/report/2019/machine-learning-in-uk-financial-services. Accessed on 25 July 2024.

Barbaglia, L., S. Manzan, and E. Tosetti (2021). Forecasting loan default in Europe with machine learning. *Journal of Financial Econometrics*. Published online, nbab010.

Barbaglia, L., S. Manzan, and E. Tosetti (2023). Forecasting loan default in Europe with machine learning. *Journal of Financial Econometrics 21*(2), 569–596.

Bartlett, J. W., S. R. Seaman, I. R. White, J. R. Carpenter, and A. D. N. Initiative* (2015). Multiple imputation of covariates by fully conditional specification: accommodating the substantive model. *Statistical Methods in Medical Research 24*(4), 462–487.

Basel Committee on Banking Supervision (2006). International Convergence of Capital Measurement and Capital Standards A Revised Framework Comprehensive Version. Technical report, Bank for International Settlements. Available at https://www.bis.org/publ/bcbs128.pdf. Accessed on 25 July 2024.

Basel Committee on Banking Supervision (2017). Basel III: Finalising post-crisis reforms. Technical report, Bank for International Settlements. Available at https://bis.org/bcbs/publ/d424.pdf. Accessed on 25 July 2024.

Basel Committee on Banking Supervision (2019). High-level summary: BCBS SIG industry workshop on the governance and oversight of artificial intelligence and machine learning in financial services. Available at https://www.bis.org/bcbs/events/191003_sig_tokyo.htm. Accessed on 25 July 2024.

Bastos, J. A. (2010). Forecasting bank loans loss-given-default. *Journal of Banking & Finance 34*(10), 2510–2517.

Bastos, J. A. and S. M. Matos (2022). Explainable models of credit losses. *European Journal of Operational Research 301*(1), 386–394.

Belkin, M., D. Hsu, S. Ma, and S. Mandal (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences 116*(32), 15849–15854.

Bellotti, A., D. Brigo, P. Gambetti, and F. Vrins (2021). Forecasting recovery rates on non-performing loans with machine learning. *International Journal of Forecasting 37*(1), 428–444.

Bellotti, T. and J. Crook (2012). Loss given default models incorporating macroeconomic variables for credit cards. *International Journal of Forecasting 28*(1), 171–182.

Bergstra, J. and Y. Bengio (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research 13*(10), 281–305.

Betz, J., R. Kellner, and D. Rösch (2018). Systematic effects among loss given defaults and their implications on downturn estimation. *European Journal of Operational Research 271*(3), 1113–1144.

Betz, J., M. Nagl, and D. Rösch (2022). Credit line exposure at default modelling using bayesian mixed effect quantile regression. *Journal of the Royal Statistical Society Series A: Statistics in Society 185*(4), 2035–2072.

Bharadiya, J. P., R. K. Thomas, and F. Ahmed (2023). Rise of artificial intelligence in business and industry. *Journal of Engineering Research and Reports 25*(3), 85–103.

Bishop, C. M. (1994). Mixture density networks. Technical report, Aston University, Birmingham, UK. (Unpublished).

Blazek, K., A. van Zwieten, V. Saglimbene, and A. Teixeira-Pinto (2021). A practical guide to multiple imputation of missing data in nephrology. *Kidney International 99*(1), 68–74.

Blundell, C., J. Cornebise, K. Kavukcuoglu, and D. Wierstra (2015). Weight uncertainty in neural network. In *Proceedings of the 32nd International Conference on Machine Learning*, Volume 37, Lille, France, pp. 1613–1622. PMLR.

Boeschoten, L., T. Waal, and J. K. Vermunt (2019). Estimating the number of serious road injuries per vehicle type in the Netherlands by using multiple imputation of latent classes. *Journal of the Royal Statistical Society Series A: Statistics in Society 182*(4), 1463–1486.

Bohr, A. and K. Memarzadeh (2020). Chapter 2 - the rise of artificial intelligence in healthcare applications. In A. Bohr and K. Memarzadeh (Eds.), *Artificial Intelligence in Healthcare*, pp. 25–60. London, UK: Academic Press.

