KU LEUVEN

FACULTY OF ECONOMICS
AND BUSINESS

Insurance analytics

Leveraging hierarchical and network data in predictive modeling

Dissertation presented to
obtain the degree of
Doctor in Business Economics

by

Bavo De Cock CAMPO

Number 857 2024

In loving memory of my mother, professor dr. Katia Campo, who helped me in ways no other could. And to my father, Ing. Luc De Cock, who has always been a pillar of support to our family. To ir. Natasha, ir. Koen, arch. Laure-Anne, MSc Wim and dr. Hector.

Committee

Advisor:

Prof. Dr. Katrien Antonio	*KU Leuven and University of Amsterdam*

Chair:

Prof. Dr. Els Breugelmans	*KU Leuven*

Members:

Prof. Dr. Andrés M. Villegas	*University of New South Wales*
Prof. Dr. Bart Baesens	*KU Leuven and University of Southampton*
Prof. Dr. Gee Y. Lee	*Michigan State University*
Prof. Dr. Jan Dhaene	*KU Leuven*

Acknowledgements

No one who achieves success does so without acknowledging the help of others. The wise and confident acknowledge this help with gratitude.

-

Alfred North Whitehead

With this quote, I am not trying to portray myself as wise or confident. Rather, I want to acknowledge and express the deepest gratitude to all people who supported me and continue to support me. I would not be where I am today without the help of many people. In particular, my mother and father. The majority of the acknowledgements start by going back to the start of the PhD. For me, however, it feels as if I have to go back even earlier and it is for this reason that this section is longer than most. Compared to the average student, my road towards a PhD was a bit atypical and longer. As such, it feels like a beautiful end to a long journey and as the beginning of a whole new journey.

Before going all the way back, let me start by thanking the one who made this journey possible and who guided me over the past four years. Professor *Katrien Antonio*, I am incredibly grateful that you gave me the opportunity and convinced me to do a PhD. I would not have made any other choice knowing what I know today. I am aware that you do not like long-winded texts, but just for once this will be a lengthy one. You are a role model to me, as well as to every other (PhD) student that you come into contact with. You posses an incredible wide array of diverse skills and the university can consider itself lucky to have a person, leader like you. You have multitaskers, supertaskers and then you have Katrien Antonio. Over the past four years, I have learned a lot from you and you have helped me grow in many ways. Thanks to you, I gained a whole new skill set and took the

skills I already had to another level.

Before extending my thanks to the other professors, I want to express the deepest gratitude to my mother, professor dr. *Katia Campo*, and father, Ing. *Luc De Cock*. Mom, your continued support and love have been invaluable in my life. Next to growing and developing your career, you took time to spend with your family and helping each one of us. I still vividly remember all of our vacations together, our family meals as well as every time when you helped me with statistics. You have and always will be my main role model. Dad, just as Mom, you have always been there for me and supported me. Even more, you are the pillar of support to our family. Not only did you support Mom throughout her whole career, but you also worked incredibly hard every day so you could take care of us and were always there for all of your children. Even after mom passed away, you continued to do so and kept providing us with a warm home that we can come back to, our safe base. I feel lucky to have (had) parents who are so caring and supportive.

I am also profoundly grateful to my PhD committee members. Professor *Jan Dhaene*, professor *Bart Baesens*, thank you for all the effort you put in reading my first papers, attending my doctoral seminars and for all the helpful feedback. Jan, I still remember you telling me about the progress I made since the first doctoral seminar and how much this motivated me to continue. Bart, thank you for your guidance on how to stay connected with academia and the kind words about my mother. Professor *Gee Y. Lee*, professor *Andrés M. Villegas*, thank you for agreeing to be part of my committee, to go through my PhD thesis and for giving me insightful feedback which helped me improve my thesis. Thank you, for the positive communication and for being equally enthusiastic about the research as I was.

In addition to the professors of insurance and data analytics, I would also like to thank everyone of the *Marketing* department and professor *Bart Van Looy* from the MSI department. Bart, you guided me during my first statistical research projects as a psychology student. Together with my mother, you guided and set me towards the path that ultimately led to a PhD. Professor *Els Breugelmans*, you were there, as one of my mother's closest colleagues, for our family when our mother passed away. You showed us that you were more than just colleagues. This is also the feeling I got when I attended the Katia Campo Marketing Retailing symposium. It felt as if it was one big, close family of researchers. I loved how passionate each one of you were, just like my mom. Thank you and thank you for agreeing to be chair of my committee. While my mom may not be here today, she still lives in all of us and I am happy that, in this way, it feels as if she is here. Professor *Lien Lamey*,

you were the first colleague of my mother that I met after having started my PhD at the FEB. Thanks to you, I got to know another side of my mom a bit better and I am truly grateful for this. It felt good to be able to talk about my mom and to hear how she was as a colleague, as a researcher and as a mentor.

Before I started my PhD at the FEB, I worked for four years as a biostatistician in the *IOTA* team and I am forever grateful to everyone of the IOTA team. You gave me the opportunity to start in research and provided me with a solid basis of (statistical) research. The passion that each on of you had for research quickly spread to me. You were more than just colleagues and I still think fondly of all the moments we worked together. Professor *Ben Van Calster*, thank you for training me and helping me become a biostatistician. Everything you taught me (going from multiple testing to calibration), I am still using today and informing colleagues about. Professor *Dirk Timmerman*, thank you for involving me in and allowing me to be part of so many different research projects. You not only made me learn quick and fast, but also helped in building an impressive publication list. Professor *Tom Bourne*, one of my first research projects was led by you and you even made me second author of one of the papers that was important to you. Thank you and thank you for lightening up the marathon meetings with your funny jokes and remarks. Professor *Wouter Froyman*, professor *Chiara Landolfo*, we spent a lot of time together at the office. You took working hard to the next level and taught me that you can also have fun doing so. I enjoyed all the late nights that we stayed at the office, because this was way better than sitting at home alone. Professor *Laure Wynants*, thank you for helping me, together with Ben to become a biostatistician. You helped me gain insight into (the fundamentals of) statistics. I also loved coming to the office and telling about the funny things that happened during the weekend. Dr. *Evangelia Christodoulou*, you became a close friend of mine and your friendship means a lot to me. I can talk to you about everything and I love it that Natasha became a friend of yours as well. Thank you for everything. Dr. *Ruben Heremans*, another colleague that became a good friend. I appreciate how we can both have fun and be serious the next moment, even after having not seen each other for a long time. Dr. *Dries Ceulemans*, you made me laugh so many times and it was fun to have someone else who used `R` to let it do the work for you. Jolien, you took over my role within the IOTA team and I am glad that also you feel at home with them. Even though we may not see each other that often, you know that it's friend for life when you can come together and it feels as if you last met yesterday. Thank you everyone. I hope that, one day, we can meet and work together again.

Additionally, I want to thank everyone of my family. Sadly, over the past few years, we have lost more than just a few of our loved ones. I may not be able to thank you in person, but I at least want to do so by writing it down here. *Little grandma*, just as mom, you loved us as no one else and were always there for us. Not only for us, but for every single person that you came into contact with. *Big grandpa*, we may have gotten out of touch and things were more complicated compared to when we were little, but I hope you knew that we still loved you. *Hélène*, our grandma in Limburg, who provided us with as much 'Limburgse vlaai' as love and made sure that we never were short of either. *Bompa Crauwels*, just as *Nonkel Paul*, you were a special and beloved family member that always made sure that we were entertained. We may have not shared that many moments together, but I still vividly remember each one of them.

Even though our family has seen its share of grief, it also became closer and we continue to share so many beautiful and funny moments together. Life may not always be easy, but it would be infinitely more difficult without my girlfriend Natasha. Моя любовь, words cannot express how happy I am that you are part of my life. You are the true definition of a partner. Someone who is always there for me, someone who supports me in both good and bad times, someone who is part of me. I could not have done it without you and your support. Спасибо за всё! Я очень счастлив что у меня есть ты! Я тебя сильно люблю! *Koen, Wim*, I am incredibly grateful to have you as brothers. All the weekends together kept me sane, as well as all the (pointless) discussions. Even more, both of you helped me when I had to study mathematics from scratch again. It were a rough couple of years, but we sticked together and grew even closer. Koen, I love how you have so much knowledge on a lot of different topics and how you are able to explain even the most complicated subjects in an understandable manner. You are one of the smartest and kind persons I know. Wim, otoutu, you as well. I am always surprised by how much you still remember of physics, chemistry and mathematics (because I basically forgot everything about chemistry and physics) and I love how passionate you are when we are discussing all sorts of topics. Your quote "mathematics is a language" will always stick with me. You may not realize it, but you are one of the hardest workers I know. Even more impressive is that, at the same time, you are able to keep close contact with your friends and organize events. *Laure-Anne*, I am incredibly happy to have you as my sister-in-law. You helped support us all and are one of the reasons why we, as a family, became so close over the past years. I love how you take initiative in organizing our get-togethers, how organized you are

and how you are able to withstand the many (pointless) discussions I have with my brothers and father.

Nonkel Kris, tante Ingrid, I am immensely grateful to have you as part of our family. You made me realize that, in the end, your family is always there for you and that, together, you manage to survive some of the hardships of life. The well-timed humour also helped to make everything a bit easier. This seems to be a family trait and eccentricity of the Campo family. *Bompa Hugo*, my other role model, I love how you teach each one of us to go through life. Your never-ending optimism, (inappropriate and) funny remarks and your endless supply of energy are admirable. *Lien, professor Jonas, Stijn, Dieter, Nikole, everyone of my family*, I will thank all of you in person, because this text is becoming way too long.

Next up, my *AFI* friends and colleagues. Unfortunately, I did not get to know everyone as much as I wanted to. Partially because I was going through a rough time myself (in addition to spending nearly two years in lock down). Dr. *Eva*, we started at the same time and I am so glad that both of us made it to the end. Over the past four years, we had a lot of fun moments together (I never had so much fun doing karaoke), but also difficult ones (well, remember COVID). You were one of the colleagues that I could ask anything to, be it professional or personal. Thank you for everything! *Jens*, in the middle of one of the many lockdowns, you joined our team. I learned a lot from you and will always remember your advice to "not be so stressed". We had a lot of laughs together and it was awesome that you attended all parties (even though you don't drink and must have seen some funny things). *Freek*, I was happy that someone my age joined which made me feel less old. In addition, seeing you take initiative and network motivated me to do the same. Next to working hard, you also love to party hard which led to some unforgettable nights with the whole insurance and AFI group. Even more amazing is that you are able to present flawlessly the day after. Never have I been so jealous. *Paul*, unfortunately we were only colleagues for one year, but this was more than enough to make a lasting impression. You take being chill to a whole other level. Just as Freek, you quickly took initiative and organized the trip to the best museum ever. Thanks to you, everything went smooth as butter. The team would not have been the same without all of you.

I am happy that I got to meet a lot of other colleagues of the AFI department too. *Churui*, both me and Natasha had a lot of good talks with you as well as fun moments. Wentao, Biwen, Athibav, the same goes for you. Dr. *Álvaro* (Gallegos), of all people who know how to party, you know it best. I am looking forward to our

future parties together. Next to being a great party companion, you have this gift to really tune in when someone is talking. You, together with Sonia, became close friends of both me and Natasha and we are incredibly happy to have you as our friends. You introduced us to so many other awesome people as well. Amelia, Jeff, dr. Álvaro (Gutiérrez Vargasto), Valeria, just to name a few. Professor *Лиза*, я очень счастлив, что мы встретились и я очень рад, что ты мой и Наташин друг. Ты наш любимый гость! I also admire your bravery and determination. Literally nothing can stop you, not even harsh living conditions. *Lars*, unfortunately the COVID lockdowns prevented us from partying the first two years. Nonetheless, the few parties we had were always fun. *Maxim, Carola*, thank you for all the awesome moments together and the unforgettable night at the fuse. *Lotte, Mieke*, thank you for organizing the first AFI parties after the lockdowns. It must not have been easy with all the restrictive and changeable measures, but I am glad that you took the initiative.

(We're almost there, just a little bit more.)

We can, of course, not forgot the *AFI seniors*. Of the seniors, professor *Dieter* makes sure to keep the AFI spirit alive and guides all new colleagues or lets them feel welcomed with his well-timed sarcastic remarks. *José* is another senior that started before my time, but someone that I got to know better over the past four years. I admire how you dare to take calculated risks as an entrepreneur and, at the same time, keep your calm composure. Dr. *Roel*, you are as good at partying as you are at doing research. You show us that you can excel at both at the same time. Dr. *Sander*, dr. *Jonas*, thank you for welcoming me when I just started and all the good talks.

Some other colleagues I absolutely cannot forget are *Monique Smets* and *Natacha Janssens*. Both of you are vital in making sure that everything runs (smoothly) in the department. Thank you for all the help, both professionally and personally. Sometimes I just needed to vent (about the efficiency of Belgium and what not) and the fact that I could do this when I came by, meant a lot to me.

Further, I want to thank all of my friends who have supported me throughout everything. *Vincent, Inge, Jonas, Cédric, Rekha, Jordi, Daniella, everyone else from Wuustwezel and Sint-Job*, thank you for keeping me sane throughout this whole period. *Jan, Andries*, thank you for the unforgettable moments at Heidestraat 1 and for the many dinners, during which we discussed and laughed about all sorts of things. *Joris, Karen, Nico, Davide, Giulia*, my fellow adulty adults, I am immensely

thankful to have you as my friends. I can literally discuss everything with you and I often have the impression that you know me better than I know myself. Together with Natasha, you make sure that I don't overdo it and burn the candle at both ends.

Next to my family and friends, there are some other people who played a pivotal role in my life. *Jan Van Doren*, during high school you were (literally) the only teacher that believed in me and that kept motivating me. When I had the worst exam results ever in history, according to M.V.D., you took me apart and had a heart-to-heart conversation with me. You told me that you believed in me, but that I had to start taking things seriously. After this, I met more people just like you. You, together with my family, made sure that I kept going. Everyone from psychology, statistics, insurance, thank you for believing in me.

Last, but not least, I want to give a special thank you to *Willem*. Willem guided me when I was working together with the insurance company and helped form the basis of the two first research papers. You showed me that not only researchers, but also practitioners are passionate about their work. Thank you for all the insights, great ideas and discussions. In a way, this thesis is also a tribute to you and the knowledge that you passed on to me.

Kessel-Lo, January 2024,
Bavo Ingrid De Cock Campo.

Contents

Chapter 1

Introduction

Insurance plays a pivotal role in society, by providing risk management strategies to safeguard entities against uncertain financial losses. Property and casualty (P&C) insurance, which covers one's belongings and liability, allows individuals and businesses to mitigate the financial burden of unexpected events by transferring the risk to an insurance company. For example, a company can mitigate the financial consequences arising from job-related injuries by taking out workers' compensation insurance for its employees. To cover the costs, insurance companies pool together a large number of entities exposed to similar risks.

Policyholders transfer the risk to the insurance company by signing an insurance contract, which specifies the precise criteria that activate the financial compensation. As part of the contract, policyholders need to pay a fixed premium at the beginning of the coverage period. As such, policyholders are relieved from the financial burden associated with the event as outlined in the contract. In return for the premium, the insurer commits to reimbursing future losses incurred by the policyholders. However, at the inception of the contract, the precise cost for the insurer is unknown and this is referred to as the *inverse production cycle*. Consequently, one of the essential aspects of insurance is to accurately quantify the risks and hereto related costs.

The portfolio of an insurance company typically consists of several diverse risk profiles. The likelihood of an event occurring varies across entities, leading to adjustments in premiums that mirror the heterogeneity of the covered risk. Hereto, insurers rely on observable characteristics to categorize policyholders with a comparable risk profile into tariff classes. To create and define these risk segments, we rely on different types of risk factors. Subsequently, using predictive modeling

techniques, we estimate the loss cost as a function of the observed risk characteristics for each segment.

Technological innovations have facilitated the process of gathering and storing copious amounts of data. As a consequence, actuaries typically have several risk factors at their disposal when constructing the pricing model. Furthermore, the insurer's database typically encompasses a wide range of covariates, including nominal, ordinal, numeric, and spatial variables. For example, to determine the premium for a motor insurance cover, the pricing model will commonly include the type of fuel of the vehicle, the level occupied in the bonus-malus scale, the policyholder's age and residence area (see, for example, Henckaerts et al. (2018)).

To develop an insurance pricing model, actuaries rely on statistical or machine learning methods. Both types of modeling techniques are well-equipped to handle different types of risk factors. In general, these risk factors are organized and stored in a tabular structure within a historical database. Notwithstanding, certain types of variables present a challenge when incorporating them into a predictive model using default methods. For instance, within a workers' compensation insurance product, we frequently encounter an industrial classification system to categorize companies based on their economic activity. Given the extensive array of diverse economic activities, the variable encoding this information consists of an exceedingly large number of categories. The default method to handle nominal variables is through dummy encoding. However, within this particular context, generalized linear models (GLMs) may yield unreliable parameter estimates and machine learning methods may become computationally intractable. Within machine learning, this type of risk factor is referred to as a high-cardinality attribute. Following Ohlsson (2008), we refer hereto as a multi-level factor (MLF). Further, in certain cases, we encounter MLFs with a hierarchical structure. Such as the said industrial classification systems that typically adopt a hierarchical design to classify companies.

Social network data is another type of information that creates difficulties when attempting to integrate it in its original form, due to the tabular structure of the insurer's database. Within an insurance context, we employ this type of data to represent the relationship between claims and the parties involved. Using network data, insurers can unveil connections among individuals that remain hidden from conventional data sources. Consequently, it can be especially valuable in detecting insurance fraud, for example. Within the actuarial literature, there are several papers that demonstrate the importance of social network analytics to identify fraudsters (Van Vlasselaer et al., 2016; Óskarsdóttir et al., 2022; Tumminello et al.,

2023). However, research on employing social network analytics to detect insurance fraud is hindered by the lack of publicly available data.

This thesis consists of three chapters. In the first two chapters we focus on the use of hierarchical MLFs within insurance pricing models. We investigate existing techniques for integrating a hierarchical MLF into a predictive model. Further, we develop a data-driven procedure to construct an insurance pricing model when both contract-specific and hierarchically structured risk factors are available. In addition, we present a data-driven algorithm to reduce a hierarchical MLF to its essence. Using the algorithm, we group similar categories at every level in the hierarchy. As such, we expand the actuary's toolkit to handle hierarchical MLFs. The third chapter focuses on social network analytics in insurance fraud detection. We present a simulation engine to generate synthetic insurance fraud network data. Hereby, we provide a powerful and flexible toolbox that is able to simulate a wide variety of scenarios, tailored to different research purposes.

1.1 Multi-level factors

The first two chapters focus on handling hierarchical MLFs. One strategy to incorporate hierarchical MLFs into our predictive model is by introducing random effects. In Chapter 2, we provide a comprehensive overview of the random effects approach. We discuss which estimation methods are available and perform an in-depth comparison of the different estimation techniques. Furthermore, we develop a data-driven workflow to construct an insurance pricing model when both hierarchically structured risk factors and contract-specific risk factors are available. In addition, we also examine the effect of the distributional assumption on the response. The random effects approach enables us to efficiently compute and analyze the effect of the hierarchical MLF categories. Moreover, this approach delivers an insurance pricing model that is easy to implement and understand.

In specific instances, however, the random effects approach may not be feasible or appropriate. This can occur when certain categories have too few observations or when there are categories with an identical risk profile. For these situations, an alternative solution is needed, and this is what we present in Chapter 3. In this chapter, we propose an algorithm to simplify the hierarchical MLF to its fundamental components. Hereto, it relies on a combination of feature engineering, clustering techniques and cluster evaluation criteria. Our algorithm considerably reduces the number of categories and creates a grouping that generalizes to out-of-sample data.

Furthermore, when used as a risk factor in a linear mixed model, the clustering solution allows for a better differentiation between high-risk and low-risk companies.

1.2 Social network data

In the third and final chapter we focus on insurance fraud detection. Here, we present a simulation engine to address the lack of publicly available insurance fraud network data. The engine enables researchers and practitioners to generate diverse scenarios that closely mirror real-life data sets, encompassing all the inherent challenges they typically present (e.g., the high class imbalance). We show that the simulation engine is capable of accurately generating the specified (network) characteristics. For example, by specifying a network effect in the fraud generating model, we can create a dense network of fraudulent claims which have fewer connections to non-fraudulent claims. Further, the results indicate that, in data sets characterized by a strong interconnectedness between fraudsters, we can improve the fraud detection model's accuracy by combining the traditional claim characteristics with social network features.

1.3 Research contributions

The thesis chapters are based on the following publications and working paper:

1. Campo, B.D.C. and Antonio, K. (2023). Insurance pricing with hierarchically structured data: an illustration with a workers' compensation insurance portfolio. *Scandinavian Actuarial Journal*. **2023**(9), 853-884. `https://doi.org/10.1080/03461238.2022.2161413`.

2. Campo, B.D.C. and Antonio, K. (2024). On clustering levels of a hierarchical categorical risk factor. *Annals of Actuarial Science*. In press.

3. Campo, B.D.C. and Antonio, K. (2023). An engine to simulate insurance fraud network data. *arXiv: 2304.09046*. Available at: `https://arxiv.org/abs/2304.09046`.

The author published the following package on the Comprehensive R Archive Network (CRAN):

Campo, B.D.C. (2023). *The actuaRE package: Handling Hierarchically Structured Risk Factors using Random Effects Models.* R package version 0.1.3, `https://cran.r-project.org/package=actuaRE`

Additionally, the following `R` packages were developed and released on Github:

Campo, B.D.C (2021). *BiRankFraud: implementation of the BiRank algorithm to calculate network-based fraud scores.* Available at: `https://github.com/BavoDC/BiRankFraud/`.

Campo, B.D.C. (2023). *iFraudSimulator: implementation of the simulation engine to generate insurance fraud network data.* Available at: `https://github.com/BavoDC/iFraudSimulator/`

Chapter 2

Insurance pricing with hierarchically structured data: An illustration with a workers' compensation insurance portfolio

Actuaries use predictive modeling techniques to assess the loss cost on a contract as a function of observable risk characteristics. State-of-the-art statistical and machine learning methods are not well equipped to handle hierarchically structured risk factors with a large number of levels. In this chapter, we demonstrate the data-driven construction of an insurance pricing model when hierarchically structured risk factors, contract-specific as well as externally collected risk factors are available. We examine the pricing of a workers' compensation insurance product with a hierarchical credibility model (Jewell, 1975), Ohlsson's combination of a generalized linear and a hierarchical credibility model (Ohlsson, 2008) and mixed models. We compare the predictive performance of these models and evaluate the effect of the distributional assumption on the target variable by comparing linear mixed models with Tweedie generalized linear mixed models. For our case-study the Tweedie distribution is well suited to model and predict the loss cost on a contract. Moreover, incorporating

contract-specific risk factors in the model improves the predictive performance and the risk differentiation in our workers' compensation insurance portfolio.