Bryzgalova, S., S. Lerner, M. Lettau, and M. Pelger (2024). Missing financial data. *The Review of Financial Studies*, hhae036.

Burkart, N. and M. F. Huber (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research 70*, 245–317.

Bussmann, N., P. Giudici, D. Marinelli, and J. Papenbrock (2020). Explainable AI in fintech risk management. *Frontiers in Artificial Intelligence 3*, 26.

Bussmann, N., P. Giudici, D. Marinelli, and J. Papenbrock (2021). Explainable machine learning in credit risk management. *Computational Economics 57*, 203–216.

Calabrese, R. (2014). Downturn loss given default: Mixture distribution estimation. *European Journal of Operational Research 237*(1), 271–277.

Calabrese, R. and L. Zanin (2022). Modelling spatial dependence for loss given default in peer-to-peer lending. *Expert Systems with Applications 192*, 116295.

Caprio, G., L. Laeven, and R. Levine (2007). Governance and bank valuation. *Journal of Financial Intermediation 16*(4), 584–617.

Carmona, C. P., C. G. Bueno, A. Toussaint, S. Träger, S. Díaz, M. Moora, A. D. Munson, M. Pärtel, M. Zobel, and R. Tamme (2021). Fine-root traits in the global spectrum of plant form and function. *Nature 597*(7878), 683–687.

Carnahan, J. (1989). Maximum likelihood estimation for the 4-parameter beta distribution. *Communications in Statistics-Simulation and Computation 18*(2), 513–536.

Carpenter, J. R. and M. Smuk (2021). Missing data: A statistical framework for practice. *Biometrical Journal 63*(5), 915–947.

Chen, S., Z. Guo, and X. Zhao (2021). Predicting mortgage early delinquency with machine learning methods. *European Journal of Operational Research 290*(1), 358–372.

Chlebicki, P., Ł. Chrostowski, and M. Beręsewicz (2024). Data integration of non-probability and probability samples with predictive mean matching. *arXiv*, 2403.13750v2. Working paper.

Choudhury, S. J. and N. R. Pal (2019). Imputation of missing data with neural networks for classification. *Knowledge-Based Systems 182*, 104838.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B: Methodological 39*(1), 1–22.

Deng, Y. and T. Lumley (2023). Multiple imputation through XGBoost. *Journal of Computational and Graphical Statistics 33*(2), 352–363.

Der Kiureghian, A. and O. Ditlevsen (2009). Aleatory or epistemic? Does it matter? *Structural Safety 31*(2), 105–112.

Deutsche Bundesbank (2020). The use of artificial intelligence and machine learning in the financial sector. Policy discussion paper. Available at https://www.bundesbank.de/resource/blob/598256/d7d26167bceb18ee7c0c296902e42162/mL/2020-11-policy-dp-aiml-data.pdf. Accessed on 25 July 2024.

Doove, L. L., S. Van Buuren, and E. Dusseldorp (2014). Recursive partitioning for missing data imputation in the presence of interaction effects. *Computational Statistics & Data Analysis 72*, 92–104.

Dumitrescu, E., S. Hué, C. Hurlin, and S. Tokpavi (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research 297*(3), 1178–1192.

Emmanuel, T., T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona (2021). A survey on missing data in machine learning. *Journal of Big Data 8*(140), 1–37.

Enders, C. K. (2022). *Applied Missing Data Analysis* (2 ed.). Methodology in the Social Sciences Series. New York, NY, USA: The Guilford Press.

European Banking Authority (2021). Risk Assessment of the European Banking System. Technical report. Available at https://www.eba.europa.eu/sites/default/files/document_library/Risk%20Analysis%20and%20Data/EU%20Wide%20Transparency%20Exercise/2021/1025102/Risk_Assessment_Report_December_2021.pdf. Accessed on 25 July 2024.

European Banking Authority (2022). Risk Assessment of the European Banking System. Technical report. Available at https://www.eba.europa.eu/sites/default/files/document_library/Risk%20Analysis%20and%20Data/Risk%20Assessment%20Reports/2022/RAR/1045298/Risk%20Assessment%20Report%20December%202022.pdf. Accessed on 25 July 2024.