This chapter is based on joint work with Katrien Antonio, which is published in the Scandinavian Actuarial Journal (Campo and Antonio, 2023).

2.1 Introduction

When pricing insurance contracts via risk classification, property and casualty (P&C or general, non-life) insurers use observable characteristics to group policyholders with a similar risk profile in tariff classes. To construct these tariff classes, we either use supervised or unsupervised learning techniques or a combination of both. For example, Henckaerts et al. (2021) developed a tariff structure using tree-based machine learning methods, Gao and Wüthrich (2018) employed clustering techniques to group policyholders with similar driving behavior and Zhu and Wüthrich (2021) combined image classification with clustering techniques to differentiate between driving styles.

Actuaries then estimate the loss cost for each constructed tariff class as a function of the observed risk characteristics using supervised learning methods. Within P&C insurance, continuous and geographical risk factors are typically binned into categorical variables with a limited number of levels. This transformation is either based on expert opinion (Frees and Valdez, 2008; Antonio et al., 2010) or obtained in a data-driven way (Henckaerts et al., 2018). The categorical format enables the construction of an interpretable tariff list that is easily explainable to all stakeholders. However, certain types of risk factors pose a challenge when we want to incorporate them in a pricing model. This particularly holds true for hierarchically structured risk factors with a large number of levels, which are also known as high-cardinality risk factors within the machine learning literature (Micci-Barreca, 2001) or as multi-level risk factors (MLF) within the actuarial literature (Ohlsson and Johansson, 2010). In this chapter, we illustrate the construction of a data-driven insurance pricing model when both hierarchically structured risk factors and contract-specific risk factors are available.

Currently, generalized linear models (GLMs) (McCullagh and Nelder, 1999) are regarded as state-of-the-art for insurance pricing (Haberman and Renshaw, 1996; de Jong and Heller, 2008; Frees, 2015). One of the main advantages of GLMs is that the assumed distribution of the response variable belongs to the exponential family, thereby facilitating the modeling of non-normally distributed response variables

such as the claim frequency or severity. The frequency-severity decomposition is a popular modeling strategy among P&C insurers (Denuit et al., 2007; Frees et al., 2014; Parodi, 2014; Ohlsson and Johansson, 2010; Henckaerts et al., 2018, 2021), where separate predictive models are built for the claim frequency and severity. In this approach we include contracts that reported zero claims during the policy period in the frequency model, but omit these when modeling the claim severity. Alternatively, we can use a Tweedie GLM which enables modeling the zero and continuous positive claim costs simultaneously (Jørgensen and Souza, 1994; Smyth and Jørgensen, 2002; Ohlsson and Johansson, 2010). Recently, the traditional GLM is being challenged by machine learning methods. In contrast to GLMs, such methods are able to learn complex nonlinear transformations and interactions between risk factors without having to specify them explicitly (Blier-Wong et al., 2021). Henckaerts et al. (2021) and Yang et al. (2018), for example, showed how tree-based machine learning methods can be used to develop pricing models that outperform the classical GLM. Notwithstanding, machine learning methods have their own drawbacks. They might be more prone to overfitting (Ying, 2019; Fang, 2019; Colbrook et al., 2022), less transparent (Henckaerts et al., 2022; Dastile et al., 2020) and cannot reliably estimate the prediction uncertainty (Lakshminarayanan et al., 2017; Ovadia et al., 2019; Tohme et al., 2022; Kläs and Vollmer, 2018).

Both GLMs and machine learning methods experience difficulties when confronted with MLFs. Within car insurance, a typical example would be the car model. Due to the large number of levels we often have insufficient data to get reliable parameter estimates when using a GLM with car model as a factor variable. Further, machine learning methods become computationally intractable when dummy encoding is applied to the MLFs. We focus on MLFs that exhibit a hierarchical structure and a typical example hereof, within workers' compensation insurance, is the NACE code. NACE stands for the statistical classification of economic activities in the European community (European Commission and Eurostat, 2017) and is used as a hierarchical classification system to group similar companies based on their economic activities. The NACE code consists of 4 hierarchical levels. When only using the first two levels, an example of a NACE code would be *A03*. The letter is used to identify the first level and *A* stands for *Agriculture, Forestry and Fishing*. The numbers following the letter identify the second level, nested within the first level. Here, *03* refers to *Fishing and aquaculture*. One way to handle (hierarchical) MLFs is via preprocessing with encoding methods that transform the categorical variable into a continuous one, see e.g. the strategy proposed in Micci-Barreca (2001).

Alternatively, we can introduce hierarchically structured random effects into our predictive model to handle the hierarchical MLF. Random effects make optimal use of both the within-cluster and between-cluster claims experience. Applied to our example, at the second level in the hierarchy, the within-cluster claims experience for *A03* refers to the experience obtained from all companies in *03* within *A*. At the first level in the hierarchy, it entails the experience from all companies within cluster *A*. Between-cluster experience refers to the differences observed when comparing the claims experience across different clusters at the first (i.e. clusters *A*, *B*, ...) and second (i.e. clusters within *A*) level in the hierarchy. Random effects allow to account for within-cluster dependency and between-cluster heterogeneity present in hierarchically structured data and enable the prediction of the loss cost as a function of both the contract-specific risk factors and the hierarchical MLF. Further, to estimate the effect of the hierarchical MLF, we only have to estimate the variance of the (hierarchical) level-specific effects. Consequently, in comparison to dummy encoding, the random effects approach drastically reduces the number of parameters. A drawback of the random effects approach is that their estimation and the interpretation of the model output is more cumbersome than with a traditional GLM (Bolker et al., 2009; Zuur et al., 2009; Harrison et al., 2018).

The hierarchical credibility model of Jewell (1975) is one of the best-known actuarial random effects models. In this model, only assumptions on the mean and variance of the random variables (i.e. the response variable and the random effects) are made, making it a distribution-free approach. The hierarchical credibility model (or Jewell model, we use these terms interchangeably), however, does not allow the inclusion of contract-specific risk factors. Ohlsson (2008) therefore combined a GLM with the hierarchical credibility model which allows for a distributional assumption on the response. Another approach that makes use of random effects is the mixed models framework. Mixed models extend GLMs to accommodate correlated or clustered responses. In this framework, we impose distributional assumptions on the response, conditional on the random effects, and on the random effects. Within the actuarial literature, there are numerous papers that illustrate and advocate their use in ratemaking. Moreover, Frees et al. (1999) showed how several credibility models, including the hierarchical credibility model, can be expressed as special cases of the linear mixed model (LMM). Antonio and Beirlant (2007) gave a detailed overview of the theory and actuarial applications of generalized linear mixed models (GLMMs) as well as several advantages of using GLMMs. Another illustration is given in Antonio et al. (2010), where a hierarchically structured intercompany claim data set

on fleet contracts was analyzed using GLMMs and Bayesian estimation techniques.

This chapter contributes to the actuarial literature in three ways. First, we provide a detailed discussion (with strengths and weaknesses) of pricing a workers' compensation insurance product with the hierarchical credibility model (Jewell, 1975), Ohlsson's combination of a generalized linear and a hierarchical credibility model (Ohlsson, 2008) and via the framework of (generalized and linear) mixed models. Second, we develop and demonstrate a comprehensive, data-driven workflow for the use of continuous and spatial covariates in such pricing models. Third, we compare the predictive performance of these models and evaluate the effect of the distributional assumption on the target variable. Hereto we compare linear mixed models and Tweedie generalized linear mixed models.

The organization of this chapter is as follows. In Section 2, we illustrate the general structure of a workers' compensation insurance portfolio and use this as a basis to introduce the theoretical framework. In Section 3, we present a case study on a workers' compensation insurance product including an exploratory analysis, the preprocessing of the database, the development of predictive models and the evaluation of their predictive performance. We construct models under different distributional assumptions for the outcome variable, using different sets of company-specific risk factors (i.e. no risk factors, internal risk factors only, internal and externally collected risk factors). We assess the effect of the distributional assumption and the added value of internally and externally collected risk factors by comparing the predictive performance of different model specifications. We conclude with a discussion in Section 4.

2.2 Predictive modeling with hierarchically structured data in the presence of observable risk factors

2.2.1 Portfolio of hierarchically structured risks

Within insurance pricing, we are interested in determining the loss cost Y defined as the ratio between the claim cost Z and a corresponding exposure measure w of a contract, such as the duration of a policy (Ohlsson and Johansson, 2010). Some insurance portfolios are characterized by an inherent hierarchical structure and of these, a portfolio of workers' compensation insurance contracts is a prime example.

This insurance product provides a financial compensation for lost wages and medical expenses to employees who suffer from a job-related injury (European Insurance and Occupational Pensions Authority, 2020; Stassen et al., 2017). Most often this product is subscribed by the employer, which is typically the company where the employee works. In a workers' compensation insurance product, we commonly define the loss cost Y as the ratio of the total claim amount Z to the salary mass w (Frees, 2010; Denuit, Hainaut and Trufin, 2019). To illustrate the typical hierarchical structure of a workers' compensation insurance portfolio, a hypothetical example is given in Figure 2.1. We first group the companies into different clusters based on their primary business activity and we refer to this as the industry level. Next, we group the companies via branches within industries. Within each of these branches, we have several companies for which we have yearly data available. Due to the nested structure, there will be heterogeneity between clusters and dependency among observations belonging to the same cluster. It is of utmost importance that this is accurately reflected in our predictive models.

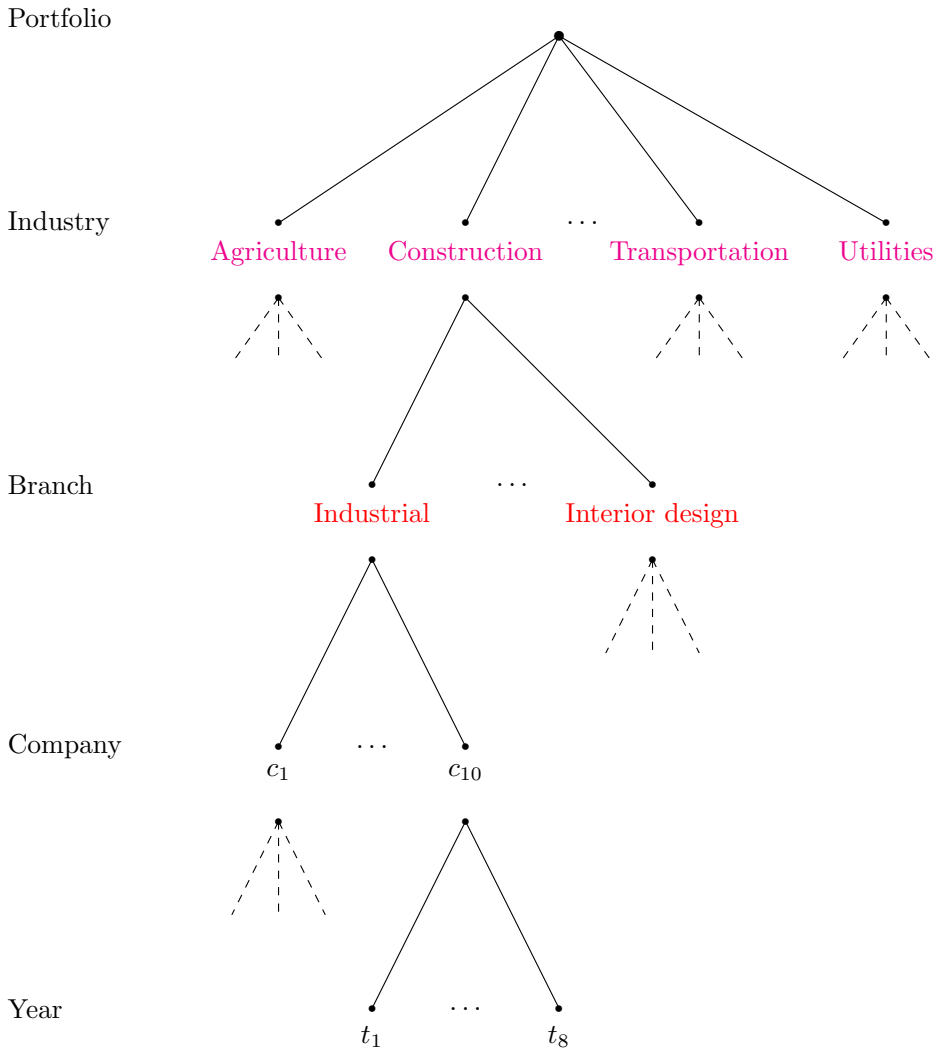
In addition to the hierarchical MLF, insurance companies use historical data on policies and claims which is their main source of information (Ohlsson and Johansson, 2010) and we refer to this as internal data. Depending on the insurer, information at various hierarchical levels may be available. For example, at the company-level, the insurer may have information on the company size or number of employees. This internally collected data can be supplemented with data from an external source to obtain complementary information on the contracts (e.g. financial statements). This information may help in explaining an additional part of the observed heterogeneity.

Our analysis puts focus on the loss cost Y_{ijkt}

$$Y_{ijkt} = \frac{Z_{ijkt}}{w_{ijkt}} \tag{2.1}$$

where we account for inflation by using the year-specific salary mass w_{ijkt}. Here, i serves as an index for the risk profile based on the internally and externally collected company-specific risk factors, j denotes the industry, k the branch and t the annual or repeated observations. Using these indices, the company-specific covariate vector is denoted as $\boldsymbol{x}_{ijkt}$.

Figure 2.1: Hierarchical structure of a hypothetical example.



2.2.2 Random effects model specification

We specify the following functional form for a random effects model that satisfies our requirements

$$\begin{aligned} g(E[Y_{ijkt}|U_j, U_{jk}]) &= \mu + \boldsymbol{x}_{ijkt}^\top \boldsymbol{\beta} + U_j + U_{jk} \\ &= \zeta_{ijkt} \end{aligned} \tag{2.2}$$

where $g(\cdot)$ denotes the link function (for example the identity or log link), μ the intercept, $\boldsymbol{x}_{ijkt}$ the company-specific covariate vector and $\boldsymbol{\beta}$ the corresponding parameter vector. With the model parameters μ and $\boldsymbol{\beta}$ we capture the company-specific effects. To assess the effect of the hierarchical MLF, we introduce the random effects U_j and U_{jk} which capture the unobservable effects of the industry and the branch in which the company operates. U_j denotes the industry-specific deviation from $\mu + \boldsymbol{x}_{ijkt}^\top \boldsymbol{\beta}$ and U_{jk} denotes the branch-specific deviation from $\mu + \boldsymbol{x}_{ijkt}^\top \boldsymbol{\beta} + U_j$. We assume that the random industry effects U_j are independent and identically distributed (i.i.d.) with $E[U_j] = 0$ and $\text{Var}[U_j] = \sigma_I^2$. Similarly, the random branch effects U_{jk} are assumed to be i.i.d. with $E[U_{jk}] = 0$ and $\text{Var}[U_{jk}] = \sigma_B^2$. We do not specify any company-specific random effects since we want to construct an a priori pricing model.

We refer to the right hand-side of equation (2.2) as the systematic model component, which specifies how the company-specific covariates and hierarchical MLF are combined with $\mu, \boldsymbol{\beta}, U_j$ and U_{jk} to give the linear predictor ζ_{ijkt}. Next to this systematic component, we introduce a distributional assumption for the conditional response $Y_{ijkt}|U_j, U_{jk}$. We assume that the distribution belongs to the exponential family with probability density function (pdf)

$$\begin{aligned} f(Y_{ijkt}|\mu, \boldsymbol{\beta}, U_j, U_{jk}, \phi, w_{ijkt}) = \\ \exp\left\{ \frac{Y_{ijkt}\theta_{ijkt} - \psi(\theta_{ijkt})}{\phi} w_{ijkt} + c(y_{ijkt}, \phi, w_{ijkt}) \right\} \end{aligned} \tag{2.3}$$

where $\psi(\cdot)$ and $c(\cdot)$ are known functions, ϕ denotes the dispersion parameter and θ_{ijkt} is the natural parameter. Further, the following conditional relations hold

$$\begin{aligned} g^{-1}(\zeta_{ijkt}) &= E[Y_{ijkt}|U_j, U_{jk}] = \psi'(\theta_{ijkt}), \\ \text{Var}[Y_{ijkt}|U_j, U_{jk}] &= \frac{\phi}{w_{ijkt}} \psi''(\theta_{ijkt}) = \frac{\phi}{w_{ijkt}} V(g^{-1}(\zeta_{ijkt})), \end{aligned} \tag{2.4}$$

where $V(\cdot)$ denotes the variance function.

Given the continuous nature of the registered losses, we can choose to model the loss cost by assuming a Gaussian distribution with an identity link function. However, a disadvantage of the Gaussian assumption is the handling of contracts with a zero loss cost (i.e. no claim occurred). Moreover, when fitting a Gaussian distribution on a sample containing zero-valued observations, the resulting fit will implicitly assume the existence of negative values due to the symmetric nature of

this distribution. Consequently, this may inadvertently lead to predictions with a negative value which is undesirable in a pricing model. To address this shortcoming of the Gaussian distribution, we can opt for either the frequency-severity or Tweedie approach to appropriately model the zero and non-zero valued observations. The advantage of the Tweedie approach, compared to the frequency-severity strategy, is that we are able to model the claim frequency and severity simultaneously. This allows us to estimate the loss cost directly.

One of the most defining characteristics of the Tweedie distribution (see Delong et al. (2021), for example) is the relationship between the variance function and the mean

$$\text{Var}[Y_{ijkt}|U_j, U_{jk}] = \frac{\phi \cdot (g^{-1}(\zeta_{ijkt}))^p}{w_{ijkt}} \tag{2.5}$$

with $p \in (-\infty, 0] \cup [1, \infty)$. The Tweedie family of distributions encompasses a large range of distributions, which are characterized by the value of p (see Table 2.1).

Table 2.1: The power parameter p and its associated distribution.

Value of p	Distribution
$p = 0$	Normal
$p = 1$	Poisson
$p \in (1, 2)$	Compound Poisson - Gamma
$p = 2$	Gamma
$p = 3$	Inverse Gaussian

2.2.3 Parameter estimation

Hierarchical credibility model The basic hierarchical credibility model of Jewell (1975) corresponds to a random effects model where no company-specific covariates (i.e. $\boldsymbol{x}_{ijkt}^\top \boldsymbol{\beta} = 0$ in (2.2)) are included and where $g(\cdot)$ is the identity link function

$$E[Y_{ijkt}|U_j, U_{jk}] = \mu + U_j + U_{jk}. \tag{2.6}$$

We refer to (2.6) as the additive Jewell model. We focus on the estimation of the industry expectation $V_j = \mu + U_j$ and the branch expectation $V_{jk} = \mu + U_j + U_{jk}$ (Dannenburg et al., 1996). These represent the conditional mean of all observations in industry j and of all observations in branch k within industry j, respectively, since $E[Y_{ijkt}|U_j] = V_j$ and $E[Y_{ijkt}|U_j, U_{jk}] = V_{jk}$. Following Ohlsson (2005; 2008),

we rewrite the hierarchical credibility model of Jewell (1975) as

$$E[Y_{ijkt}|V_j, V_{jk}] = V_{jk} \quad \text{and} \quad E[Y_{ijkt}|V_j] \quad = V_j. \tag{2.7}$$

We make the following assumptions

Assumption 1.

(a) The industries are independent, i.e. (Y_{ijkt}, V_j, V_{jk}) and $(Y_{i'j'k't'}, V_{j'}, V_{j'k'})$ are independent for $j \neq j'$.

(b) For every j, conditional on the industry effect V_j, the branches are independent, i.e. (Y_{ijkt}, V_{jk}) and $(Y_{i'jk't'}, V_{jk'})$ are conditionally independent for $k \neq k'$.

(c) All the pairs (V_j, V_{jk}), $j = (1, \ldots, J)$; $k = 1, \ldots, K_j$; are identically distributed, with $E[V_j] = \mu > 0$, $E[V_{jk}|V_j] = V_j$, $\text{Var}[V_j] = \sigma_I^2$ and $E[\text{Var}[V_{jk}|V_j]] = \sigma_B^2$.

(d) For any $(j,\ k)$, conditional on (V_j, V_{jk}), the Y_{ijkt} are independent, with mean V_{jk} and with variance satisfying $E[\text{Var}[Y_{ijkt}|V_j, V_{jk}]] = \sigma^2/w_{ijkt}$.

We use $\mu = E[Y_{ijkt}] = E[V_j] = E[V_{jk}]$ to denote the overall expectation. Using Assumption 1 (c) and (d), it follows that

$$\begin{aligned} \text{Var}[Y_{ijkt}] &= E[\text{Var}[Y_{ijkt}|V_j, V_{jk}]] + \text{Var}[E[Y_{ijkt}|V_j, V_{jk}]] \\ &= \frac{\sigma^2}{w_{ijkt}} + \sigma_I^2 + \sigma_B^2. \end{aligned} \tag{2.8}$$

The credibility estimator of V_j (Dannenburg et al., 1996; Ohlsson, 2005, 2008), under Assumption 1, is defined as

$$\widehat{V}_j = q_j \bar{Y}^z_{\cdot j \cdot \cdot} + (1 - q_j)\mu, \tag{2.9}$$

where

$$\begin{aligned} \bar{Y}_{\cdot jk \cdot} &= \frac{\sum_{i,t} w_{ijkt} Y_{ijkt}}{\sum_{i,t} w_{ijkt}}, \quad z_{jk} = \frac{w_{\cdot jk \cdot}}{w_{\cdot jk \cdot} + \sigma^2/\sigma_B^2}, \\ \bar{Y}^z_{\cdot j \cdot \cdot} &= \frac{\sum_k z_{jk} \bar{Y}_{\cdot jk \cdot}}{\sum_k z_{jk}}, \quad \text{and } q_j = \frac{z_{j \cdot}}{z_{j \cdot} + \sigma_B^2/\sigma_I^2}, \end{aligned} \tag{2.10}$$

and we define $w_{\cdot jk \cdot} = \sum_{i,t} w_{ijkt}$. The credibility estimator of V_{jk} is specified as

$$\widehat{V}_{jk} = z_{jk} \bar{Y}_{\cdot jk \cdot} + (1 - z_{jk})\widehat{V}_j. \tag{2.11}$$

Here, q_j and z_{jk} are the credibility factors at the industry- and branch-level, respectively. $\bar{Y}_{\cdot jk\cdot}$ represents the weighted average for the k^{th} branch within industry j and serves as an estimator of the average loss cost at the branch level. The estimator of the average loss cost at the industry level is denoted by $\bar{Y}^z_{\cdot j\cdot\cdot}$ and we use the superscript z to indicate that we weigh the averages $\bar{Y}_{\cdot jk\cdot}$ with the credibility factors z_{jk} instead of the original weights $w_{\cdot jk\cdot}$. The latter estimators, however, are not optimal for clusters that have a low number of observations. We therefore use the credibility estimators $\widehat{V}_j$ and $\widehat{V}_{jk}$, which are a weighted sum of a more stable average and a less stable, more cluster-specific average. To use these credibility estimators, we first require estimators of the variance parameters σ^2, σ_I^2 and σ_B^2 as well as an estimator of μ (see Appendix A.1). We refer the reader to Dannenburg et al. (1996), Ohlsson (2005) and Ohlsson and Johansson (2010) for detailed information on these estimators. Next, we predict the damage rate using $\widehat{Y}_{ijkt} = \widehat{V}_{jk}$.

The hierarchical credibility model is relatively easy to implement and computationally light, which is one of its advantages. Furthermore, we only require estimates of the mean and variance parameters to obtain the random effect estimates. Statistical inference on the estimated parameters, however, is not possible with this distribution-free approach.

Combining the hierarchical credibility model with a GLM Ohlsson (2008) reformulates the hierarchical credibility model in (2.7) as a multiplicative random effects model by defining $V_j = \tilde{\mu}\ \widetilde{U}_j$ and $V_{jk} = \tilde{\mu}\ \widetilde{U}_j\ \widetilde{U}_{jk} = V_j \widetilde{U}_{jk}$. Consequently,

$$E[Y_{ijkt}|\widetilde{U}_j, \widetilde{U}_{jk}] = \tilde{\mu}\ \widetilde{U}_j\ \widetilde{U}_{jk} \tag{2.12}$$

and we refer to (2.12) as the multiplicative Jewell model. To obtain this multiplicative structure in (2.2) we define $g(\cdot) = \log(\cdot)$. In this case, $\tilde{\mu} = e^{\mu}$, $\widetilde{U}_j = e^{U_j}$ and $\widetilde{U}_{jk} = e^{U_{jk}}$. To allow for company-specific covariates, Ohlsson extends (2.12) to

$$E[Y_{ijkt}|\widetilde{U}_j, \widetilde{U}_{jk}] = \tilde{\mu}\ \gamma_{ijkt}\ \widetilde{U}_j\ \widetilde{U}_{jk} = \gamma_{ijkt} V_{jk} \tag{2.13}$$

where γ_{ijkt} denotes the effect of the company-specific covariates. We add a distributional assumption and assume that $Y_{ijkt}|\widetilde{U}_j, \widetilde{U}_{jk} \sim \mathcal{T}(\gamma_{ijkt} V_{jk}, \frac{\phi}{w_{ijkt}}(\gamma_{ijkt} V_{jk})^p))$, where $\mathcal{T}$ denotes any member of the Tweedie family (see Table 2.1). To estimate the parameters in this model, Ohlsson (2008) devised the iterative GLMC (GLMs with credibility) algorithm which is given in Algorithm 1.

Algorithm 1: Iterative GLMC algorithm (Ohlsson, 2008)

Model: $E[Y_{ijkt}|\widetilde{U}_j, \widetilde{U}_{jk}] = \tilde{\mu}\ \gamma_{ijkt}\ \widetilde{U}_j\ \widetilde{U}_{jk}$

Initialization: Set $\widehat{\widetilde{U}}_j = \widehat{\widetilde{U}}_{jk} = 1$

repeat

1. Estimate $\tilde{\mu}$, γ_{ijkt} and p using a GLM with log link, include the $\log(\widehat{\widetilde{U}}_j)$ and $\log(\widehat{\widetilde{U}}_{jk})$ as offset variables and the w_{ijkt}'s as weights. This yields $\hat{\tilde{\mu}}$, $\hat{\gamma}_{ijkt}$ and $\hat{p}$;
2. Use $\hat{\tilde{\mu}}$ and $\hat{\gamma}_{ijkt}$ to estimate σ^2, σ_B^2 and σ_I^2 with the hierarchical credibility model (Dannenburg et al., 1996; Ohlsson, 2005, 2008) ;
3. Compute $\widehat{V}_j$ and $\widehat{V}_{jk}$ using the estimates from steps 1 and 2 (see (2.9) and (2.11)). Calculate $\widehat{\widetilde{U}}_j = \widehat{V}_j/\hat{\mu}$ and $\widehat{\widetilde{U}}_{jk} = \widehat{V}_{jk}/\widehat{V}_j$;

until *convergence*;

We initialize the model by setting $\widehat{\widetilde{U}}_j = \widehat{\widetilde{U}}_{jk} = 1$ and proceed to the first step, where we fit a GLM. When fitting the GLM, we include $\log(\widehat{\widetilde{U}}_j)$ and $\log(\widehat{\widetilde{U}}_{jk})$ as offset variables and the w_{ijkt}'s as weights. This results in the GLM estimates $\hat{\mu}$ (intercept), $\hat{\boldsymbol{\beta}}$ (company-specific parameter vector) and $\hat{p}$ (the power parameter). We compute $\hat{\tilde{\mu}} = e^{\hat{\mu}}$ and $\hat{\gamma}_{ijkt} = e^{\boldsymbol{x}_{ijkt}^\top \hat{\boldsymbol{\beta}}}$ to proceed to the second step. Here, we first transform the response variable Y_{ijkt} and weight w_{ijkt} as

$$\tilde{Y}_{ijkt} = \frac{Y_{ijkt}}{\gamma_{ijkt}} \quad \text{and} \quad \tilde{w}_{ijkt} \quad = w_{ijkt}\gamma_{ijkt}^{(2-p)}. \tag{2.14}$$

Consequently,

$$\begin{aligned} E[\tilde{Y}_{ijkt}|V_j, V_{jk}] &= \frac{1}{\gamma_{ijkt}}\gamma_{ijkt}V_{jk} = V_{jk}, \\ E[\tilde{Y}_{ijkt}|V_j] &= \frac{1}{\gamma_{ijkt}}\gamma_{ijkt}V_j = V_j, \\ E\left[\mathrm{Var}\left[\tilde{Y}_{ijkt}|V_j, V_{jk}\right]\right] &= E\left[\frac{1}{\gamma_{ijkt}^2}\frac{\phi\cdot(\gamma_{ijkt}V_{jk})^p}{w_{ijkt}}\right] \\ &= \frac{\phi\cdot E[(V_{jk})^p]}{w_{ijkt}\gamma_{ijkt}^{(2-p)}} = \frac{\sigma^2}{\tilde{w}_{ijkt}}. \end{aligned} \tag{2.15}$$

where $\sigma^2 = \phi\cdot E[(V_{jk})^p]$. $\tilde{Y}_{ijkt}$ and $\tilde{w}_{ijkt}$ now satisfy the assumptions of the hierarchical credibility model (see Assumption 1), thereby enabling us to estimate the variance parameters and to calculate $\widehat{V}_j$ and $\widehat{V}_{jk}$ using equations (2.9) and

(2.11). In the third step, we compute the random effect estimates $\widehat{\widehat{U}}_j = \widehat{V}_j/\hat{\mu}$ and $\widehat{\widehat{U}}_{jk} = \widehat{V}_{jk}/\widehat{V}_j$ using the estimates from steps 1 and 2. Steps 1 to 3 are repeated until the algorithm has converged. Once converged, we predict the damage rate as $\widehat{Y}_{ijkt} = \hat{\hat{\mu}}\ \hat{\gamma}_{ijkt}\ \widehat{\widehat{U}}_j\ \widehat{\widehat{U}}_{jk}$.

Similarly to the hierarchical credibility model, Ohlsson's GLMC algorithm is relatively easy to implement, computationally light and only requires estimates of the mean and variance parameters to obtain the random effect estimates. Further, Ohlsson's approach allows for statistical inference on the parameters estimated by the GLM. Hereto, we use the fitted GLM from the last run in Algorithm 1 and base the inference on the following likelihood

$$\prod_j \prod_k \prod_{i,t} f(Y_{ijkt}|\hat{\mu}, \hat{\boldsymbol{\beta}}, \hat{p}, \log(\widehat{\widehat{U}}_j), \log(\widehat{\widehat{U}}_{jk}), \phi, w_{ijkt}) \tag{2.16}$$

where the log-transformed random effects $\widehat{\widehat{U}}_j$ and $\widehat{\widehat{U}}_{jk}$ enter as constants. Hence, the statistical inference on $\hat{\mu}$ and $\hat{\boldsymbol{\beta}}$ ignores the variability in the random effects.

Mixed models (Generalized) Linear mixed models (GLMMs) are considered an extension of (G)LMs to the case where responses are correlated or clustered (Molenberghs and Verbeke, 2005; Tuerlinckx et al., 2006). They are founded in a well-developed statistical framework that provides us with the appropriate inferential tools. The framework of mixed models encompasses a wide range of model specifications, including models with hierarchically structured random effects.

Applied to our setting, the general equation for a mixed model is the same as in equation (2.2). We assume that $Y_{ijkt}|U_j, U_{jk} \sim \mathcal{E}(g^{-1}(\zeta_{ijkt}), \frac{\phi}{w_{ijkt}} V(g^{-1}(\zeta_{ijkt})))$, where $\mathcal{E}$ denotes any member of the exponential family, and make a distributional assumption on the random effects U_j and U_{jk}. In most applications we assume that $U_j \sim \mathcal{N}(0, \sigma_I^2)$, $U_{jk} \sim \mathcal{N}(0, \sigma_B^2)$ (McCulloch and Neuhaus, 2011; Drikvandi et al., 2017), where $\mathcal{N}$ denotes the normal distribution. Verifying these assumptions, however, is often not straightforward. The linear mixed model (LMM) is a special case of a GLMM, where we define $\mathcal{E} := \mathcal{N}$ and use the identity-link function $g(\cdot)$. The additive hierarchical credibility model discussed in (2.6) is a special case of an LMM and both use the same equations to estimate μ, U_j and U_{jk} (Frees et al., 1999). The variance parameters, however, are estimated differently in the additive hierarchical credibility model compared to the LMM.

We maximize the marginal likelihood to obtain estimates of the parameters

$\mu, \boldsymbol{\beta}, \phi, \sigma_I^2,\ \sigma_B^2$ (and p in case of a Tweedie GLMM). The marginal likelihood is obtained by integrating out the random effects and is given by

$$\prod_j \int \left[\prod_k \int \prod_{i,t} f(Y_{ijkt}|\Theta, U_j, U_{jk}, \phi, w_{ijkt}) f(U_{jk}|\sigma_B^2) dU_{jk} \right] f(U_j|\sigma_I^2) dU_j. \quad (2.17)$$

where $\Theta = (\mu, \boldsymbol{\beta}, p)$ for a Tweedie GLMM and $\Theta = (\mu, \boldsymbol{\beta})$ for other GLMMs. For an LMM, an analytical expression is available for the integrals. In this case, we use the generalized least squares estimator to estimate μ and $\boldsymbol{\beta}$ and rely on either maximum likelihood or restricted maximum likelihood estimators for the estimation of the parameters ϕ, σ_I^2 and σ_B^2. Conversely, in most GLMMs there is no analytic expression available for the integrals in (2.17) and we therefore rely on numerical approximations to estimate the parameters. These approximations can be subdivided into those that approximate the integrand, the data or the integral. A detailed discussion on the different approximation methods is covered in Molenberghs and Verbeke (2005), Tuerlinckx et al. (2006) and Frees et al. (2014). In mixed models, we base the statistical inference on (2.17) and we account for the variability in the random effects when performing inference on $\hat{\mu}$ and $\hat{\boldsymbol{\beta}}$. Further, several hypothesis tests are available for the variance parameters σ_I^2 and σ_B^2.

To predict the realized values of the random effects U_j and U_{jk}, we rely on empirical Bayes estimates. Hereto, we base the estimation on the posterior distribution of the random effects given $Y_{ijkt}, \Theta, \phi, w_{ijkt}, \sigma_I^2$ and σ_B^2 (Fitzmaurice et al., 2008; Molenberghs and Verbeke, 2005; Skrondal and Rabe-Hesketh, 2009). The density of the posterior distribution of U_j is

$$\propto \prod_k \int \prod_{i,t} f(Y_{ijkt}|\Theta, U_j, U_{jk}, \phi, w_{ijkt})\ f(U_{jk}|\sigma_B^2)\ dU_{jk}\ f(U_j|\sigma_I^2). \quad (2.18)$$

For U_{jk}, the density of the posterior distribution is

$$\propto \int \prod_{i,t} f(Y_{ijkt}|\Theta, U_j, U_{jk}, \phi, w_{ijkt})\ f(U_j|\sigma_I^2)\ dU_j\ f(U_{jk}|\sigma_B^2). \quad (2.19)$$

The estimates $\widehat{U}_j$ and $\widehat{U}_{jk}$ are those values for U_j and U_{jk} that maximize the corresponding posterior densities. In these densities, the unknown parameters are replaced by their maximum likelihood estimates. For LMMs, we have a closed-form solution for $\widehat{U}_j$ and $\widehat{U}_{jk}$. Conversely, for most GLMMs we do not have an analytical

expression available and we have to rely on numerical approximations. Hereafter, we predict the damage rate using

$$\widehat{Y}_{ijkt} = g^{-1}(\hat{\mu} + \boldsymbol{x}_{ijkt}^{\top}\hat{\boldsymbol{\beta}} + \widehat{U}_j + \widehat{U}_{jk}). \tag{2.20}$$

2.2.4 Computational aspects and implementation in R

We perform our estimations with the statistical software `R` (R Core Team, 2019). To estimate the random effects model with the hierarchical credibility model (Jewell, 1975) and the combination of the hierarchical credibility model with a GLM (Ohlsson, 2008), we developed our own package called `actuaRE`. This package is publicly available on `https://www.github.com`. For mixed models, a multitude of software implementations are available alongside with detailed documentation. We rely on the `lme4` (Bates et al., 2015) and `cplm` (Zhang, 2013) packages to estimate the random effects model using the mixed model framework.

Estimation via the hierarchical credibility model, as discussed in Section 2.2.3, is fastest in terms of computation time. Estimation via (G)LMMs is by far the slowest as they require the approximation and maximization of complicated likelihoods. Computationally, GLMMs are complex and they are more likely to experience convergence problems (see Bolker et al. (2022) for information on how to handle convergence warnings). Related hereto is that, in certain situations, we may obtain negative variance estimates and this may occur for all estimation methods. Within the mixed models framework, this is a well-known problem (Pryseley et al., 2011). Negative variance estimations may be due to low variability (Oliveira et al., 2017) or a misspecification of the hierarchical MLF (Pryseley et al., 2011).

2.3 Case study: workers' compensation insurance

We illustrate the predictive model building work flow on a workers' compensation insurance data set from a Belgian insurer. In this data set, we have a hierarchical MLF and company-specific covariates at our disposal. Further, we have the company identification number for each of the companies in the portfolio which enables us to retrieve company-specific financial information from an external data source. We refer to the externally acquired data as the external database. To preserve the confidentiality of the data, we omit all confidential information. Hereto, we either remove all values from the figures or apply a transformation when showing values.

2.3.1 Internal data set

To prevent that large claims distort our findings, we start the analysis by capping large claim amounts Z_{hijkt} using concepts from extreme value theory (EVT) (Beirlant et al., 2005). Here, the index h refers to an individual claim of company i operating in branch k within industry j in year t. In the analysis we use the ratio of Z_{hijkt} to a year-specific correction factor c_t, thereby accounting for inflation. Using tools from EVT, we determine the threshold τ between the attritional losses and the large losses. We transform τ to a year-specific threshold using $\tau_t = \tau \times c_t$ and cap Z_{ijkt} as follows

$$\tilde{Z}_{hijkt} = \min(Z_{hijkt}, \tau_t) \tag{2.21}$$

where $\tilde{Z}_{hijkt}$ denotes the capped claim amount. Thereafter, we redistribute the total capped amount among all claims based on their share in the total cost. Hereby, we ensure that the total claim amount after redistribution equals the total claim amount before capping. Given the confidentiality of the data, we do not disclose how we redistribute the total capped amount.

After this first data preprocessing step, we compute the damage rate for each of the individual companies as

$$Y_{ijkt} = \frac{\sum_h \mathcal{Z}_{hijkt}}{w_{ijkt}}. \tag{2.22}$$

The empirical distribution of the damage rates Y_{ijkt} of the individual companies is visualized in panel (a) of Figure 2.2. The empirical distribution is characterized by a strong right skew and this right skew is still present when log transforming Y_{ijkt} for $Y_{ijkt} > 0$ (see Figure 2.2(b)). Of all the individual damage rates Y_{ijkt}, 85% equals zero and 7.5%[1] of the Y_{ijkt}'s are larger than one (Figure 2.2(a)). Hence, the majority of the damage rates are either zero or relatively low compared to the salary mass.

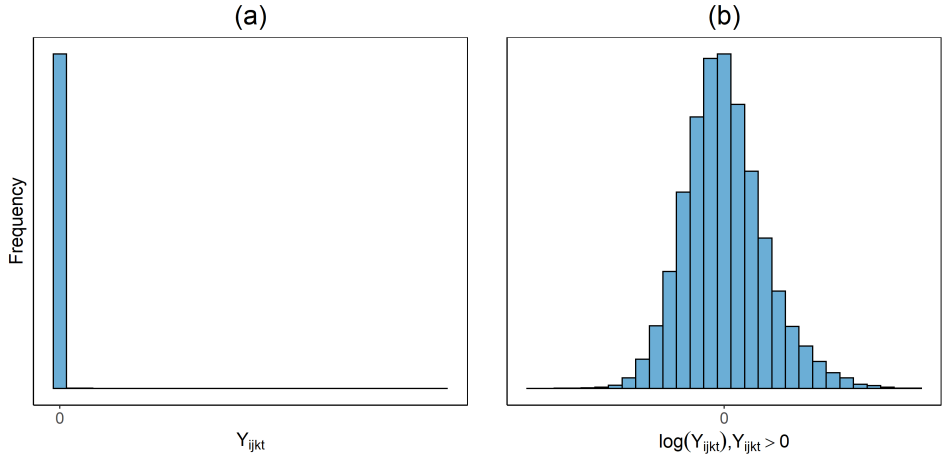
We use the individual Y_{ijkt} and corresponding w_{ijkt} to compute the weighted average of the damage rates at the industry- and branch-level

$$\bar{Y}_{\cdot j\cdot\cdot} = \frac{\sum_{i,k,t} w_{ijkt} Y_{ijkt}}{\sum_{i,k,t} w_{ijkt}}, \qquad \bar{Y}_{\cdot jk\cdot} = \frac{\sum_{i,t} w_{ijkt} Y_{ijkt}}{\sum_{i,t} w_{ijkt}} \tag{2.23}$$

and visualize these in the treemaps in Figure 2.3. Panel (a) shows the $\bar{Y}_{\cdot j\cdot\cdot}$'s and

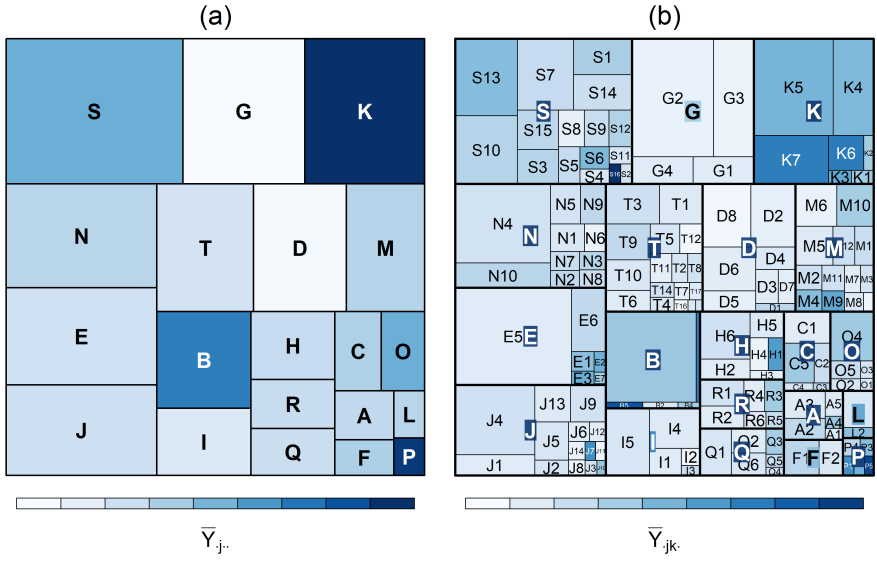
[1]The "0.4% of Y_{ijkt}'s" in Campo and Antonio (2023) refers to the damage rates calculated using the original, uncapped claim amount Z_{hijkt} (see (2.22)).

Figure 2.2: Empirical distribution of the damage rates Y_{ijkt} of the individual companies.



panel (b) the $\bar{Y}_{\cdot jk\cdot}$'s. In the treemaps, the summed salary mass of the industries and branches within industries determines the size of the rectangles and a color gradient is used for the weighted averages. The larger the summed salary mass, the larger the rectangle and the larger the weighted average, the darker the color.

Figure 2.3: Tree maps depicting hierarchical structure.



Considerable variation is present between industries as well as within industries in the weighted averages. In addition, we see that the $\bar{Y}_{\cdot jk\cdot}$'s are more similar within industries than between industries (see Figure 2.3(b)). For industry K, for example, the $\bar{Y}_{\cdot jk\cdot}$'s are visibly larger than those of industry D and within industry K, there are clear differences between the different branches (e.g. between K1 and K7). Consequently, an indispensable part of the variation of Y_{ijkt} seems to be attributable to the industry and branch in which the companies operate.