European Banking Authority (2023a). Machine Learning for IRB Models - Follow-up Report from the Consulation on the Discussion Paper on Machine Learning for IRB Models. Available at https://www.eba.europa.eu/sites/default/files/document_library/Publications/Reports/2023/1061483/Follow-up%20report%20on%20machine%20learning%20for%20IRB%20models.pdf. Accessed on 25 July 2024.

European Banking Authority (2023b). Risk Assessment Report of the European Banking Authority. Technical report. Available at https://www.eba.europa.eu/sites/default/files/2023-12/ed14314d-3194-4808-935b-afc564f748ad/Risk%20Assessment%20Report%20December%202023.pdf. Accessed on 25 July 2024.

Ferrari, S. and F. Cribari-Neto (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics 31*(7), 799–815.

Figlewski, S., H. Frydman, and W. Liang (2012). Modeling the effect of macroeconomic factors on corporate default and credit rating transitions. *International Review of Economics & Finance 21*(1), 87–105.

Fraisse, H. and M. Laporte (2022). Return on investment on artificial intelligence: The case of bank capital requirement. *Journal of Banking & Finance 138*, 106401.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics 29*(5), 1189–1232.

Fritz-Morgenthal, S., B. Hein, and J. Papenbrock (2022). Financial risk management and explainable, trustworthy, responsible ai. *Frontiers in Artificial Intelligence 5*, 779799.

Gal, Y. and Z. Ghahramani (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, Volume 48, New York, NY, USA, pp. 1050–1059. PMLR.

Gambetti, P., G. Gauthier, and F. Vrins (2019). Recovery rates: Uncertainty certainly matters. *Journal of Banking & Finance 106*, 371–383.

Gatti, L. V., L. S. Basso, J. B. Miller, M. Gloor, L. Gatti Domingues, H. L. Cassol, G. Tejada, L. E. Aragão, C. Nobre, W. Peters, and others (2021). Amazonia as a carbon source linked to deforestation and climate change. *Nature 595*(7867), 388–393.

Gawlikowski, J., C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al. (2022). A survey of uncertainty in deep neural networks. *arXiv*, 2107.03342v3. Working paper.

Giudici, P. and E. Raffinetti (2021). Shapley-Lorenz eXplainable artificial intelligence. *Expert Systems with Applications 167*, 114104.

Giudici, P. and E. Raffinetti (2022). Explainable AI methods in cyber risk management. *Quality and Reliability Engineering International 38*(3), 1318–1326.

Glynn, R. J., N. M. Laird, and D. B. Rubin (1986). Selection Modeling Versus Mixture Modeling with Nonignorable Nonresponse. In H. Wainer (Ed.), *Drawing Inferences from Self-Selected Samples*, pp. 115–142. New York, NY, USA: Springer New York.

Goldstein, A., A. Kapelner, J. Bleich, and E. Pitkin (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics 24*(1), 44–65.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press. Available at http://www.deeplearningbook.org.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, Volume 27, Montréal, Canada, pp. 2672–2680. Curran Associates, Inc.

Görgen, K., A. Nazemi, and M. Schienle (2022). Robust knockoffs for controlling false discoveries with an application to bond recovery rates. *arXiv*, 2206.06026v1. Working paper.

Gu, S., B. Kelly, and D. Xiu (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies 33*(5), 2223–2273.

Gunnarsson, B. R., S. vanden Broucke, B. Baesens, M. Óskarsdóttir, and W. Lemahieu (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research 295*(1), 292–305.

Gürtler, M. and M. Zöllner (2023). Heterogeneities among credit risk parameter distributions: the modality defines the best estimation method. *OR Spectrum 45*(1), 251–287.

Hammon, A. (2022). Multiple imputation of ordinal missing not at random data. *AStA Advances in Statistical Analysis 107*(4), 671–692.

Hammon, A. and S. Zinn (2020). Multiple imputation of binary multilevel missing not at random data. *Journal of the Royal Statistical Society Series C: Applied Statistics 69*(3), 547–564.