In addition to the hierarchical MLF, we have company-specific covariates at our disposal, such as the number of full-time equivalents (FTEs) or the company type. We refer to these covariates as the internal variables. For a particular level l of a covariate, we calculate the weighted average of the damage rates as

$$\bar{Y}_l = \frac{\sum\limits_{(i,j,k,t) \,\in\, l} w_{ijkt} Y_{ijkt}}{\sum\limits_{(i,j,k,t) \,\in\, l} w_{ijkt}}. \tag{2.24}$$

Here, $\sum_{(i,j,k,t) \,\in\, l}$ indicates that the summation is limited to those observations that are categorized into level l. For example, when computing $\bar{Y}_l$ with $l =$ `1` for `internal variable 1`, we only consider the Y_{ijkt}'s of companies that have the value `1` for `internal variable 1`. By comparing the $\bar{Y}_l$ across the different levels we empirically explore whether certain levels are considered to be more risky than others with a marginal, empirical analysis.

Figure 2.4 shows the $\bar{Y}_l$'s for the internal variables. To preserve the confidentiality of our findings, we randomly allocate the postal codes to different regions on the map. We preserve this allocation throughout the article for consistency of the results. The weighted average mainly differs between the levels of the variables `internal variable 2`, `internal variable 3`, `internal variable 4` and `postal code`.

2.3.2 External data set

The second source of information is the Bel-First database (`https://belfirst.bvdinfo.com`) which contains the financial statements of all Belgian private entities that file their financial accounts to the National Bank of Belgium (Bureau Van Dijk, 2020). For each of the companies, we have yearly data available in the Bel-First database. We link the claims observed during year t with the financial performance indicators for year $t-1$. For example, we extract the financial information of company i in year 2019 and link this to the registered claims of company i in 2020.

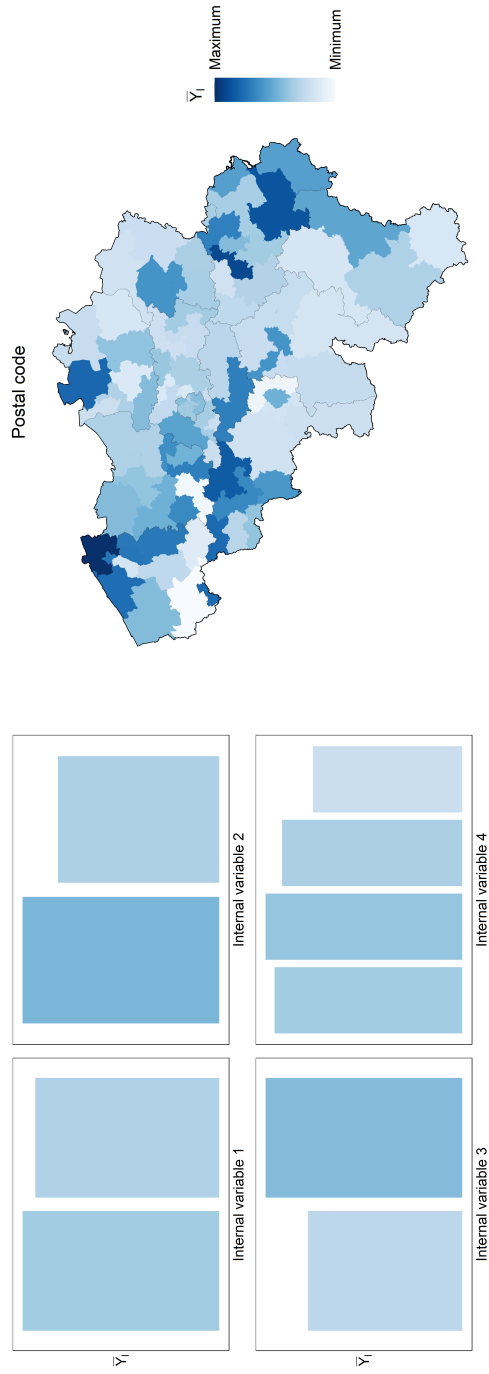


Figure 2.4: Comparison of the $\bar{Y}_l$'s per covariate.

We retrieve information on 30 variables, 26 of which are related to the financial situation of the company and we are able to retrieve information for approximately 70% of the companies. For all extracted variables, we have occasional missing values. For 30% of the companies we are not able to retrieve any financial information since these companies are not obliged to report to the National Bank of Belgium. Of this 30%, the majority of the companies are categorized as independent natural persons, craftsmen. A minority of this 30% are non-profit organizations, private companies with a limited liability, Belgian companies/associations without accounts or Belgian companies/associations that do not file their account in a standard model.

Using the available financial information, additional variables are created. We specify a binary variable that indicates whether the company is considered to be a zombie firm or not and we use the definition of McGowan et al. (2018). According to McGowan et al. (2018), a firm is identified as a zombie firm when its interest coverage ratio (ICR) has been less than one for at least three consecutive years and if the company is at least 10 years old. Next to this zombie variable, we compute the relative change of the variables that are commonly associated with growth. These variables are sales, number of employees, total assets, cash flow and added value (see, for example, Vanacker and Manigart (2008)). The relative change at time t for a variable X is then computed as $(X_t - X_{t-1})/|X_{t-1}|$.

2.3.3 Binning continuous and spatial company-specific covariates

In order to arrive at an interpretable tariff list that is easily explainable to all stakeholders, we transform continuous and spatial company-specific covariates to categorical ones. Hereto we employ a data-driven binning strategy based on the work of Henckaerts et al. (2018). We use a different strategy for continuous and spatial variables to account for the variable type. For continuous variables, we want to preserve the ordering and only allow for the binning of consecutive values. Conversely, for spatial variables we want a strategy that enables us to merge non-adjacent postal codes. In this section, we provide a general description of the binning process.

Continuous variables. We start the binning process of a continuous variable by fitting a univariate generalized additive model (GAM) to the company-level data. In this exploratory preprocessing step, we do not include any random effects in the

GAM for computational simplicity and fit the following model

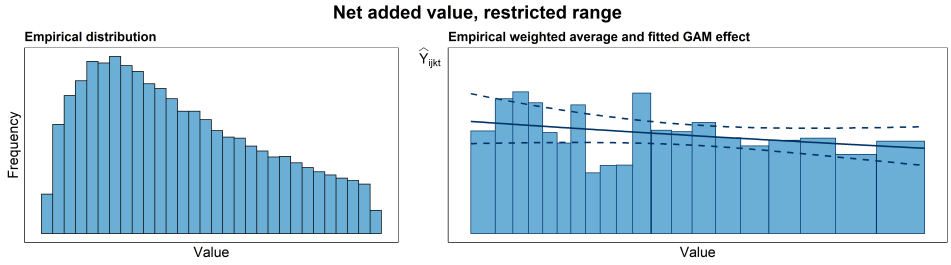
$$g(E[Y_{ijkt}|_{BF}x_{ijkt}, x_{ijkt}]) = \mu + {}_{BF}x_{ijkt}\beta_{BF} + I(x_{ijkt} \text{ available})f(x_{ijkt}) \quad (2.25)$$

where i serves as an index for the company. ${}_{BF}x_{ijkt}$ is a binary variable indicating that no financial information is available for company i in the Bel-First database, x_{ijkt} is the external variable, $I(x_{ijkt}$ available) indicates whether x_{ijkt} is known ($I(x_{ijkt}$ available$) = 1$) or not ($I(x_{ijkt}$ available$) = 0$) and $f(\cdot)$ denotes the smooth effect. This model specification allows us to use all available external information and we hereby do not omit information from companies that either cannot be found in the Bel-First database or that can be found in the Bel-First database, but have a missing value for the covariate. Missing values are assumed to be missing at random for companies found in the Bel-First database. Given the size of our data set we opt for the simplicity of the indicator method to handle missing data (Bennett, 2001). In addition, we include ${}_{BF}x_{ijkt}$ as a confounding variable to account for the fact that certain companies are not found in the Bel-First database. The wage bills w_{ijkt} are incorporated as weights. To examine the effect of the distributional assumption, we perform this procedure once using univariate GAMs with a Gaussian distribution and identity link and once using univariate GAMs with a Tweedie distribution and log link.

An illustration of the binning process for the variable `net added value` is given in Figure 2.5 when assuming a Gaussian distribution for the response in the GAM. The histogram on the left side of the figure shows the empirical distribution of the continuous variable (after limiting its range to non-outlying values to focus on the pattern seen in the majority of the companies) and the figure on the right side shows the fitted smooth effect $g^{-1}(\mu + \hat{f}(x_{ijkt}))$ (black solid line) together with the 95% confidence interval (black dashed lines). The blue bars on the right side depict the empirical weighted averages by consecutively grouping values until they contain at least 5% of the observations.

The fitted smooth effect is binned with a regression tree and the resulting bins are inspected in detail for every covariate. In addition to inspecting the fitted smooth effect and the resulting bins at the original scale of the covariate, we assess the binning based on the log-transformed counterpart for positively valued covariates and we choose the binning that best approximates the empirical weighted averages.

Figure 2.5: Illustration of the binning process for continuous covariates. The histogram on the left shows the empirical distribution of the variable `net added value`, after limiting its range to non-outlying values. The figure on the right depicts the fitted smooth effect $g^{-1}(\mu + \hat{f}(x_{ijkt}))$ (black solid line) together with the 95% confidence interval (black dashed lines). Here, the blue bars depict the empirical weighted averages by consecutively grouping values until they contain at least 5% of the observations.



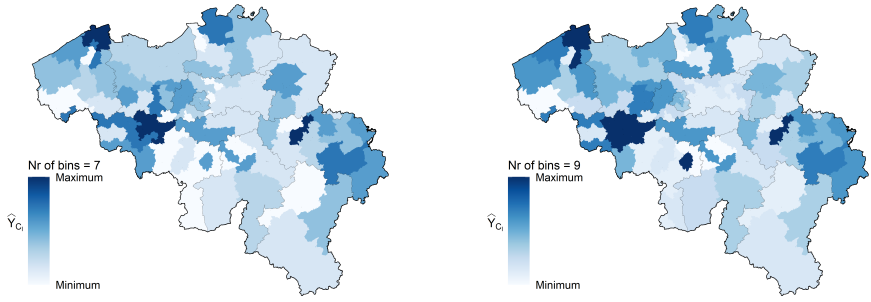
Spatial variable. For postal code, we first construct preliminary clusters by using only the first two digits of the postal code. We use dummy variables to encode the two-digit postal code and fit the following model to the company-level data

$$g(E[Y_{ijkt}|U_j, U_{jk}]) = \mu + \boldsymbol{x}_{ijkt}^\top \boldsymbol{\beta} + U_j + U_{jk}. \tag{2.26}$$

where the covariate vector $\boldsymbol{x}_{ijkt}$ consists of the dummy variables. The model is fit using Ohlsson's iterative algorithm (see Algorithm 1) and the estimated coefficients are clustered using the Ckmeans.1d.dp algorithm (Wang and Song, 2011). To select the number of clusters n_c, we perform a grid search with the AIC as criterion. The results of the clustering strategy are shown in Figures 2.6(a) and 2.6(b) when assuming a Gaussian and Tweedie distribution for the response, respectively. The Gaussian model specification results in seven separate clusters and the Tweedie model specification results in nine clusters. After clustering, we refit the model using Ohlsson's algorithm. The colors in the plot depict the estimated damage rate for each of the clusters. We calculate the estimated damage rate for cluster C_l as $\widehat{Y}_{C_l} = g^{-1}(\mu + \widehat{\beta}_{C_l})$ for $l = (1, \ldots, n_c)$. Here, $\widehat{\beta}_{C_l}$ denotes the estimated coefficient for cluster C_l. Overall, both results closely resemble each other and are in line with the results of our exploratory analysis shown in Figure 2.4.

Figure 2.6: Results binning two-digit postal code.

(a) Gaussian assumption. **(b)** Tweedie assumption.



2.3.4 Development of the predictive model

We split the data into a training and test set. We use the training set to develop the predictive model and the test set to assess its predictive performance. The training set contains data from the first seven years and the test set contains data from the eighth and most recent year.

Preselection of the external covariates. Considering that we have a substantial amount of externally selected company-specific covariates, we first perform a preliminary variable selection to retain the most important external covariates. Since this step determines which external covariates are investigated further, we rely on the well-developed statistical framework of mixed models. This framework enables us to accurately estimate the external covariate parameters and to calculate the marginal AIC (mAIC) of the fitted GLMM (Saefken et al., 2014). The mAIC focuses on the fixed effects and we use this criterion to select the covariates that fit the data well. We start by fitting univariate GLMMs (i.e. only one external covariate is included) to the company-level data with the following general equation

$$g(E[Y_{ijkt}|U_j, U_{jk}]) = \mu + {}_{BF}x_{ijkt}\beta_{BF} + {}_{Ext}\boldsymbol{x}_{ijkt}^{\top}\boldsymbol{\beta}_{Ext} + U_j + U_{jk}. \tag{2.27}$$

Here, ${}_{Ext}\boldsymbol{x}_{ijkt}$ consists of the dummy variables for the binned external covariate as obtained from Section 3.3 and $\boldsymbol{\beta}_{Ext}$ denotes the corresponding parameter vector. To make the model identifiable, observations with a missing value for the external covariate serve as a reference. We compute the mAIC of the univariate GLMM as

specified in (2.27) using

$$\text{mAIC} = -2\log(f(Y_{ijkt}|\Theta, \phi, \sigma_I^2, \sigma_B^2)) + 2(n_p + q + 1) \tag{2.28}$$

where $\log(f(Y_{ijkt}|\Theta, \sigma^2, \sigma_I^2, \sigma_B^2)$ denotes the marginal log-likelihood (see equation (2.17)), $\Theta = (\mu, \beta_{BF}, \boldsymbol{\beta}_{Ext}, p)$ for Tweedie GLMMs and $\Theta = (\mu, \beta_{BF}, \boldsymbol{\beta}_{Ext})$ for other GLMMs, n_p the number of parameters of the external covariate plus intercept and q the number of variance parameters of the random effects. Note that we add a one to $n_p + q$ to account for the estimation of the dispersion parameter ϕ. In case of a Tweedie GLMM, we add a two to $n_p + q$ to account for the estimation of the dispersion parameter ϕ and the power parameter p. We use the mAIC values to select the top 5 covariates. We rely on the mAIC, since information criteria are better suited for model selection than statistical tests (Burnham and Anderson, 2002).

This procedure results in a different set of preselected external covariates, depending on the assumed distribution for the response. Comparing the results based on the LMM and Tweedie GLMM, we see that `external variable 2` and `external variable 3` are the only variables that are selected by both models.

Using the internal and preselected external variables, we compute the damage rate Y_{ijkt} for each possible combination of company-specific covariates and hierarchical MLF values and one such combination determines a tariff class. We use i as an index for the tariff class and calculate Y_{ijkt} as

$$Y_{ijkt} = \frac{\sum_h \mathcal{Z}_{hijkt}}{\sum_h w_{hijkt}} \tag{2.29}$$

where $\mathcal{Z}_{hijkt}$ refers to the capped claim amount of the h^{th} company in tariff class i operating in branch k within industry j at time t and $\sum_h w_{hijkt}$ represents the sum of the corresponding salary masses. All possible tariff classes are combined into a tariff table.

Variable selection. Next, we apply best subset regression (Beale et al., 1967; Hocking and Leslie, 1967) with the Akaike Information Criterion (AIC) (Akaike, 1974) as selection criterion. The general model equation is given by

$$\begin{aligned} g(E[Y_{ijkt}|U_j, U_{jk}]) = \mu + {}_{Int}\boldsymbol{x}_{ijkt}^\top \boldsymbol{\beta}_{Int} + {}_{BF}x_{ijkt}\beta_{BF} + {}_{Z}x_{ijkt}\beta_Z \\ + {}_{Ext}\boldsymbol{x}_{ijkt}^\top \boldsymbol{\beta}_{Ext} + U_j + U_{jk}. \end{aligned} \tag{2.30}$$

where subscripts Int and Z refer to the internal variables and zombie variable, respectively. ${}_{Ext}\boldsymbol{x}_{ijkt}$ refers to the covariate vector of the external covariates with corresponding parameter vector $\boldsymbol{\beta}_{Ext}$. The observations with missing values serve as a reference for the zombie and external variables.

To estimate the parameters in equation (2.30) we use Ohlsson's GLMC algorithm. We fit models with all possible combinations of the company-specific covariates and include the hierarchical MLF in all models. We opt for Ohlsson's GLMC algorithm given its simplicity and computational efficiency. In comparison, GLMMs are computationally heavy, frequently experience convergence issues and are therefore not suited for exhaustive variable selection methods using this data set. Ohlsson's estimation method, however, does not allow us to calculate the mAIC since this method does not maximize the marginal likelihood. We therefore extract the AIC from the GLM fit resulting from the last iteration in Ohlsson's GLMC algorithm and we select the model with the lowest AIC. Hereby, we select the best fitting parsimonious model from a set of fitted models.

We first perform best subset regression with the set of internal covariates only and in every model, we include the hierarchical MLF. Following this, we perform best subset regression with the set of external covariates and include the selected internal covariates, ${}_{BF}x_{ijkt}$ and hierarchical MLF in all models. As such, we first identify the most important internal covariates that are readily available to the insurer and only include external covariates if they have an added predictive value. We perform this procedure once with a Gaussian model specification and identity link and once with a Tweedie model specification with a log link.

Table 2.2 shows the results of the variable selection procedure. In the internal covariates only models, the first two internal variables and `two-digit postal code binned` are selected. When external covariates are entered, `external variable 2` is selected in the Gaussian model. Conversely, with the Tweedie model specification the first and third external variable are selected.

Benchmark models. The models resulting from best subset regression allow us to examine the predictive performance when we use internally and externally collected company-specific covariates as well as a hierarchical MLF. We also fit a hierarchical credibility model and an intercept-only (G)LMM to the training set which serve as benchmark models.

Table 2.2: Results best subset regression.

		Gaussian		Tweedie	
	Predictor	Internal only	Internal + external	Internal only	Internal + external
Internal	Internal variable 1	×	×	×	×
	Internal variable 2	×	×	×	×
	Internal variable 3				
	Internal variable 4				
	Two-digit postal code binned	×	×	×	×
External	Available information Bel-First		×		×
	External variable 1				×
	External variable 2		×		
	External variable 3				×

2.3.5 Inspecting the model fits on the training set

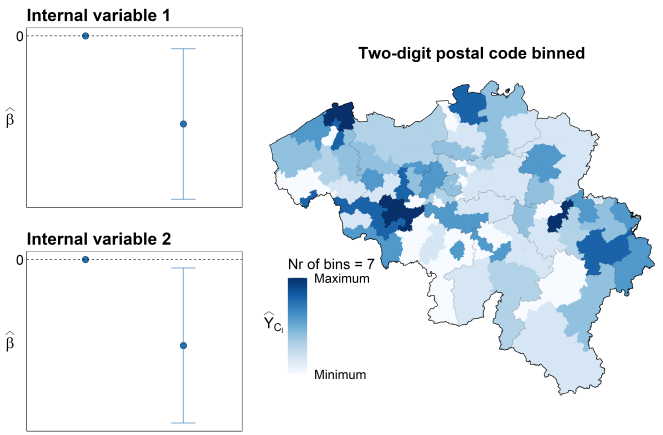
Following, we refit and refine the selected models. To evaluate the distributional assumption on the target variable and the goodness of fit, we examine the estimated effect sizes of the company-specific covariates and random effects (Section 2.3.5) as well as the fitted values on the training set (Section 2.3.6) in detail. An appropriate distributional assumption will provide a good fit with the data. In this case, the estimated values of the company-specific parameters and random effects are expected to be in line with the findings of the exploratory analysis in Section 2.3.1.

Examining the company-specific covariates We refit the models selected by best subset regression, as sketched in Section 2.3, using GLMMs and use the 95% confidence intervals (CIs) of the estimated coefficients of the company-specific covariates to refine the model. For variables with a large number of levels, we need an alternative strategy. Here, we want to reduce the number of levels. Hereto, we use the multi-type Lasso (Reynkens et al., 2018) with the Fused Lasso penalty for these variables to merge consecutive levels in a data-driven way. To account for the hierarchical structure, we use the random effect estimates of the GLMMs and specify these as offset variables in the multi-type Lasso. We include the salary mass as weight.

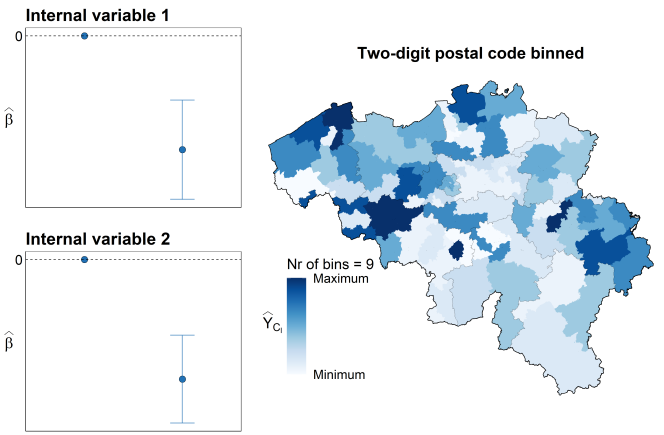
Figures 2.7(a) and 2.7(b) show the estimated coefficients of the company-specific risk factors and the 95% CIs for the internal covariates only models (i.e. the models resulting from best subset regression with only internal covariates). The direction of the estimated coefficients (positive or negative) is the same for the LMM and Tweedie GLMM. In addition, the model fits confirm the findings of our exploratory analysis. Considering that none of the confidence intervals are close to zero, no adjustments are made to the internal covariates only models.

Figure 2.7: Internal covariates only models.

(a) LMM.



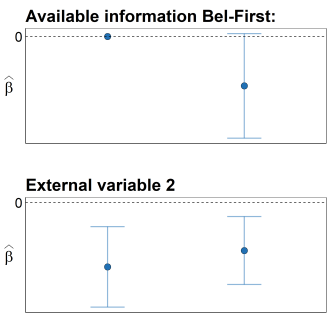
(b) Tweedie GLMM.



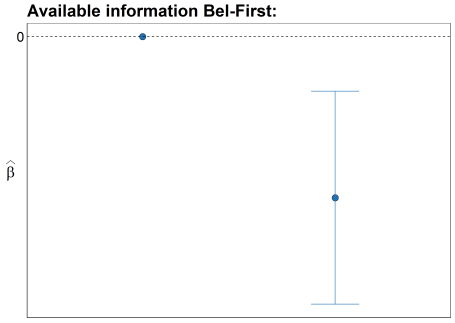
When adding external covariates to the internal covariates only LMM, `external variable 2` is selected by best subset regression. For the Tweedie GLMM, the covariates `external variable 1` and `external variable 3` are selected. For the internal and external covariates LMM, we merge two levels of `external variable 2` since the point estimates are approximately the same and the 95% CIs show a large overlap. Hence, in this model we retain the external covariates ${}_{BF}x_{ijkt}$ and `external variable 2`. Further, based on the results of the multi-type Lasso, `external variable 1` and `external variable 3` are removed from the internal and external covariates Tweedie model and ${}_{BF}x_{ijkt}$ is the only remaining external covariate in (2.30). After these adjustments, the internal and external covariate models are refit. Figures 2.8(a) and 2.8(b) show the estimated coefficients of the external covariates only.

Figure 2.8: Internal + external covariates models: coefficient estimates external covariates.

(a) LMM



(b) Tweedie GLMM.



When rounded, the estimated power parameter $\hat{p} = 1.77$ in both the internal covariates only and internal and external covariates Tweedie GLMMs, which corresponds to a claim-size distribution with mode in zero since $\hat{p} \in (1.5, 2)$ (Jørgensen and Souza, 1994). This value seems appropriate considering that our data set is characterized by a large amount of $Y_{ijkt} = 0$ (see Section 2.3.1).

Examining the random effect estimates. To examine and compare the random effect estimates across the different estimation methods, we plot the estimates obtained for the industries and branches. Figure 2.9 shows the random effect estimates of the industries for the LMMs and Tweedie GLMMs. In these plots, we add the random effect estimates of the hierarchical credibility model (see Section

2.2). For the LMMs, we use the additive Jewell model (see equation (2.6)) and for the Tweedie GLMMs, we use the multiplicative Jewell model (see equation (2.12)). We use the exponent of the random effect estimates of the Tweedie GLMMs to be able to compare these with the estimates resulting from the multiplicative Jewell model.

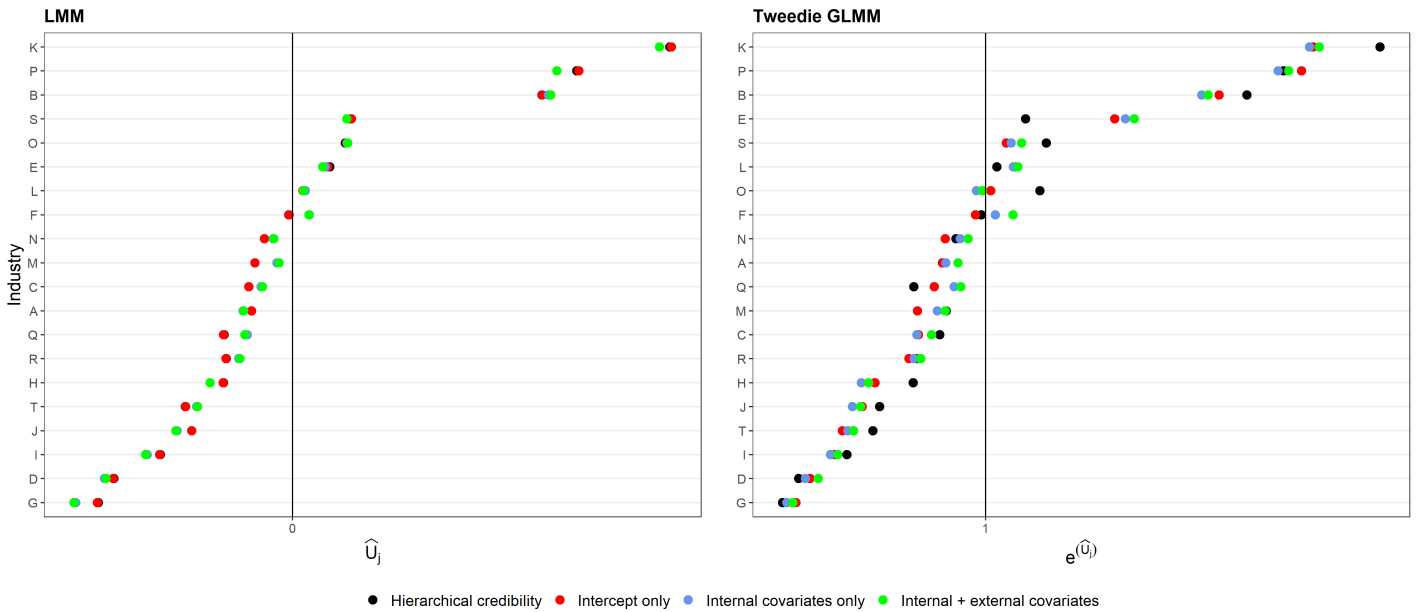
The random effect estimates of the LMMs are approximately equal to those of the additive Jewell model. In addition, the random effect estimates of the internal covariates only LMM (blue) and internal and external LMM (green) are nearly identical. This causes the estimates to overlap in the left plot of Figure 2.9. In contrast, the differences across the different estimation methods are larger for the multiplicative models. Here, we see a large difference between the random effect estimates of the Tweedie GLMMs and the random effect estimates of the multiplicative Jewell model. Comparing the random effect estimates of the industries across the different Tweedie GLMMs, we see that these are slightly higher for the internal and external covariates model compared to estimates of the intercept only and internal covariates only models. In contrast, the random effect estimates of the branches are approximately equal for all Tweedie GLMMs (see Appendix A.2). There are, however, large differences between the random effect estimates of the branches estimated by the multiplicative Jewell model and those estimated by the Tweedie GLMMs. We therefore inspect these estimates in detail for a selected group of branches, but can only give the overall conclusion due to the confidentiality of the data. We observe large contract-specific damage rates Y_{ijkt} as well as high weighted averages $\bar{Y}_{\cdot jk\cdot}$ in branches with large corresponding random effect estimates. Both the estimation method as well as the distributional assumption seem to have an impact on the random effect estimates. The results indicate that the random effect estimates of the Tweedie GLMMs are more in line with the empirical results compared to the random effect estimates of the multiplicative Jewell model.

2.3.6 Inspecting the fitted values on the training set

Next, we examine the fitted values on the company-level training set in more detail. We focus on the (Gaussian and Tweedie) internal covariates only models and we use the hierarchical credibility model as a benchmark. We compute the damage rate for an individual company as

$$Y_{hijkt} = \frac{\mathcal{Z}_{hijkt}}{w_{hijkt}} \tag{2.31}$$

Figure 2.9: Random effect estimates of the industries.



where $\mathcal{Z}_{hijkt}$ denotes the capped claim amount of the h^{th} company of tariff class i operating in branch k within industry j at time t and w_{hijkt} is the salary mass. $\widehat{Y}_{hijkt}$ stands for the fitted damage rate. To enable a detailed description of the results whilst preserving the confidentiality of the data, we multiply both Y_{hijkt} and $\widehat{Y}_{hijkt}$ with a constant.

Balance property. For insurance applications, it is crucial that the models provide us a reasonable premium volume at portfolio level. Hereto, we examine the balance property (Bühlmann and Gisler, 2006; Wüthrich, 2020) on the training set. That is,

$$\sum_{i,j,k,t} w_{ijkt}\ Y_{ijkt} = \sum_{i,j,k,t} w_{ijkt}\ \widehat{Y}_{ijkt} \tag{2.32}$$

where i serves as an index for the tariff class. GLMs fulfill the balance property when we use the canonical link (see Wüthrich (2020)). For LMMs and hence, the hierarchical credibility model this property also holds. Conversely, most GLMMs do not have this property. To regain the balance property, we introduce a quantity α

$$\alpha = \frac{\sum_{i,j,k,t} w_{ijkt}\ Y_{ijkt}}{\sum_{i,j,k,t} w_{ijkt}\ \widehat{Y}_{ijkt}} \tag{2.33}$$

which quantifies the deviation of the total predicted damage from the total observed damage. In case of the log link, we can then use α to update the intercept to $\hat{\mu} + \log(\alpha)$ to regain the balance property. We therefore update the intercept for all Tweedie GLMMs, at the level of the training data, and calculate the fitted values using the updated intercept.

Company-specific covariate levels. The internal covariates only models contain the first two internal variables and `two-digit postal code binned`. For each tariff class i, we compute the empirical weighted average of the damage rates $\bar{Y}_{\cdot i\cdots}$ and weighted average of the predictions $\bar{\widehat{Y}}_{\cdot i\cdots}$ using

$$\bar{Y}_{\cdot i\cdots} = \frac{\sum_{h,j,k,t} w_{hijkt} Y_{hijkt}}{\sum_{h,j,k,t} w_{hijkt}} \quad \text{and} \quad \bar{\widehat{Y}}_{\cdot i\cdots} = \frac{\sum_{h,j,k,t} w_{hijkt} \widehat{Y}_{hijkt}}{\sum_{h,j,k,t} w_{hijkt}}. \tag{2.34}$$

Figure 2.10 depicts the results for two different tariff classes. The plots on the left show the empirical distribution of the Y_{hijkt}'s together with the $\bar{Y}_{\cdot i\cdots}$. The plots on the right show the distribution of the $\widehat{Y}_{hijkt}$'s and the $\bar{\widehat{Y}}_{\cdot i\cdots}$ of the different models.

For the majority of the tariff classes, the predictions of the Tweedie model most closely correspond with what we observe empirically. Overall, we observe that, as the range of the Y_{hijkt}'s increases, the range of the $\widehat{Y}_{hijkt}$'s increases correspondingly. The predictions are centered at $\bar{Y}_{\cdot i\cdots}$ and $\bar{\widehat{Y}}_{\cdot i\cdots}$ is approximately equal to $\bar{Y}_{\cdot i\cdots}$. In comparison, for the LMM we have negative $\widehat{Y}_{ijkt}$'s and the predictions show a larger deviation from what we observe in the data.

Hierarchical MLF levels. To inspect the predictions at the different hierarchical MLF levels, we split the company-level training set using the hierarchical MLF. We compute the empirical weighted average of the damage rates $\bar{Y}_{\cdot\cdot jk\cdot}$ and weighted average of the predictions $\bar{\widehat{Y}}_{\cdot\cdot jk\cdot}$ using

$$\bar{Y}_{\cdot\cdot jk\cdot} = \frac{\sum_{h,i,t} w_{hijkt} Y_{hijkt}}{\sum_{h,i,t} w_{hijkt}} \quad \text{and} \quad \bar{\widehat{Y}}_{\cdot\cdot jk\cdot} = \frac{\sum_{h,i,t} w_{hijkt} \widehat{Y}_{hijkt}}{\sum_{h,i,t} w_{hijkt}}. \tag{2.35}$$

Figure 2.11 shows the results for branch D4 in industry D (associated with a low random effect of the branch and industry) and for branch P2 in industry P (associated with a high random effect for the branch and industry). As before, the Tweedie GLMM predictions most closely resemble the empirical results in most branches. The range of the $\widehat{Y}_{hijkt}$'s increases as the range of the Y_{hijkt}'s increases, $\bar{\widehat{Y}}_{\cdot\cdot jk\cdot}$ is approximately equal to $\bar{Y}_{\cdot\cdot jk\cdot}$ and the predictions are centered at $\bar{Y}_{\cdot\cdot jk\cdot}$. Furthermore, the predominant covariate pattern in a branch determines whether the average prediction of the (G)LMM is lower or higher compared to the prediction of the hierarchical credibility model. Within branch D4, for example, the majority of the observations are categorized into covariate levels that are considered to be less risky relative to the other levels. Consequently, the average prediction of the (G)LMM is lower than the prediction of the hierarchical credibility model.

2.3.7 Assessing the predictive performance

We assess the predictive performance of the pricing model on the test set, which contains damage rates of the individual companies i in the most recent year available. The empirical distribution of the damage rates Y_{ijkt} of the individual companies in the test set is shown in Figure 2.12. Panel (a) contains all Y_{ijkt}'s present in the test set and panel (b) shows the empirical distribution of the log transformed Y_{ijkt} for $Y_{ijkt} > 0$. The empirical distribution of the Y_{ijkt} in the test set is similar to the one observed when using all available data (Figure 2.2).

Figure 2.10: Distribution and weighted averages of Y_{hijkt} and $\widehat{Y}_{hijkt}$ for a selected set of tariff classes. Both Y_{hijkt} and $\widehat{Y}_{hijkt}$ are multiplied with a constant to preserve the confidentiality of the data.

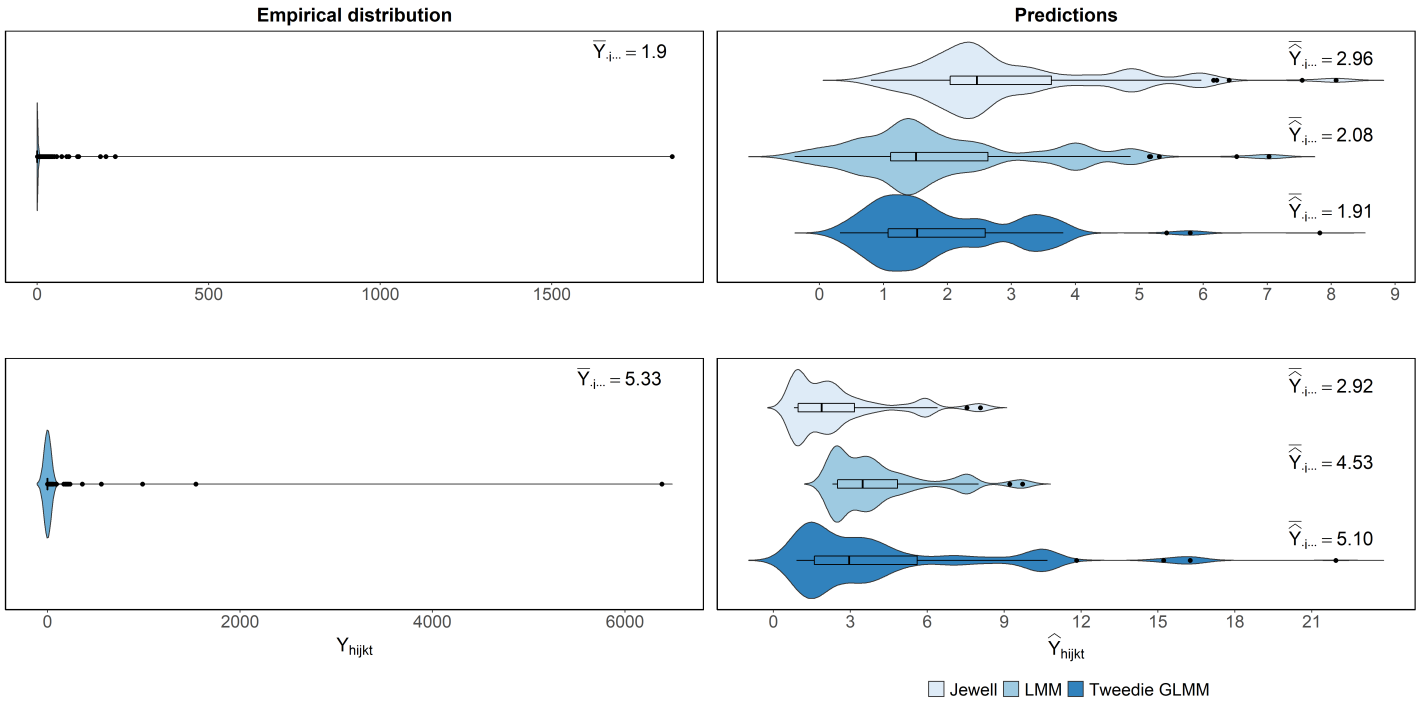
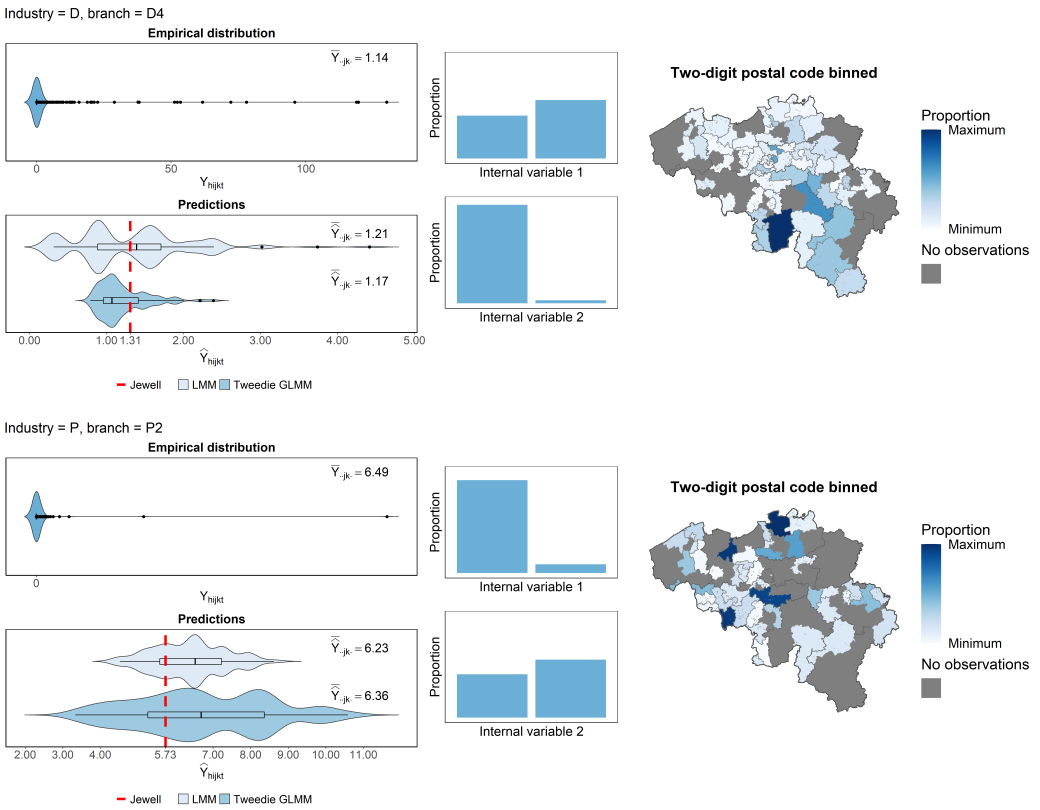


Figure 2.11: The distribution and weighted averages of Y_{hijkt} and $\widehat{Y}_{hijkt}$ for branch D4 in industry D and for branch P2 in industry P are shown on the left. The bar plots and map on the right depict the composition of the covariate levels in these branches. Both Y_{hijkt} and $\widehat{Y}_{hijkt}$ are multiplied with a constant to preserve the confidentiality of the data.



Performance measures. To assess the performance of the models, we predict the damage rates in the test set and evaluate the model predictions using the Lorenz curve (Lorenz, 1905), Gini-index (Gini, 1921) and loss ratio. The Lorenz-curve and Gini-index are considered to be appropriate tools to compare competing pricing models (Denuit, Sznajder and Trufin, 2019) and assess how well the models are able to differentiate between low- and high-risk companies. Conversely, the loss ratio gives an indication of the overall accuracy of the model predictions.

The Lorenz curve plots the cumulative percentage of the predicted damage rates against the cumulative proportion of damage rates, with the latter sorted by the predicted damage rates from high to low. An ideal Lorenz curve situates itself in the upper-left corner and indicates that it perfectly distinguishes high-risk companies from low-risk companies. The Gini-index is defined as the ratio of the area between the Lorenz curve and the line of equality (A) over the total area between the upper-left corner and the line of equality ($= 0.5$)

$$\mathrm{G} = \frac{A}{0.5}. \tag{2.36}$$

For a perfect model, we obtain the maximum theoretical value of $G = 1$. To compute the loss ratio, the total damage on the test set is computed together with the predicted damage by each of the models by transforming the individual predictions $\widehat{Y}_{ijkt}$ as follows (see equation (2.1))

$$\widehat{\mathcal{Z}}_{ijkt} = \widehat{Y}_{ijkt}\ w_{ijkt}. \tag{2.37}$$

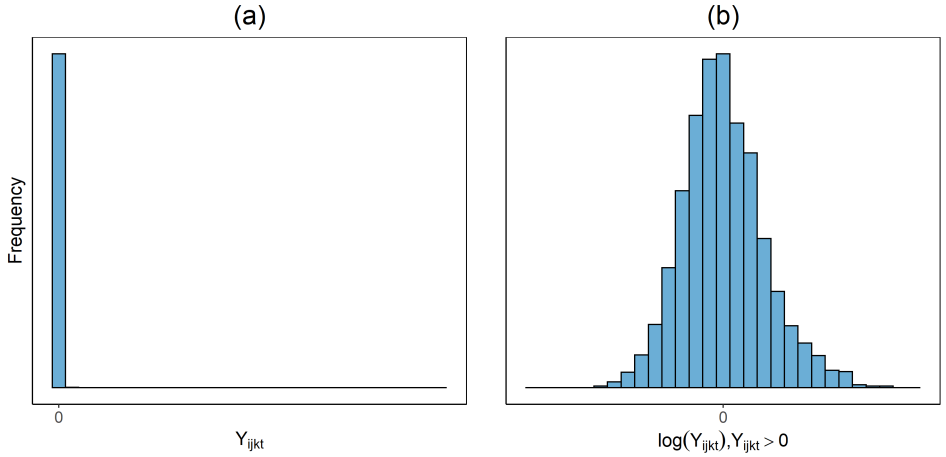
When we denote the total capped claim amount as $\mathcal{Z}_t^{tot} = \sum_{i,j,k} \mathcal{Z}_{ijkt}$ and the total predicted claim amount as $\widehat{\mathcal{Z}}_t^{tot} = \sum_{i,j,k} \widehat{\mathcal{Z}}_{ijkt}$, the loss ratio is computed as $\mathcal{Z}_t^{tot}/\widehat{\mathcal{Z}}_t^{tot}$. Next to these performance measures, we also inspect the difference in technical premium between the (G)LMMs and the hierarchical credibility model by calculating the relative difference R_{ijkt}

$$R_{ijkt} = \frac{\widehat{Y}_{ijkt}^{M} - \widehat{Y}_{ijkt}^{J}}{\widehat{Y}_{ijkt}^{J}} \tag{2.38}$$

where $\widehat{Y}_{ijkt}^{M}$ denotes the predicted pure premium by the (G)LMMs and $\widehat{Y}_{ijkt}^{J}$ the predicted pure premium by the hierarchical credibility model, which serves as the benchmark model. This allows us to identify both overpriced and underpriced policies. Compared to the hierarchically credibility model, policies are currently

overpriced when $R_{ijkt} < 0$ and can potentially be lost to competitors. Conversely, $R_{ijkt} > 0$ indicates that the policy is underpriced compared to the hierarchical credibility model and this necessitates appropriate loss control measures to prevent future financial losses.