Han, J. and S. Kang (2022). Dynamic imputation for improved training of neural network with missing values. *Expert Systems with Applications 194*, 116508.

Havrylenko, Y. and J. Heger (2024). Detection of interacting variables for generalized linear models via neural networks. *European Actuarial Journal 14*, 551–580.

He, K., X. Zhang, S. Ren, and J. Sun (2015). Delving deep into rectifiers: Surpassing Human-Level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 1026–1034. IEEE.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770–778. IEEE.

Heckman, J. J. (1976). The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models. In S. V. Berg (Ed.), *Annals of Economic and Social Measurement, Volume 5, Number 4*, pp. 475–492. NBER.

Hornik, K., M. Stinchcombe, and H. White (1989). Multilayer feedforward networks are universal approximators. *Neural Networks 2*(5), 359–366.

Hwang, R.-C. and C.-K. Chu (2018). A logistic regression point of view toward loss given default distribution estimation. *Quantitative Finance 18*(3), 419–435.

Hwang, R.-C., C.-K. Chu, and K. Yu (2020). Predicting LGD distributions with mixed continuous and discrete ordinal outcomes. *International Journal of Forecasting 36*(3), 1003–1022.

Jankowitsch, R., F. Nagler, and M. G. Subrahmanyam (2014). The determinants of recovery rates in the US corporate bond market. *Journal of Financial Economics 114*(1), 155–177.

Jurado, K., S. C. Ludvigson, and S. Ng (2015). Measuring uncertainty. *American Economic Review 105*(3), 1177–1216.

Kalotay, E. A. and E. I. Altman (2017). Intertemporal forecasts of defaulted bond recoveries and portfolio losses. *Review of Finance 21*(1), 433–463.

Kaposty, F., J. Kriebel, and M. Löderbusch (2020). Predicting loss given default in leasing: A closer look at models and variable selection. *International Journal of Forecasting 36*(2), 248–266.

Kellner, R., M. Nagl, and D. Rösch (2022). Opening the black box – Quantile neural networks for loss given default prediction. *Journal of Banking & Finance 134*, 106334.

Kim, S., C. A. Sugar, and T. R. Belin (2015). Evaluating model-based imputation methods for missing covariates in regression models with interactions. *Statistics in Medicine 34*(11), 1876–1888.

King, G., J. Honaker, A. Joseph, and K. Scheve (2001). Analyzing incomplete political science data: An alternative algorithm for multiple imputation. *American Political Science Review 95*(1), 49–69.

Kingma, D. P. and J. Ba (2017). Adam: A method for stochastic optimization. *arXiv*, 1412.6980v9. Working paper.

Kleinke, K. (2017). Multiple imputation under violated distributional assumptions: A systematic evaluation of the assumed robustness of predictive mean matching. *Journal of Educational and Behavioral Statistics 42*(4), 371–404.

Knell, G., M. C. Robertson, E. E. Dooley, K. Burford, and K. S. Mendez (2020). Health behavior changes during COVID-19 pandemic and subsequent "stay-at-home" orders. *International Journal of Environmental Research and Public Health 17*(17), 6268.

Kniss, J. M. (2008). Managing uncertainty in visualization and analysis of medical data. In *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Paris, France, pp. 832–835. IEEE.

Krueger, D., C.-W. Huang, R. Islam, R. Turner, A. Lacoste, and A. Courville (2017). Bayesian hypernetworks. *arXiv*, 1710.04759v1. Working paper.

Krüger, S. and D. Rösch (2017). Downturn LGD modeling using quantile regression. *Journal of Banking & Finance 79*, 42–56.

Kvamme, H., N. Sellereite, K. Aas, and S. Sjursen (2018). Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications 102*, 207–217.

Lakshminarayanan, B., A. Pritzel, and C. Blundell (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, Volume 30, Long Beach, CA, USA, pp. 6402–6413. Curran Associates, Inc.