Figure 2.12: Empirical distribution of the damage rates Y_{ijkt} of the individual companies in the test set.



Out-of-sample performance. Table 2.3 summarizes the out-of-sample performance of the models on the test set. For both the LMM and Tweedie GLMM, the Gini-index increases when company-specific risk factors are included. Consequently, by adding company-specific risk factors we are better able to distinguish high- from low-risk companies compared to when we do not include these in the model. Further, in both the LMMs and Tweedie GLMMs the model performance decreases when we add external covariates. In addition, the loss ratio of the internal and external covariates LMM is higher than the loss ratio of the internal covariates only LMM. When the external covariate ${}_{BF}x_{ijkt}$ is added to the internal covariates only Tweedie GLMM, the loss ratio shows a slight improvement. Comparing the internal covariates only LMM and internal covariates only Tweedie GLMM, we see that the predictive performance of the Tweedie model is better. The Gini-index is higher and the loss ratio is closer to one, indicating that the Tweedie GLMM is better able to differentiate between low- and high-risk companies and results in a more accurate estimation of the total damage. In addition, the loss ratio of the internal covariates

only Tweedie GLMM is lower than the loss ratio of the hierarchical credibility model.

Table 2.3: Comparison predictive performance on the test set.

Model	Distribution	Variable set	Gini-index	Loss ratio
Jewell			0.592	1.009
LMM	Gaussian	Intercept only	0.591	1.009
		Internal covariates	0.653	1.013
		Internal + external covariates	0.644	1.032
GLMM	Tweedie	Intercept only	0.607	1.010
		Internal covariates	0.660	1.007
		Internal + external covariates	0.650	1.006

Figure 2.13: Lorenz curves.

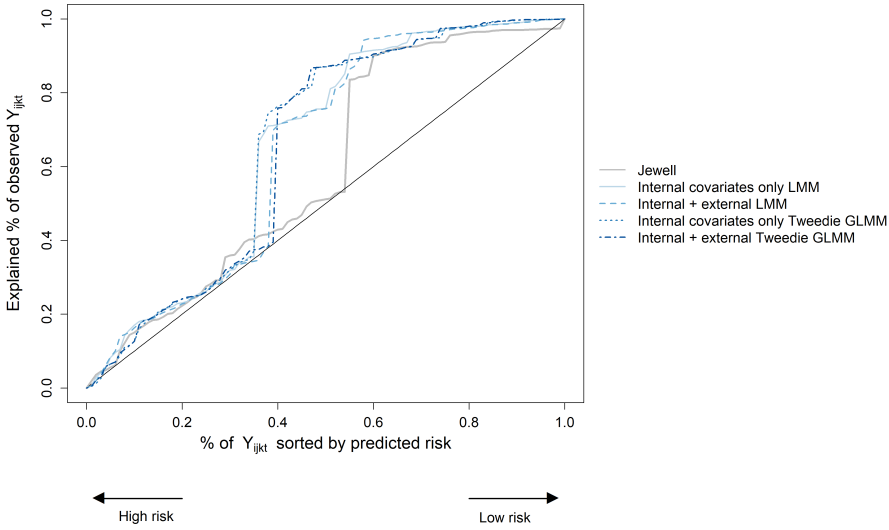


Figure 2.13 shows the Lorenz curves of the different models. The hierarchical credibility model has the lowest performance and the internal covariates only Tweedie GLMM the best performance. For all models the Lorenz curve is close to the diagonal line for observations that are considered to be high-risk. This indicates that the models experience difficulties with accurately ordering companies characterized by high $\widehat{Y}_{ijkt}$'s. Conversely, when the predicted risk decreases, the ordering of the companies gets more accurate as the Lorenz curves are further removed from the diagonal line. Consequently, all models are better able to differentiate high- from low-risk companies that have medium to low predicted damage rates.

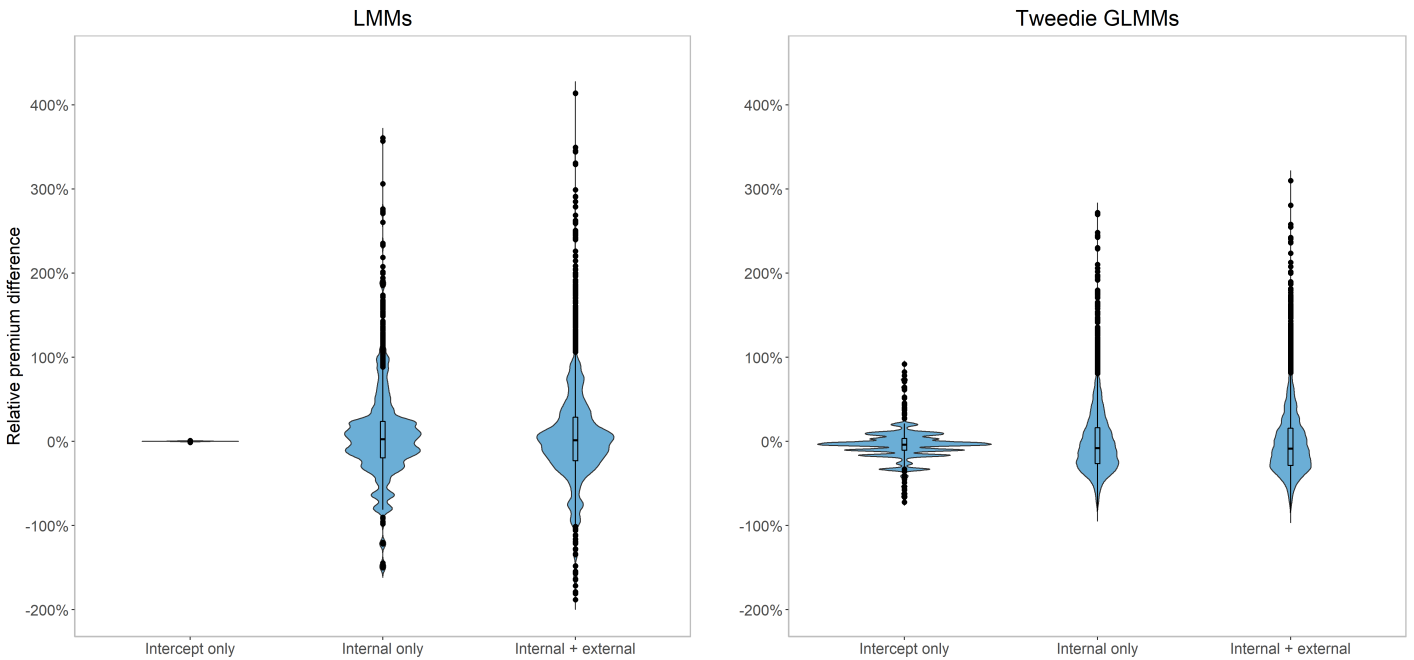
Figure 2.14 depicts the relative premium differences. The difference is negligible when using the intercept-only LMM, which is due to the equivalence between the intercept-only LMM and Jewell model. Larger differences are seen in the $\widehat{Y}_{ijkt}$'s when using the Tweedie intercept-only GLMM which is caused by larger differences in the random effect estimates. When adding company-specific risk factors to the LMM and Tweedie GLMM, the majority of the companies see a decrease in the expected pure premium. Here, the density is larger for $R_{ijkt} < 0$ indicating that $\widehat{Y}^M_{ijkt} < \widehat{Y}^J_{ijkt}$. In addition, for most companies that see an increase in the expected pure premium when company-specific risk factors are added, this will be a 50% increase of the pure premium at most and this will be more than 50% for only a few companies.

2.4 Conclusions

In this chapter, we show how random effects models can be used to construct a data-driven insurance pricing model when working with hierarchically structured data supplemented by internal and external contract-specific risk factors. We examine several random effects models previously proposed within the actuarial literature, such as the hierarchical credibility model of Jewell (1975), the combination of a GLM with the hierarchical credibility model (Ohlsson, 2008) and mixed models. We examine and compare the performance of these random effects models using a workers' compensation insurance portfolio from a Belgian insurer. In addition, we assess the effect of the distributional assumption of the response as well as the added value of contract-specific risk factors derived from an external data source.

The random effects specification allows us to efficiently estimate and quantify the effect of the different hierarchical MLF levels. Further, incorporating contract-specific information in the model results in an improved predictive performance. With regard to the estimation methods, we find that Ohlsson's iterative GLMC algorithm is ideal in combination with (exhaustive) variable selection methods. Its simplicity and computational efficiency allows for a quick estimation of the parameters. In addition, the parameter estimates can be used as starting values when fitting GLMMs. The GLMMs are computationally heavy and are prone to convergence issues. Providing appropriate starting values drastically speeds up the GLMM algorithm and frequently helps to overcome convergence issues. Given their well-developed statistical framework, which allows for statistical inference, the GLMMs are well suited to examine the model and can be used as a final estimation step to

Figure 2.14: Relative premium differences on the test set.



obtain accurate estimates. With regard to the distributional assumptions, the results indicate that the Gaussian distribution is not ideal in combination with company-specific covariates. Due to the presence of zero valued claims and the symmetric nature of the Gaussian distribution, some companies obtain a negative predicted damage rate. Conversely, the Tweedie distribution is especially suited for modeling and predicting damage rates. As previously stated by Jørgensen and Souza (1994), the Tweedie distribution handles zero valued observations in a natural, satisfactory way. Moreover, compared to the LMMs, the Tweedie GLMMs are better able to differentiate between low- and high-risk companies and result in a more accurate estimation of the total claim amount in the test set. In addition, including company-specific covariates allows for more and better differentiation between companies and the Tweedie model is possibly better able to detect groups characterized by large damage rates. Adding external company-specific covariates to the internal covariates only model, however, did not result in significant improvements.

The absence of an added value of incorporating external data may be caused by limiting ourselves to one specific external data source. Future research can examine whether this generalizes to other data sets and examine the potential predictive value of other external data sources. Further, we limit ourselves to regression-type random effects models. The theoretical framework can be extended to include random effects machine-learning methods as well, such as the RE-EM tree of Sela and Simonoff (2012). Given the promising results of machine-learning methods within actuarial applications, it may prove to be worthwhile to examine whether this generalizes to hierarchically structured data as well.

Chapter 3

On clustering levels of a hierarchical categorical risk factor

Handling nominal covariates with a large number of categories is challenging for both statistical and machine learning techniques. This problem is further exacerbated when the nominal variable has a hierarchical structure. The industry code in a workers' compensation insurance product is a prime example hereof. We commonly rely on methods such as the random effects approach (Campo and Antonio, 2023) to incorporate these covariates in a predictive model. Nonetheless, in certain situations, even the random effects approach may encounter estimation problems. We propose the data-driven Partitioning Hierarchical Risk-factors Adaptive Top-down (PHiRAT) algorithm to reduce the hierarchically structured risk factor to its essence, by grouping similar categories at each level of the hierarchy. We work top-down and engineer several features to characterize the profile of the categories at a specific level in the hierarchy. In our workers' compensation case study, we characterize the risk profile of an industry via its observed damage rates and claim frequencies. In addition, we use embeddings (Mikolov et al., 2013; Cer et al., 2018) to encode the textual description of the economic activity of the insured company. These features are then used as input in a clustering algorithm to group similar categories. We show that our method substantially reduces the number of categories and results in a grouping that is generalizable to out-of-sample data. Moreover, when estimating

the technical premium of the insurance product under study as a function of the clustered hierarchical risk factor, we obtain a better differentiation between high-risk and low-risk companies.

This chapter is based on the paper by Bavo D.C. Campo and Katrien Antonio titled "On clustering levels of a hierarchical categorical risk factor", which has been accepted for publication in Annals of Actuarial Science on the 14^{th} of November, 2023 and which is currently in press. The preprint is available at arXiv: https://arxiv.org/abs/2304.09046.

3.1 Introduction

At the heart of a risk-based insurance pricing model is a set of risk factors that are predictive of the loss cost. To model the relation between the risk factors and the loss cost, actuaries rely on statistical and machine learning techniques. Both approaches are able to handle different types of risk factors (i.e. nominal, ordinal, geographical or continuous). In this contribution we put focus on challenges imposed by nominal variables with a hierarchical structure. Such variables may cause estimation problems, due to an exceedingly large number of categories and a limited number of observations for some of the categories. Using default methods to handle these, such as dummy encoding, may result in unreliable parameter estimates in generalized linear models (GLMs) and may cause machine learning methods to become computationally intractable. We refer to this type of risk factor as a hierarchical multi-level factor (MLF) (Ohlsson and Johansson, 2010) or a hierarchical high-cardinality attribute (Micci-Barreca, 2001; Pargent et al., 2022). Examples of such a nominal variable include provinces and municipalities within provinces, or vehicle brands and models within brands. Within workers' compensation insurance, a typical example is the hierarchical MLF derived from the numerical codes of the NACE system. The NACE system is a hierarchical classification system used in the European Union to group similar companies based on their economic activity (European Commission and Eurostat, 2017). A similar example is the Australian and New Zealand Standard Industrial Classification (ANZSIC) system (Australian Bureau of Statistics and New Zealand, 2006), which is closely related to the NACE system (European Commission and Eurostat, 2017).

In predictive modelling, such risk factors are potentially a great source of information. In workers' compensation insurance, certain industries (e.g., manufacturing, construction) and occupations (e.g., labouring, roofer) are associated with an in-

creased risk of filing claims (Walters et al., 2010; Holizki et al., 2008; Wurzelbacher et al., 2021). Furthermore, companies operating in the same industry are exposed to similar risks. This creates a dependency among companies active in the same industry and heterogeneity between companies working in different industries (Campo and Antonio, 2023). Industry classification systems, such as the NACE and ANSZIC, allow to group companies based on their economic activity at varying levels of granularity. Most industry classification systems are hierarchical classifications that work top-down. The classification of a company starts at the highest level in the hierarchy and, from here, proceeds successively to lower levels in the hierarchy. At the top level of the hierarchy, the categories are broad and general, covering a wide range of economic activities. As we move down the hierarchy, the categories are broken down into increasingly specific subcategories that encompass more detailed economic activities. Further, industry classification systems typically provide a textual description for the categories at all levels in the hierarchy. This description explains why companies are grouped in the same category and can be used to judge the similarity of activities among categories.

To incorporate the hierarchical MLF in a predictive model, we can opt for the hierarchical random effects approach (Campo and Antonio, 2023). Here, we specify a random effect at each level in the hierarchy. The random effects capture the unobservable characteristics of the categories at the different levels in the hierarchy. Moreover, random effects models account for the within-category dependency and between-category heterogeneity. To estimate a random effects model, we can either use the hierarchical credibility model (Jewell, 1975), Ohlsson's combination of the hierarchical credibility model with a GLM (Ohlsson, 2008) or the mixed models framework (Molenberghs and Verbeke, 2005). These estimation procedures rely on the estimation of variance parameters and we require these estimates to be non-negative (Molenberghs and Verbeke, 2011; Oliveira et al., 2017). In some cases, however, we obtain negative variance estimates and this can occur when there is low variability (Oliveira et al., 2017) or when the hierarchical structure of the MLF is misspecified (Pryseley et al., 2011). In these situations, the estimation procedure yields nonsensical results. With the random effects approach we implicitly assume that the risk profiles differ between the different categories (Tutz and Oelker, 2017). However, it is not an unreasonable assumption that certain categories have an identical effect on the response and that these should be grouped into homogeneous clusters. Decreasing the total number of categories leads to sparser models that are easier to interpret and less likely to experience estimation problems or to overfit.

Additionally, individual categories will have more observations, leading to more precise estimates of their effect on the response.

To group data into homogeneous clusters, we typically rely on clustering techniques. These techniques partition the data points into clusters such that observations within the same cluster are more similar compared to observations belonging to other clusters (Hastie et al., 2009, Chapter 14). Within actuarial sciences, clustering methods recently appeared in a variety of applications. In motor insurance, for example, clustering algorithms are employed to group driving styles of policyholders (Wüthrich, 2017; Zhu and Wüthrich, 2021), to construct tariff classes in an unsupervised way (Yeo et al., 2001; Wang and Keogh, 2008) and to bin continuous or spatial risk factors (Henckaerts et al., 2018). Also in health insurance we find several examples. Rosenberg and Zhong (2022) used clustering techniques to identify high-cost health care utilizers who are responsible for a substantial amount of health expenditures.

Our aim is to use clustering algorithms in workers' compensation insurance pricing, to group categories of the hierarchical MLF that are similar in riskiness and economic activity. To characterize the riskiness, we rely on risk statistics such as the average damage rate and the expected claim frequency. Moreover, we also use the textual description of the categories to obtain information on the economic activity. The risk statistics are expressed as numerical values, whereas the textual descriptions are presented as nominal categories. Both types of features are then used as input in a clustering algorithm to group similar categories of the hierarchical MLF. To create clusters, most algorithms rely on distance or (dis)similarity metrics to quantify the degree of relatedness between observations in the feature space (Hastie et al., 2009; Foss et al., 2019). These metrics, however, are different for numeric and nominal features which makes it challenging to cluster mixed-type data (Cheung and Jia, 2013; Foss et al., 2019; Ahmad et al., 2019). One approach to tackle this problem is to convert the nominal to numeric features. Hereto, we commonly employ dummy encoding (Hsu, 2006; Cheung and Jia, 2013; Ahmad et al., 2019; Foss et al., 2019). This encoding creates binary variables that represent category membership. Hereby, it results in a loss of information when the categories have textual labels. Labels provide meaning to categories and reflect the degree of similarity between different categories. A more suitable encoding is obtained using embeddings, a technique developed within natural language processing (NLP). Embeddings are vector representations of textual data that capture the semantic information (Verma et al., 2021; Schomacker and Tropmann-Frick, 2021; Ferrario and Naegelin, 2020).

In the actuarial literature, Lee et al. (2020) show how embeddings can be used to incorporate textual data into insurance claims modelling. Xu et al. (2022) create embedding based risk factors that are used as features in a claim severity model. Zappa et al. (2021) used embeddings to create risk factors that predict the severity of injuries in road accidents.

Research on grouping categories of hierarchical MLFs is limited. Most of the research is focused on nominal variables that do not have a hierarchical structure. For example, the Generalized Fused Lasso (GFL) (Höfling et al., 2010; Gertheiss and Tutz, 2010; Oelker et al., 2014) groups MLF categories within a regularized regression framework. Here, categories are merged when there is a small difference between the regression coefficients. Nonetheless, the GFL does not scale well to high-cardinality features since the number of estimated coefficient differences grows exponentially with the number of categories. Another example to fuse non-hierarchically structured nominal variables is the method of Carrizosa et al. (2021). Here, the authors first specify the order of the categories and the number of clusters. Next, to create the prespecified number of clusters they group consecutive categories. For a specific number of clusters, multiple solutions exist and the solution with the highest out-of-sample accuracy is preferred. The disadvantages of this approach are three-fold. First, it only merges neighbouring categories. Second, there is no procedure to select the optimal number of groups and third, the procedure can not immediately be applied to hierarchical MLFs. To the best of our knowledge, the method described in Carrizosa et al. (2022) is the only approach that puts focus on reducing the number of categories of a hierarchical MLF. Here, the authors propose a bottom-up clustering strategy. This technique begins by considering the categories at the lowest level in the hierarchy. Hereafter, starting from the categories at the lowest level in the hierarchy, they consecutively merge categories at the lowest level into broader categories at higher levels in the hierarchy. As such, categories that are nested within the same category at a higher level in the hierarchy are grouped. Notwithstanding, their proposed optimization strategy is only suited for linear regression models. Further, it is not suitable when we want to maintain the levels in the hierarchical structure. Certain granular categories are replaced the broader categories they are nested in. Hence, it is possible that the optimal solution produces a hierarchical structure with a different depth in its representation.

This chapter contributes to the existing literature in the following ways. Firstly, we present a data-driven approach to reduce an existing granular hierarchical structure to its essence, by grouping similar categories at every level in the hierarchy.

We devise a top-down procedure, where we start at the top level, to preserve the hierarchical structure. At a specific level in the hierarchy, we engineer several features to characterize the risk profile of each category. In a case-study with a workers' compensation insurance product, we use predicted random effects obtained with a generalized linear mixed model for damage rates on the one hand and claim frequencies on the other hand. To extract the textual information contained in the category description, we use embeddings. Next, we use these features as input in a clustering algorithm to group similar categories into clusters. Hereafter, we proceed to grouping the categories at the next hierarchical level. The procedure stops once we grouped the categories at the lowest level in the hierarchy. Secondly, we provide a concise overview of important aspects and algorithms in cluster analysis. Furthermore, we demonstrate that the clustering algorithm and evaluation criterion affect the clustering solution. Thirdly, we show that embeddings can be used to group similar categories of a nominal variable. Contrary to Lee et al. (2020); Zappa et al. (2021); Xu et al. (2022), we do not employ embeddings to create new risk factors in a pricing model. Instead, we use embeddings to extract the textual information of category labels and to cluster categories based on their semantic similarities.

The remainder of this chapter is structured as follows. In Section 3.2, we use a workers' compensation insurance product as a motivating example with the NACE code as hierarchical MLF. We illustrate the structure of this type of data set, which information is typically available and we explain how to engineer features that characterize the risk profile of the categories. In Section 3.3, we define a top-down procedure to cluster similar categories at a specific level in the hierarchy and we discuss several aspects of clustering techniques. The results of applying our procedure to reduce the hierarchical structure to its essence are discussed in Section 3.4. Moreover, in this section we also compare the use of the original and the reduced structure in a technical pricing model for workers' compensation insurance. Section 4.5 concludes the article.