Li, S. C.-X., B. Jiang, and B. Marlin (2019). MisGAN: Learning from incomplete data with generative adversarial networks. *arXiv*, 1902.09599v1. Working paper.

Li, Y. and W. Chen (2021). Entropy method of constructing a combined model for improving loan default prediction: A case study in China. *Journal of the Operational Research Society 72*(5), 1099–1109.

Liang, Z. and Q. Wang (2023). A robust model averaging approach for partially linear models with responses missing at random. *Scandinavian Journal of Statistics 50*(4), 1933–1952.

Lin, W.-C. and C.-F. Tsai (2020). Missing value imputation: A review and analysis of the literature (2006–2017). *Artificial Intelligence Review 53*, 1487–1509.

Little, R. J., R. D'Agostino, M. L. Cohen, K. Dickersin, S. S. Emerson, J. T. Farrar, C. Frangakis, J. W. Hogan, G. Molenberghs, S. A. Murphy, and others (2012). The prevention and treatment of missing data in clinical trials. *New England Journal of Medicine 367*(14), 1355–1360.

Little, R. J. A. (1992). Regression with missing X's: a review. *Journal of the American Statistical Association 87*(420), 1227–1237.

Little, R. J. A. (1993). Pattern-mixture models for multivariate incomplete data. *Journal of the American Statistical Association 88*(421), 125–134.

Little, R. J. A. and D. B. Rubin (2019). *Statistical Analysis with Missing Data* (3 ed.). Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons.

Loterman, G., I. Brown, D. Martens, C. Mues, and B. Baesens (2012). Benchmarking regression algorithms for loss given default modeling. *International Journal of Forecasting 28*(1), 161–170.

Ludvigson, S. C., S. Ma, and S. Ng (2021). Uncertainty and business cycles: Exogenous impulse or endogenous response? *American Economic Journal: Macroeconomics 13*(4), 369–410.

Lundberg, S. M. and S.-I. Lee. A unified approach to interpreting model predictions. *arXiv*, 1705.07874v2. Working paper.

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

Matuszyk, A., C. Mues, and L. C. Thomas (2010). Modelling LGD for unsecured personal loans: Decision tree approach. *Journal of the Operational Research Society 61*(3), 393–398.

Meinert, N., J. Gawlikowski, and A. Lavin (2022). The unreasonable effectiveness of deep evidential regression. *arXiv*, 2205.10060v1. Working paper.

Meinert, N. and A. Lavin (2022). Multivariate deep evidential regression. *arXiv*, 2104.06135v4. Working paper.

Miller, P. and E. Töws (2018). Loss given default adjusted workout processes for leases. *Journal of Banking & Finance 91*, 189–201.

Mobiny, A., P. Yuan, S. K. Moulik, N. Garg, C. C. Wu, and H. Van Nguyen (2021). Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Scientific Reports 11*, 5458.

Molnar, C. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable* (2 ed.). Available at `https://christophm.github.io/interpretable-ml-book`.

Murray, J. S. (2018). Multiple Imputation: A review of practical and theoretical findings. *Statistical Science 33*(2), 142–159.

Nagl, M. (2023). Does non-linearity in risk premiums vary over time? *SSRN*, 4638168. Working paper.

Nagl, M., M. Nagl, and D. Rösch (2022). Quantifying uncertainty of machine learning methods for loss given default. *Frontiers in Applied Mathematics and Statistics 8*, 1076083.

Nazemi, A., F. Baumann, and F. J. Fabozzi (2021). Intertemporal defaulted bond recoveries prediction via machine learning. *European Journal of Operational Research*. Published online.

Nazemi, A., F. Fatemi Pour, K. Heidenreich, and F. J. Fabozzi (2017). Fuzzy decision fusion approach for loss-given-default modeling. *European Journal of Operational Research 262*(2), 780–791.

Nazemi, A., K. Heidenreich, and F. J. Fabozzi (2018). Improving corporate bond recovery rate prediction using multi-factor support vector regressions. *European Journal of Operational Research 271*(2), 664–675.