3.2 Feature engineering for industrial activities in a workers' compensation insurance product

To illustrate the importance of hierarchical MLFs in insurance pricing, a workers' compensation insurance product is a particularly suitable example. This insurance product compensates employees for lost wages and medical expenses resulting from

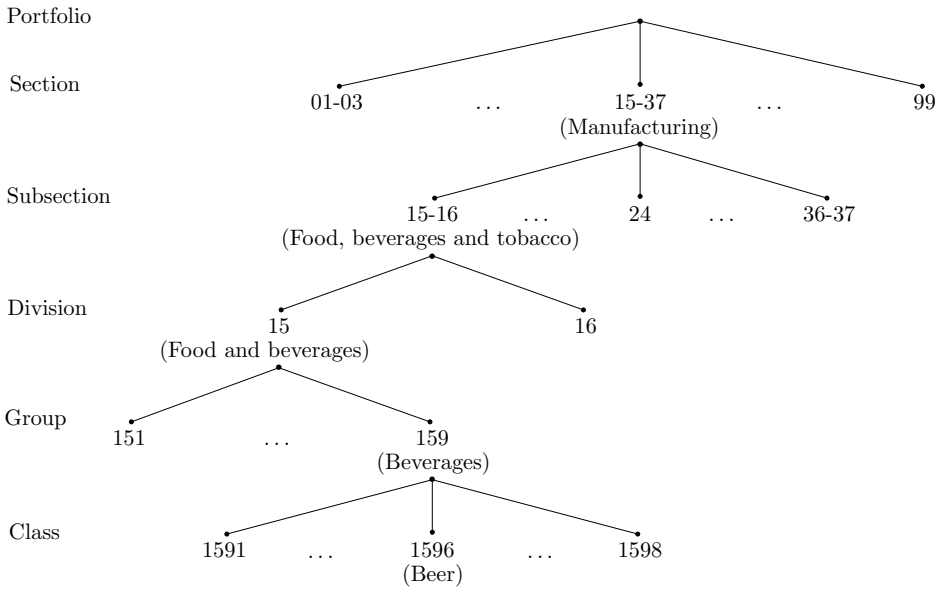
job-related injury (see Campo and Antonio (2023) for more information). In this type of insurance, we generally work with an industrial classification system to group companies based on their economic activity. Hereto, we demonstrate and discuss the NACE classification system in Section 3.2.1. We explain the typical structure of a workers' compensation insurance data set and discuss which information is available. In Section 3.2.2 we show how we use this information to engineer features that characterize the riskiness and economic activity of categories at a specific level in the hierarchy.

3.2.1 A hierarchical classification scheme for industrial activities

In a workers' compensation insurance data set, it is common to work with an industrial classification system. Within the European Union, a wide range of organizations (e.g., national statistical institutes, business and trade associations, insurance companies and European national central banks (European Commission and Eurostat, 2017; Stassen et al., 2017; European Central Bank, 2021)) work with the NACE system (European Commission and Eurostat, 2017). This is a hierarchical classification system to group companies based on their economic activity. Each company is assigned a four-digit numerical code, which is used to identify the categories at different levels in the hierarchy. The NACE system works top-down, starts at the highest level in the hierarchy and then proceeds to the lower levels. In this chapter, we work with NACE Rev. 1 (Statistical Office of the European Communities, 1996), which has five hierarchical levels (in descending order): `section`, `subsection`, `division`, `group` and `class`. Most member states of the EU have a national version which follows the same structural and hierarchical framework as the NACE. In Belgium, the national version of NACE Rev. 1 is called the NACE-Bel (2003) (FOD Economie, 2004) and adds one more level to the hierarchy by adding a fifth digit. We refer to this level as `subclass`. Insurance companies may choose to add a sixth digit to include yet another level, allowing for even more differentiation between companies. This level in the hierarchy is referred to as `tariff group` and we denote the insurance company's version as NACE-Ins.

To illustrate how the NACE system works, suppose that we have a company that manufactures beer. Using NACE Rev. 1 (Statistical Office of the European Communities, 1996), this company gets the code 1596. At the top level `section`, the first two digits (i.e. 15) classify this company into *manufacturing* (see Figure 3.1).

Figure 3.1: Illustration of the NACE system for a company that manufactures beer. This economic activity is encoded as 1596 in NACE Rev. 1. Based on the NACE code, we assign the company to a certain category at a specific level in the hierarchy. For the purpose of this illustration, we shortened the textual description of the categories.



This category contains all NACE codes that start with numbers 15 to 37. Following, the first two digits categorize the company into *manufacture of food products, beverages and tobacco* at the `subsection` level, which is nested within the `section` *manufacturing*. The category *manufacture of food products, beverages and tobacco* includes all NACE codes starting with numbers 15 and 16. At the third level in the hierarchy - `division` - the company is classified into *15 - manufacture of food products and beverages*. At the fourth level `group`, we use the first three digits to classify the company in *159 - manufacture of beverages*. Finally, at the fifth and lowest level `class`, the four digit code assigns the company to *1596 - manufacture of beer*.

This example also shows that, at all levels in the hierarchy, the NACE provides a textual description for each category. This text briefly describes the economic activity of a specific category, thereby explaining why certain companies are grouped. We illustrate this in Table 3.1. Herein, we show the textual information that is available for companies with NACE codes 1591, 1596, and 1598. This table displays

a separate column for every level in the hierarchy. The corresponding value indicates which category the codes belong to. At the `section`, `subsection`, `division` and `group` level, the NACE codes 1591, 1596, and 1598 are grouped in the same categories (i.e. `15-37`, `15-16`, `15` and `159` respectively). At the `class` level, each of the codes is assigned to a different category. The column description presents the textual information for each corresponding category. For example, at the `division` level, `15` has the description *manufacture of food products and beverages*.

Table 3.1: Illustration of the textual information for NACE codes 1591, 1596, and 1598.

Section	Subsection	Division	Group	Class	Description
15-37					Manufacturing
	15-16				Manufacture of food products, beverages and tobacco
		15			Manufacture of food products and beverages
			159		Manufacture of beverages
				1591	Manufacture of distilled potable alcoholic beverages
				...	...
				1596	Manufacture of beer
				...	...
				1598	Production of mineral waters and soft drinks

The textual description of other categories in the NACE system is similar to the example in Table 3.1. Overall, the description is brief and consists of a single word or phrase. Further, categories at higher levels in the hierarchy have a more concise description using overarching terms such as *manufacturing*. Conversely, at lower levels in the hierarchy, the descriptions are typically more detailed and extensive (e.g. *manufacture of distilled potable alcoholic beverages* and *production of mineral waters and soft drinks*).

Table 3.2: Number of unique categories per level in the hierarchy of the NACE-Bel (2003).

	`Section`	`Subsection`	`Division`	`Group`	`Class`	`Subclass`
Number of categories:						
NACE-Bel (2003)	17	31	62	224	515	800
Portfolio	17	30	56	197	398	581

3.2.1.1 Selecting levels in the hierarchy

When working with NACE-Ins, we have seven levels in the hierarchy: `section`, `subsection`, `division`, `group`, `class`, `subclass` and `tariff group`. This results

in an immense amount of categories, some of which contain few to no observations. In this chapter, we work with a NACE-Ins that is based on the NACE-Bel (2003) (FOD Economie, 2004). Table 3.2 shows the unique number of categories at each level in the hierarchy. The first row indicates how many categories there are at each level in the NACE-Bel (2003) classification system. The second row specifies how many categories are present at each level in our portfolio. Due to the confidentiality of the data, we do not disclose the number of categories at the `tariff group` level.

The more levels in the hierarchy, the more complex a tariff model becomes that incorporates this hierarchical MLF in its full granularity. Consequently, in practice, the NACE-Ins may not be used in its entirety. In our database, we have access to a hierarchical MLF designed by the insurance company that offers the workers' compensation insurance product. This structure has been created by merging similar NACE-Ins codes, based on expert judgment. This hierarchical MLF has two levels in the hierarchy, referred to as `industry` and `branch` in Campo and Antonio (2023). In this chapter, we illustrate how such a hierarchical MLF can be constructed using our proposed data-driven approach, as an alternative for the manual grouping by experts. To demonstrate our method we put focus on two selected levels in the hierarchy of NACE-Ins. This is merely for illustration purposes. Our approach is applicable with any number of levels in the hierarchy.

To align with the insurance company's hierarchical MLF, we select the `subsection` and `tariff group` level (see Figure 3.1) and aim to reduce the hierarchical structure consisting of these levels. We use $l = (1, \ldots, L)$ to index the levels in the hierarchy, where L denotes the total number of levels. In our illustration, `subsection` corresponds to the highest level $l = 1$ in the hierarchy. At $l = 1$, we index the categories using $j = (1, \ldots, J)$ where J denotes the total number of categories. The `tariff group` level represents the second level in the hierarchy. Here, we use $jk = (j1, \ldots, jK_j)$ to index the categories nested within `subsection` j. We refer to k as the child category that is nested within parent category j and K_j denotes the total number of categories nested within j. Due to confidentiality of the classification system and data, no comparisons between the company's hierarchical MLF and our proposed clustering solutions will be provided in this chapter.

3.2.2 Feature engineering

We start at the highest level in the hierarchy, `subsection`, and engineer a set of features that capture the riskiness and the economic activity of the categories. Using

these features, the clustering algorithms can identify and group similar categories at the `subsection` level. Hereafter, we proceed to the `tariff group` level and engineer the same type of features for the categories at this level in the hierarchy.

We assume that we have a workers' compensation insurance data set with historical, claim related information of the companies in our portfolio. For each company i, we have the NACE-Ins code in year t. We use this code to categorize company i in `subsection` j and `tariff group` k. In our database, we have the total claim amount Z_{ijkt}, the number of claims N_{ijkt} and the salary mass w_{ijkt} for year t and company i, a member of `subsection` j and `tariff group` k.

3.2.2.1 Riskiness

We express the riskiness of the `subsection` and `tariff group` in terms of their damage rate and their claim frequency. The higher the damage rate and claim frequency, the riskier the category. For an individual company i, the damage rate in year t is calculated as

$$Y_{ijkt} = \frac{Z_{ijkt}}{w_{ijkt}}. \tag{3.1}$$

To capture the `subsection`- and `tariff group`-specific effect on the damage rate and claim frequency, we use random effects models (Campo and Antonio, 2023). In this approach, the prediction of the category-specific random effect is dependent on how much information is available. The random effect predictions are shrunk towards zero for categories with high variability, a low number of observations or when the variability between categories is small (Breslow and Clayton, 1993; Gelman and Hill, 2017; Brown and Prescott, 2006; Pinheiro et al., 2009).

We start at the `subsection` level and model the damage rate as a function of the `subsection` using a (Tweedie generalized) linear mixed model

$$g(E[Y_{ijkt}|U_j^d]) = \mu^d + U_j^d. \tag{3.2}$$

Here, μ^d denotes the intercept and U_j^d the random effect of `subsection` j. We include the salary mass w_{ijkt} as weight. As discussed in Campo and Antonio (2023), we typically model the damage rate by assuming either a Gaussian or Tweedie distribution for the response. The U_j^d's represent the between-`subsection` variability and enable us to discern between low- and high-risk profiles. The higher U_j^d, the higher the expected damage rate for `subsection` j and vice versa. To engineer the first feature, we extract the $\widehat{U}_j^d$'s from the fitted damage rate model

(3.2). Hereafter, we fit a Poisson generalized linear mixed model

$$g(E[N_{ijkt}|U_j^f]) = \mu^f + U_j^f + \log(w_{ijkt}) \tag{3.3}$$

to assess the `subsection`'s effect on the claim frequency. μ^f denotes the intercept and U_j^f the random effect of `subsection` j. We include the log of the salary mass as an offset variable. We extract the $\widehat{U}_j^f$'s from the fitted claim frequency model (3.3) to engineer the second feature. The $\widehat{U}_j^d$'s and $\widehat{U}_j^f$'s will be combined with the features representing the economic activity to group similar categories at the `subsection` level. We index the resulting grouped categories at the `subsection` level using $j' = (1, \ldots, J')$.

Next, we engineer the features at the `tariff group` level. To capture the `tariff group`-specific effect on the damage rate, we fit a (Tweedie generalized) linear mixed model

$$g(E[Y_{ij'kt}|U_{j'}^d, U_{j'k}^d]) = \mu^d + U_{j'}^d + U_{j'k}^d \tag{3.4}$$

where the random effect $U_{j'k}^d$ represents the `tariff group`-specific deviation from $\mu^d + U_{j'}^d$. As before, we include the $w_{ij'kt}$ as weight. The $U_{j'k}^d$'s reflect the between-`tariff group` variability, after having accounted for the variability between the grouped `subsections`. $U_{j'k}^d$ quantifies the `tariff group`-specific effect of `tariff group` k, in addition to the (grouped) `subsection`-specific effect $U_{j'}^d$, on the expected damage rate. To characterize the riskiness of the categories at the `tariff group` level in terms of the claim frequency, we extend (3.3) to

$$g(E[N_{ij'kt}|U_{j'}^f, U_{j'k}^f]) = \mu^f + U_{j'}^f + U_{j'k}^f + \log(w_{ij'kt}). \tag{3.5}$$

In this model, $U_{j'k}^f$ represents the `tariff group`-specific deviation from $\mu^d + U_{j'}^f$. We extract the $\widehat{U}_{j'k}^d$'s and $\widehat{U}_{j'k}^f$'s from the fitted damage rate (3.4) and claim frequency model (3.5) to engineer the features at the `tariff group` level.

We do not include any additional covariates in the random effects models (see equations Eqs. (3.2)–(3.5)) to fully capture the variability that is due to heterogeneity between categories. If, however, there are covariates available at the `subsection` or `tariff group` level, these can be incorporated in the model specification in equations Eqs. (3.2)–(3.5) and used further on in the construction of the feature matrix at the `subsection` or `tariff group` level.

The approach to construct features based on the random effect predictions is closely related to target encoding (Micci-Barreca, 2001). In target encoding, the

numerical value of a category is the weighted average of the category-specific average of the response variable and the response variable's average at the higher levels in the hierarchy. Micci-Barreca (2001) presents various approaches to determine the weights. A linear mixed model can be seen as a special case of the weights specification as it has a closed-form solution for the random effects predictions. For most GLMMs, however, there is no available analytical expression. In these cases, there is no strict equivalence with target encoding.

3.2.2.2 Economic activity

Next, we require a feature that expresses similarity in economic activity. Given that industry codes of similar activities tend to be closer, we might rely on their numerical values. The closer the numerical value, the more overlap there will be between categories at a specific level in the hierarchy. Hence, we could use the four-digit NACE codes as discussed in Section 3.2.1. Notwithstanding, not all categories can readily be converted to a numerical format. At the higher levels in the hierarchy we have categories that encompass various NACE codes, such as *manufacture of pulp, paper and paper products: publishing and printing* at the `subsection` level. This specific `subsection` consists of all NACE codes that start with numbers 21 to 22. To obtain a numerical representation of this `subsection` we can, for example, take the mean of these NACE codes. We then obtain the encodings as illustrated in Table 3.3. The column `subsection` indicates which category the NACE codes are appointed to at the `subsection` level and the column description gives the textual description of this category. The examples in this table highlight two issues with this approach. Firstly, the numerical distance may not reflect the similarity between economic activities. The difference between *manufacture of wood and wood products* and *manufacture of pulp, paper and paper products: publishing and printing* is the same as the difference between *manufacture of pulp, paper and paper products: publishing and printing* and *manufacture of coke, refined petroleum products and nuclear fuel*. Hence, using this encoding would imply that *manufacture of pulp, paper and paper products: publishing and printing* is as similar to *manufacture of wood and wood products* as it is to *manufacture of coke, refined petroleum products and nuclear fuel*. Secondly, there might be gaps between consecutive codes in the classification system. In NACE Rev. 1, for example, there are no codes that start with 43 or 44. Gaps such as these can be present at various levels in the hierarchy and generally exist to allow for future additional categories (European Commission

and Eurostat, 2017).

Table 3.3: Illustration of a possible encoding for the categories at the `subsection` level of the NACE Rev. 1.

`Subsection`	Encoding	Description
...	...	...
20	20	Manufacture of wood and wood products
21-22	21.5	Manufacture of pulp, paper and paper products: publishing and printing
23	23	Manufacture of coke, refined petroleum products and nuclear fuel
24	24	Manufacture of chemicals, chemical products and mad-made fibres
...	...	...
36-37	36.5	Manufacturing not elsewhere classified
40-41	40.5	Electricity, gas and water supply
45	45	Construction
...	...	...

Alternatively, we use embeddings to encode the economic activity of a category (Mikolov et al., 2013). Embeddings map the textual information to a continuous vector. Hence, we use embeddings to map the description for `subsection` j to a vector $\boldsymbol{e}_j = (e_{j1}, e_{j2}, \ldots, e_{jE})$. For `tariff group` k within `subsection` j, we denote the embedding vector as $\boldsymbol{e}_{jk} = (e_{jk1}, e_{jk2}, \ldots, e_{jkE})$. Here, E is the dimension of the vector which depends on the encoder. The textual information from similar categories is expected to lie closer in the vector space and this enables us to group semantically similar texts (i.e. texts that are similar in meaning).

Consequently, we group categories with comparable economic activities based on the embeddings of their textual labels. To engineer these embeddings, we may train an encoder on a large corpus of text. Within the area of Natural Language Processing (NLP), researchers commonly use neural networks (NN) as encoders. By training the NN on large amounts of unstructured text data, it is able to learn high-quality vector representations (Mikolov et al., 2013). In addition, the encoders can be trained to either learn vector representations for words, phrases or paragraphs. The disadvantage of these models is that they generally need a large amount of data (Arora et al., 2020; Troxler and Schelldorfer, 2022). Furthermore, words that do not appear often are poorly represented (Luong et al., 2013). We therefore prefer pre-trained encoders, which are trained on large text corpuses, to encode the textual description of a `subsection`. For example, the universal sentence encoder of Cer et al. (2018) is trained on Wikipedia, web news, web question-answer pages

and discussion forums. This encoder is publicly available via TensorFlow Hub (`https://tfhub.dev/`).

3.2.2.3 Feature matrix

After engineering the features at level l in the hierarchy, we assemble them into a feature matrix $\mathcal{F}_l$. Table 3.4 shows an example of the feature matrix $\mathcal{F}_1$ at the `subsection` level. Herein, ${}_1\widehat{\boldsymbol{U}}^d = (\widehat{U}_1^d, \ldots, \widehat{U}_j^d, \ldots, \widehat{U}_J^d)$ represents the vector with the predicted random effects from the fitted damage rate GLMM (see (3.2)). ${}_1\widehat{\boldsymbol{U}}^f = (\widehat{U}_1^f, \ldots, \widehat{U}_j^f, \ldots, \widehat{U}_J^f)$ denotes the vector with the predicted random effects of the claim frequency GLMM (see (3.3)) and we use the notation $\boldsymbol{e}_{\star 1} = (e_{11}, \ldots, e_{j1}, \ldots, e_{J1})$ for the embeddings. Each row in $\mathcal{F}_1$ corresponds to a numerical representation of a specific `subsection` j, with $j = (1, \ldots, J)$. We denote the feature vector of `subsection` j by $\boldsymbol{x}_j = (\widehat{U}_j^d, \widehat{U}_j^f, \boldsymbol{e}_j)$.

Table 3.4: Feature matrix $\mathcal{F}_1$, consisting of the engineered features for the categories at $l = 1$ in the hierarchy. The columns ${}_1\widehat{\boldsymbol{U}}^d$ and ${}_1\widehat{\boldsymbol{U}}^f$ contain the predicted random effects of the damage rate and claim frequency GLMM, respectively. The embedding vector is represented by the values in columns $\boldsymbol{e}_{\star 1}, \boldsymbol{e}_{\star 2}, \ldots, \boldsymbol{e}_{\star E}$.

`Subsection`	${}_1\widehat{\boldsymbol{U}}^d$	${}_1\widehat{\boldsymbol{U}}^f$	$\boldsymbol{e}_{\star 1}$	$\boldsymbol{e}_{\star 2}$	$\boldsymbol{e}_{\star 3}$	$\ldots$	$\boldsymbol{e}_{\star E}$
1	-1.25	-0.25	-2.13	1.25	0.15	$\ldots$	-0.05
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
J	0.75	0.15	1.79	-2.13	0.5	$\ldots$	1.07

At the `tariff group` level, we gather the features in $\mathcal{F}_2$. An example hereof is given in Table 3.5. ${}_2\widehat{\boldsymbol{U}}^d = (\widehat{U}_{11}^d, \ldots, \widehat{U}_{jk}^d, \ldots, \widehat{U}_{JK_J}^d)$ and ${}_2\widehat{\boldsymbol{U}}^f = (\widehat{U}_{11}^f, \ldots, \widehat{U}_{jk}^f, \ldots, \widehat{U}_{JK_J}^f)$ denote the vectors of the `tariff group`-specific random effects. To denote the embeddings, we use $\boldsymbol{e}_{\star\star 1} = (e_{111}, \ldots, e_{jk1}, \ldots, e_{JK_J 1})$. $\boldsymbol{x}_{jk} = (\widehat{U}_{jk}^d, \widehat{U}_{jk}^f, \boldsymbol{e}_{jk})$ denotes the feature vector of `tariff group` k, nested within `subsection` j.

Table 3.5: Feature matrix $\mathcal{F}_2$, consisting of the engineered features for the categories at $l = 2$ in the hierarchy. The columns ${}_2\widehat{\boldsymbol{U}}^d$ and ${}_2\widehat{\boldsymbol{U}}^f$ contain the predicted random effects of the damage rate and claim frequency GLMM, respectively. The embedding vector is represented by the values in columns $\boldsymbol{e}_{\star\star 1}, \boldsymbol{e}_{\star\star 2}, \ldots, \boldsymbol{e}_{\star\star E}$.