Nusinovici, S., Y. C. Tham, M. Y. C. Yan, D. S. W. Ting, J. Li, C. Sabanayagam, T. Y. Wong, and C.-Y. Cheng (2020). Logistic regression was as good as machine learning for predicting major chronic diseases. *Journal of Clinical Epidemiology 122*, 56–69.

Olson, L. M., M. Qi, X. Zhang, and X. Zhao (2021). Machine learning loss given default for corporate debt. *Journal of Empirical Finance 64*, 144–159.

Petropoulos, A., V. Siakoulis, E. Stavroulakis, and N. E. Vlachogiannakis (2020). Predicting bank insolvencies using machine learning techniques. *International Journal of Forecasting 36*(3), 1092–1113.

Qi, M. and X. Yang (2009). Loss given default of high loan-to-value residential mortgages. *Journal of Banking & Finance 33*(5), 788–799.

Qi, M. and X. Zhao (2011). Comparison of modeling methods for loss given default. *Journal of Banking & Finance 35*(11), 2842–2855.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference* *on Knowledge Discovery and Data Mining*, KDD '16, New York, NY, USA, pp. 1135–1144. Association for Computing Machinery.

Rockel, T. (2022). missMethods: Methods for Missing Data. *R package version 0.4.0*. Available at https://CRAN.R-project.org/package=missMethods.

Rubin, D. B. (1976). Inference and missing data. *Biometrika 63*(3), 581–592.

Rubin, D. B. (2003). Nested multiple imputation of NMES via partially incompatible MCMC. *Statistica Neerlandica 57*(1), 3–18.

Rubin, D. B. (2004). The design of a general and flexible system for handling nonresponse in sample surveys. *The American Statistician 58*(4), 298–302.

Rässler, S. (2003). A non-iterative bayesian approach to statistical matching. *Statistica Neerlandica 57*(1), 58–74.

Sadhwani, A., K. Giesecke, and J. Sirignano (2021). Deep learning for mortgage risk. *Journal of Financial Econometrics 19*(2), 313–368.

Satopaa, V., J. Albrecht, D. Irwin, and B. Raghavan (2011). Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, Minneapolis, MN, USA, pp. 166–171. IEEE.

Schafer, J. L. and J. W. Graham (2002). Missing data: Our view of the state of the art. *Psychological methods 7*(2), 147–177.

Seaman, S. R., J. W. Bartlett, and I. R. White (2012). Multiple imputation of missing covariates with non-linear effects and interactions: an evaluation of statistical methods. *BMC Medical Research Methodology 12*(46), 1–13.

SIFMA Research (2022). SIFMA Research Quarterly -1Q22 US Fixed Income Markets - Outstanding. Technical report, Securities Industry and Financial Markets Association (SIFMA). Available at https://www.sifma.org/wp-content/uploads/2022/03/US-Research-Quarterly-Fixed-Income-Outstanding-2022-06-22-SIFMA.pdf. Accessed on 25 July 2024.

Sigrist, F. and C. Hirnschall (2019). Grabit: Gradient tree-boosted Tobit models for default prediction. *Journal of Banking & Finance 102*, 177–192.

Simas, A. B., W. Barreto-Souza, and A. V. Rocha (2010). Improved estimators for a general class of beta regression models. *Computational Statistics & Data Analysis 54*(2), 348–366.

Smithson, M. and J. Verkuilen (2006). A better lemon squeezer? Maximum-likelihood regression with beta-distributed dependent variables. *Psychological Methods 11*(1), 54–71.

Sopitpongstorn, N., P. Silvapulle, J. Gao, and J.-P. Fenech (2021). Local logit regression for loan recovery rate. *Journal of Banking & Finance 126*, 106093.

Spilimbergo, A. (2009). Democracy and foreign education. *American Economic Review 99*(1), 528–543.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*(56), 1929–1958.

Starosta, W. (2021). Loss given default decomposition using mixture distributions of in-default events. *European Journal of Operational Research 292*(3), 1187–1199.

Stekhoven, D. J. and P. Bühlmann (2012). MissForest—Non-parametric missing value imputation for mixed-type data. *Bioinformatics 28*(1), 112–118.

Su, Z., Q. Guo, and H.-T. Lee (2023). Retraction notice to "Green finance policy and enterprise energy consumption intensity: Evidence from a quasi-natural experiment in China" [Energy Economics 115 (2022) 106374]. *Energy Economics 128*, 107137.

Sun, Y., J. Li, Y. Xu, T. Zhang, and X. Wang (2023). Deep learning versus conventional methods for missing data imputation: A review and comparative study. *Expert Systems with Applications 227*, 120201.

Tang, F. and H. Ishwaran (2017). Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal 10*(6), 363–377.

Tobback, E., D. Martens, T. V. Gestel, and B. Baesens (2014). Forecasting loss given default models: Impact of account characteristics and the macroeconomic state. *Journal of the Operational Research Society 65*(3), 376–392.

Tomarchio, S. D. and A. Punzo (2019). Modelling the loss given default distribution via a family of zero-and-one inflated mixture models. *Journal of the Royal Statistical Society Series A: Statistics in Society 182*(4), 1247–1266.

Tong, E. N. C., C. Mues, and L. Thomas (2013). A zero-adjusted gamma model for mortgage loan loss given default. *International Journal of Forecasting 29*(4), 548–562.

Tyralis, H. and G. Papacharalampous (2024). A review of predictive uncertainty estimation with machine learning. *Artificial Intelligence Review 57*, 94.

Valdenegro-Toro, M. (2019). Deep sub-ensembles for fast uncertainty estimation in image classification. *arXiv*, 1910.08168v2. Working paper.

Van Buuren, S. (2018). *Flexible Imputation of Missing Data* (2 ed.). CRC Press. Available at `https://stefvanbuuren.name/fimd/`.

Van Buuren, S., J. P. Brand, C. G. Groothuis-Oudshoorn, and D. B. Rubin (2006). Fully conditional specification in multivariate imputation. *Journal of Statistical Computation and Simulation 76*(12), 1049–1064.

Van Buuren, S. and K. Groothuis-Oudshoorn (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software 45*(3), 1–67.

Vateekul, P. and K. Sarinnapakorn (2009). Tree-based approach to missing data imputation. In *2009 IEEE International Conference on Data Mining Workshops*, Miami, FL, USA, pp. 70–75. IEEE.

Vink, G., L. E. Frank, J. Pannekoek, and S. Van Buuren (2014). Predictive mean matching imputation of semicontinuous variables. *Statistica Neerlandica 68*(1), 61–90.

Visani, G., E. Bagli, F. Chesani, A. Poluzzi, and D. Capuzzo (2022). Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models. *Journal of the Operational Research Society 73*(1), 91–101.

Von Hippel, P. T. (2009). How to impute interactions, squares, and other transformed variables. *Sociological Methodology 39*(1), 265–291.

Wen, Y., D. Tran, and J. Ba (2020). Batchensemble: An alternative approach to efficient ensemble and lifelong learning. *arXiv*, 2002.06715v2. Working paper.

Wooldridge, J. M. (2020). *Introductory Econometrics: A Modern Approach* (7 ed.). Victoria, Australia: Cengage Learning.

Xia, Y., J. Zhao, L. He, Y. Li, and X. Yang (2021). Forecasting loss given default for peer-to-peer loans via heterogeneous stacking ensemble approach. *International Journal of Forecasting 37*(4), 1590–1613.

Yang, S. and J. K. Kim (2020). Asymptotic theory and inference of predictive mean matching imputation using a superpopulation model framework. *Scandinavian Journal of Statistics 47*(3), 839–861.

Yang, Z., Y. Yu, C. You, J. Steinhardt, and Y. Ma (2020). Rethinking Bias-Variance Trade-off for generalization of neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, Virtual, pp. 10767–10777. PMLR.

Yoon, J., J. Jordon, and M. van der Schaar (2018). GAIN: Missing Data Imputation using Generative Adversarial Nets. In *Proceedings of the 35th International Conference on Machine Learning*, Volume 80 of *Proceedings of Machine Learning Research*, Stockholm, Sweden, pp. 5689–5698. PMLR.