Subsection	Tariff group	${}_2\widehat{\boldsymbol{U}}^d$	${}_2\widehat{\boldsymbol{U}}^f$	$\boldsymbol{e}_{\star\star 1}$	$\boldsymbol{e}_{\star\star 2}$	$\boldsymbol{e}_{\star\star 3}$	...	$\boldsymbol{e}_{\star\star E}$
1	11	-1.55	-0.01	-0.54	1.08	2.12	...	0.10
...	...	...	...	...	...	...	...	...
1	$1K_1$	0.15	0.96	-0.37	-0.26	0.58	...	-0.99
...	...	...	...	...	...	...	...	...
j	$j1$	-0.29	-0.41	-0.05	-0.72	0.41	...	-0.73
...	...	...	...	...	...	...	...	...
j	jK_j	0.11	0.26	0.16	0.69	0.87	...	1.59
...	...	...	...	...	...	...	...	...
J	$J1$	0.11	0.26	0.16	0.69	0.87	...	1.59
...	...	...	...	...	...	...	...	...
J	JK_J	0.67	-1.74	0.19	0.45	0.5	...	-0.61

3.3 Clustering levels in a hierarchical categorical risk factor

3.3.1 Partitioning Hierarchical Risk-factors Adaptive Top-down

To group similar categories at each level in the hierarchy, we devise the PHiRAT algorithm to **P**artition **Hi**erarchical **R**isk-factors in an **A**daptive **T**op-down way, see Algorithm 2. We introduce some additional notation to explain how PHiRAT works. $\mathcal{J}_l$ denotes the set of categories at a specific level l in the hierarchy. Hence, when the total number of levels $L = 2$, $\mathcal{J}_1 = (1, \ldots, J)$ and $\mathcal{J}_2 = (11, \ldots, 1K_1, \ldots, j1, \ldots, jK_j, \ldots, J1, \ldots, JK_J)$ as illustrated in Tables 3.4 and 3.5, respectively. We use $\pi(c) = p$ to indicate that child category c has parent category p and $\{c : \pi(c) = p\}$ denotes the set of all child categories c nested within parent category p. $\mathcal{F}_{l,\{c:\pi(c)=p\}}$ represents the subset of feature matrix $\mathcal{F}_l$, which contains only those rows of $\mathcal{F}_l$ that correspond to the child categories nested in parent category p. For example, when $L = 2$, $\mathcal{F}_{2,\{c:\pi(c)=j\}}$ contains only the rows corresponding to the $(j1, \ldots, jK_j)$ child categories of j (see Table 3.5). Further, $\mathcal{K}_l$ denotes the number of clusters at level l in the hierarchy. For $l > 1$, we extend the notation to $\mathcal{K}_{l,\{c:\pi(c)=p\}}$ to indicate that, at level l in the hierarchy, we group the

child categories of p into $\mathcal{K}_{l,\{c:\pi(c)=p\}}$ clusters. The subscript l indicates at which level in the hierarchy we are. We use the additional subscript $\{c : \pi(c) = p\}$ to specify that we only consider the set of child categories of parent category p.

PHiRAT works top-down, starting from the highest level ($l = 1$) and working its way down to the lowest level ($l = L$) in the hierarchy (see Algorithm 2). Every iteration consists of three steps. First, we engineer features for the categories in $\mathcal{J}_l$. Second, we combine these features in a feature matrix $\mathcal{F}_l$. Third, we employ clustering techniques to group the categories in $\mathcal{J}_l$ using (a subset of) $\mathcal{F}_l$ as input. In most clustering methods, we define a tuning grid and perform a grid search to determine the optimal number of clusters. Consequently, the minimum value within the tuning grid sets the lower bound for the number of grouped categories. Further, the third step differs slightly when $l = 1$. At $l = 1$, we use the full feature matrix as input in the clustering algorithm. Conversely, when $l > 1$, we loop over the parent categories in $\mathcal{J}_{l-1}$ and in every loop, we use a different subset of $\mathcal{F}_l$ as input. For parent category p, we only consider $\{c : \pi(c) = p\}$ and use $\mathcal{F}_{l,\{c:\pi(c)=p\}}$ as input in the clustering algorithm. Hereby, we ensure that we only group child categories nested within parent category p. Further, we remove a specific level l from the hierarchy if $\mathcal{J}_l$ is reduced to $\mathcal{J}_{l-1}$ (i.e. all child categories, of each parent category in $\mathcal{J}_{l-1}$, are merged into a single group). The algorithm stops when the clustering at level $l = L$ is done.

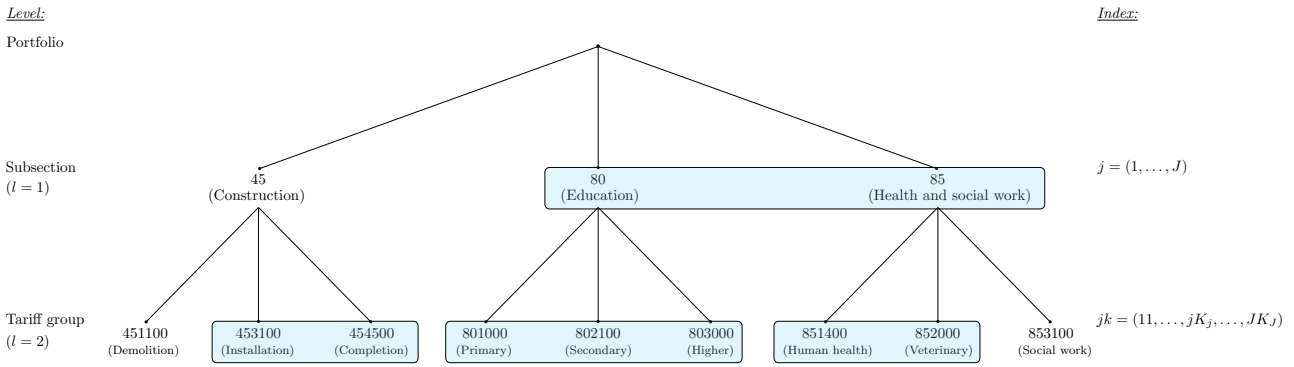
Algorithm 2: PHiRAT Pseudo-code

```
for l = 1 to L do
    Engineer features that characterize the categories;
    Combine the features in a feature matrix F_l;
    if l = 1 then
        Use a clustering algorithm to group the (1, ..., J) categories into K_1
         clusters, with F_1 as input;
    else
        foreach p in J_{l-1} do
            Use a clustering algorithm to group the {c : π(c) = p} child
             categories of parent category p into K_{l,{c:π(c)=p}} clusters, using
             F_{l,{c:π(c)=p}} as input;
        end
end
```

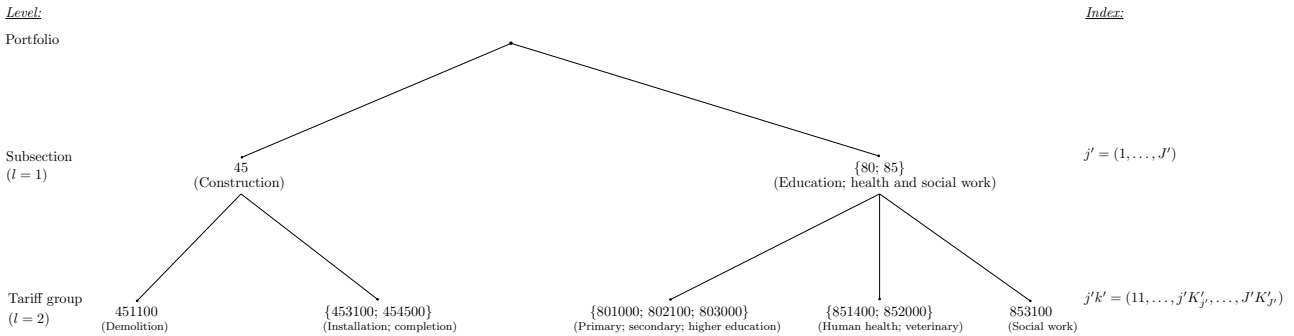
We visualize how the procedure works in Figure 3.2. In this fictive example, we focus on the `subsection` and `tariff group` level. At the `subsection` level, there

Figure 3.2: A fictive example illustrating how the PHiRAT algorithm clusters categories at the `subsection` and `tariff group` level. The textual labels of the categories are shortened for the purpose of this illustration.

(a) Visualization of the hierarchically structured categories at the `subsection` and `tariff group` level before clustering. The blue rectangles depict which categories are grouped when employing the PHiRAT algorithm.



(b) Visualization of the reduced hierarchical structure after clustering with the PHiRAT algorithm.



are three unique categories $\mathcal{J}_1 = (45, 80, 85)$ and at the `tariff group` level, we have nine unique categories $\mathcal{J}_2 = (451100, 453100, 454500, 801000, 802100, 803000, 851400, 852000, 853100)$ (see Figure 3.2(a)). We use the index $j = (1, \ldots, J)$ at $l = 1$. At $l = 2$, we use $jk = (j1, \ldots, jK_j)$ to index the categories nested within j. Using PHiRAT, we first group the categories 80 and 85 at $l = 1$. This is depicted in Figure 3.2(a) by the blue rectangle. Consequently, $\mathcal{J}_1 = (45, \{80; 85\})$ where $\{80; 85\}$ denotes that categories 80 and 85 are merged. The grouped categories at $l = 1$ are now indexed by $j' = (1, \ldots, J')$. Hereafter, the algorithm iterates over the (fused) categories in $\mathcal{J}_1$ and clusters the child categories at $l = 2$. Within `subsection` 45, it groups the categories 453100 and 454500. Within $\{80; 85\}$, the clustering results in three groups of categories: (1) $\{801000; 802100; 803000\}$; (2) $\{851400; 852000\}$ and (3) 853100. We index the fused categories at $l = 2$ and nested within j' using $j'k' = (j'1, \ldots, j'K'_{j'})$. At this point the algorithm stops and Figure 3.2(b) depicts the clustering solution.

We opt for a top-down approach for several reasons. Firstly, at the highest level we have more observations available for the categories. The more data, the more precise the category-specific risk estimates will be. Conversely, categories at more granular levels have fewer observations, leading to less precise risk estimates. Secondly, we preserve the original hierarchical structure and maintain the parent-child relationship between categories at different levels in the hierarchy. Hereby, we divide the grouping of categories, at a specific level in the hierarchy, into smaller and more specific separate clustering problems.

Alternatively, it might be interesting to construct a similar algorithm that works bottom-up and groups child categories that have different parent categories. We leave this as a topic for future research.

3.3.2 Clustering analysis

To partition the set of categories $\mathcal{J}_l$ into homogeneous groups, we rely on clustering algorithms using (a subset of) $\mathcal{F}_l$ as input. At $l = 1$, for example, we employ clustering to divide the rows in $\mathcal{F}_1$ into $\mathcal{K}_1$ homogeneous groups such that categories in each cluster j' are more similar to each other compared to categories of other clusters $j' \neq j'$. In the remainder of this section, we continue with the example of grouping the $(1, \ldots, J)$ categories at $l = 1$ to explain and illustrate the key concepts of the clustering methods used in this chapter.

Most clustering algorithms rely on distance or (dis)similarity metrics to quantify

the proximity between observations. Table 3.6 gives an overview of a selected set of (dis)similarity measures relevant to this chapter. A dissimilarity measure $d(\boldsymbol{x}_j, \boldsymbol{x}_{j'})$ expresses how different two observations $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j'}$ are (the higher the value, the more they differ). Dissimilarity metrics that satisfy the triangle inequality $d(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) \leqslant d(\boldsymbol{x}_j, \boldsymbol{x}_z) + d(\boldsymbol{x}_z, \boldsymbol{x}_{j'})$ for any z (Schubert, 2021; Phillips, 2021) are considered proper distance metrics. Most dissimilarity metrics can easily be converted to a similarity measure $s(\cdot, \cdot)$ which expresses how comparable two observations are, with similar observations obtaining higher values for these measures. The most commonly used dissimilarity metric is the squared Euclidean distance $\|\boldsymbol{x}_j - \boldsymbol{x}_{j'}\|_2^2$. Here, $\|\boldsymbol{x}_j\|_2 \coloneqq \sqrt{x_{j1}^2 + \cdots + x_{jn_f}^2}$ and n_f denotes the number of features considered. Euclidean based (dis)similarity measures, however, are not appropriate to capture the similarities between embeddings (Kogan et al., 2005). Within NLP, the cosine similarity is therefore most often used to measure the similarity between embeddings (Mohammad and Hirst, 2012; Schubert, 2021). Notwithstanding, the cosine similarity ranges from -1 to 1 and in cluster analysis we generally require the (dis)similarity measure to be non-negative (Everitt et al., 2011; Kogan et al., 2005; Hastie et al., 2009). In this case, we can convert the cosine similarity to the angular similarity which is restricted to [0, 1]. The angular similarity can be converted to the angular distance, which is a proper distance metric. Conversely, the cosine dissimilarity is not a distance measure since it does not satisfy the triangle inequality. In our study, we therefore rely on the angular similarity and angular distance to group comparable categories.

Table 3.6: Overview of existing (dis)similarity metrics to quantify the proximity between observations. We select the angular similarity and angular distance, as they are better suited to measure the similarity between embeddings and also compatible with clustering algorithms.

	Dissimilarity	Similarity
Euclidean[a]	$\|\boldsymbol{x}_j - \boldsymbol{x}_{j'}\|_2^2$	$\exp\left(\frac{-\|\boldsymbol{x}_j - \boldsymbol{x}_{j'}\|_2^2}{\sigma^2}\right)$
Cosine	$1 - \frac{\boldsymbol{x}_j^\top \boldsymbol{x}_{j'}}{\|\boldsymbol{x}_j\|_2 \cdot \|\boldsymbol{x}_{j'}\|_2}$	$\frac{\boldsymbol{x}_j^\top \boldsymbol{x}_{j'}}{\|\boldsymbol{x}_j\|_2 \cdot \|\boldsymbol{x}_{j'}\|_2}$
Angular	$\pi^{-1} \cos^{-1}\left(\frac{\boldsymbol{x}_j^\top \boldsymbol{x}_{j'}}{\|\boldsymbol{x}_j\|_2 \cdot \|\boldsymbol{x}_{j'}\|_2}\right)$	$1 - \pi^{-1} \cos^{-1}\left(\frac{\boldsymbol{x}_j^\top \boldsymbol{x}_{j'}}{\|\boldsymbol{x}_j\|_2 \cdot \|\boldsymbol{x}_{j'}\|_2}\right)$

[a] σ is a scaling parameter set by the user (Ng et al., 2001; Poon et al., 2012)

Using a selected (dis)similarity metric, we compute the proximity measure between all pairwise observations in the matrix with input features. We combine all values in a $J \times J$ similarity matrix S or dissimilarity matrix D, which is used as input in a clustering algorithm. Most algorithms require these matrices to be symmetric (Hastie et al., 2009). In literature on unsupervised learning algorithms, clustering methods are typically divided into three different types: combinatorial algorithms, mixture modelling and mode seeking (Hastie et al., 2009). Both the mixture modelling and mode seeking algorithms rely on probability density functions. Conversely, the combinatorial algorithms do not rely on an underlying probability model and work directly on the data. We opt for a distribution-free approach and therefore focus on the combinatorial algorithms summarized in Table 3.7. For the interested reader, a detailed overview of these algorithms is given Appendix B.2.

In most clustering techniques, the number of clusters $\mathcal{K}$ can be considered a tuning parameter that needs to be carefully chosen from a range of possible (integer) values. Hereto, we require a cluster validation index to select the value for $\mathcal{K}$ which results in the most optimal clustering solution. We divide the cluster validation indices into two groups: internal and external (Liu et al., 2013; Everitt et al., 2011; Wierzchoń and Kłopotek, 2019; Halkidi et al., 2001). Using external validation indices, we evaluate the clustering criterion with respect to the true partitioning (i.e. the actual assignment of the observations to different groups is known). Conversely, we rely on internal validation indices when we do not have the true cluster label at our disposal. With such indices, we evaluate the compactness and separation of a clustering solution. The compactness indicates how dense the clusters are and compact clusters are characterized by observations that are similar and close to each other. Clusters are well separated when observations belonging to different clusters are dissimilar and far from each other. Consequently, we employ internal validation indices to choose the value for $\mathcal{K}$ which results in compact clusters that are well separated (Liu et al., 2013; Everitt et al., 2011; Wierzchoń and Kłopotek, 2019).

Several internal validation indices exist and each index formalizes the compactness and separation of the clustering solution differently. An extensive overview of internal (and external) validation indices is given in Liu et al. (2013), Wierzchoń and Kłopotek (2019) and in the benchmark study of Vendramin et al. (2010). The authors concluded that the silhouette and Caliński-Harabasz (CH) indices are superior compared to other validation criteria. While these indices are well-known within cluster analysis (Wierzchoń and Kłopotek, 2019; Govender and Sivakumar, 2020; Vendramin et al., 2010), the results of Vendramin et al. (2010) do not necessarily generalize to our data

Table 3.7: Overview of clustering algorithms, together with their strengths and drawbacks.

Algorithm	**Strengths**	**Drawbacks**
k-means (MacQueen et al., 1967)	- Well-known - Simple and easy to implement - Computationally efficient	- Only suited for numeric features - Sensitive to outliers and the initialization - Local optima
k-medoids (Kaufman and Rousseeuw, 1990*b*)	- Applicable to any feature type - Less sensitive to outliers	- Sensitive to the initialization - Local optima
Spectral clustering (Hastie et al., 2009)	- Applicable to any feature type - Less sensitive to outliers, initialization and local optima - Able to identify non-convex clusters[a]	- Computationally expensive - Sensitive to the employed similarity metric
HCA[b] (Hastie et al., 2009)	- Applicable to any feature type - Less sensitive to outliers[c], initialization and local optima	- Computationally expensive - Static; divisions or fusions of clusters are irrevocable

[a] For every pair of points inside a convex cluster, the connecting straight line segment is within this cluster
[b] Hierarchical clustering analysis
[c] When using the single-linkage criterion

set. We therefore include two additional, commonly used criteria: the Dunn index and Davies-Bouldin index. Table 3.8 shows how these four indices are calculated. For the Caliński-Harabasz, Dunn and silhouette index, higher values are associated with a better clustering solution. Conversely, for the Davies-Bouldin index, we arrive at the best partition by minimizing this criterion. A more in-depth discussion of these criteria can be found in Appendix B.3.

Numerous papers compared the performance of different clustering algorithms using different data sets and various evaluation criteria (Mangiameli et al., 1996; Costa et al., 2004; de Souto et al., 2008; Kinnunen et al., 2011; Jung et al., 2014; Kou et al., 2014; Rodriguez et al., 2019; Murugesan et al., 2021). They concluded that none of the considered algorithms consistently outperforms the others and that the performance is dependent on the type of data (Hennig, 2015; McNicholas, 2016*a*,*b*; Murugesan et al., 2021). The authors advise to compare different clustering methods using more than one performance measure. Consequently, we test the PHiRAT algorithm (see Algorithm 2) with all possible combinations of the selected clustering methods (i.e. k-medoids, spectral clustering and HCA, see Table 3.7) and internal validation criteria (i.e. Caliński-Harabasz index, Davies-Bouldin index, Dunn index and silhouette index, see Table 3.8).

3.4 Clustering NACE codes in a workers' compensation insurance product

We use a workers' compensation insurance data set from a Belgian insurer to illustrate the PHiRAT procedure. The portfolio consists of Belgian companies that are active in various industries and occupations. The database contains claim-related information, such as the number of claims and the claim sizes, over a course of eight years. Additionally, for each of the companies we have the corresponding NACE-Ins code (see Section 3.2). In this section, we demonstrate how to reduce the NACE-Ins to its essence using PHiRAT. Our end objective is to incorporate the reduced version of the hierarchical MLF as a risk factor in a technical pricing model as discussed in Campo and Antonio (2023). We therefore also assess the effect of clustering on the predictive accuracy. Furthermore, if the reduced structure properly captures the essence, the predictive accuracy should generalize to out-of-sample and out-of-time data. We employ PHiRAT using the training data set, which contains data from the first seven years, to construct the reduced hierarchical risk factor. To examine

Table 3.8: Internal clustering validation criteria used in this chapter.

Criterion	Definition
Caliński-Harabasz index (Caliński and Harabasz, 1974)	$\frac{\sum_{j'=1}^{J'} n_{j'} \|\boldsymbol{c}_{j'} - \boldsymbol{c}\|_2^2/(J'-1)}{\sum_{j'=1}^{J'} \sum_{\boldsymbol{x}_j \in C_{j'}} \|\boldsymbol{x}_j - \boldsymbol{c}_{j'}\|_2^2/(J-J')}$ where $\boldsymbol{c} = \frac{1}{J}\sum_{j=1}^{J} \boldsymbol{x}_j$
Davies-Bouldin index (Davies and Bouldin, 1979)	$\frac{1}{J'}\sum_{j'=1}^{J'} \max_{j', j' \neq j'} \left(\frac{\frac{1}{n_{j'}}\sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'}) + \frac{1}{n_{j'}}\sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'})}{d(\boldsymbol{c}_{j'}, \boldsymbol{c}_{j'})} \right)$
Dunn index (Dunn, 1974)	$\min_{1 \leqslant j' \leqslant J'} \left(\min_{\substack{1 \leqslant j' \leqslant J' \\ j \neq j}} \left(\frac{\min_{\substack{\boldsymbol{x}_j \in C_{j'} \\ \boldsymbol{x}_j \in C_{j'}}} d(\boldsymbol{x}_j, \boldsymbol{x}_j)}{\max_{1 \leqslant \kappa \leqslant J'} \left\{ \max_{\boldsymbol{x}_j, \boldsymbol{x}_j \in C_\kappa} d(\boldsymbol{x}_j, \boldsymbol{x}_j) \right\}} \right) \right)$
Silhouette index (Rousseeuw, 1987)	$\tilde{s} = \frac{1}{J}\sum_{j=1} s(\boldsymbol{x}_j)$ where $s(\boldsymbol{x}_j) = \frac{b(\boldsymbol{x}_j) - a(\boldsymbol{x}_j)}{\max(a(\boldsymbol{x}_j), b(\boldsymbol{x}_j))}$, $b(\boldsymbol{x}_j) = \min_{C_{j'} \neq C_{j'}} \frac{1}{n_{j'}} \sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{x}_j)$ and $a(\boldsymbol{x}_j) = \frac{1}{n_{j'}-1} \sum_{\boldsymbol{x}_j \in C_{j'}, j \neq j} d(\boldsymbol{x}_j, \boldsymbol{x}_j)$.

$\boldsymbol{c}_{j'}$ denotes the cluster center or centroid of cluster $C_{j'}$; $n_{j'}$ denotes the number of observations in cluster $C_{j'}$

the generalizability of the reduced structure to out-of-sample and out-of-time data, we use the test data set which contains data from the eight and most recent year.

3.4.1 Exploring the workers' compensation insurance database

Claim-related information at the company level We first explore the distribution of the claim-related information in our database. We consider the damage rate Y_{it} and the number of claims N_{it} of company i in year t. When calculating $Y_{it} = \mathcal{Z}_{it}/w_{it}$ (see equation (3.1)), we use the capped claim amount $\mathcal{Z}_{it}$ for company i in year t to prevent that large losses have a disproportionate impact on the results (see Campo and Antonio (2023) for details on the capping procedure). To account for inflation and for the size of the company, we express the damage rate per unit of company- and year-specific salary mass w_{it}. Figure 3.3 depicts the empirical distribution of the damage rates Y_{it} and number of claims N_{it} of the individual companies. A strong right skew is visible in the empirical distribution of the Y_{it} (Figure 3.3(a)) and the N_{it} (Figure 3.3(c)). Moreover, this right skewness persists in the log transformed counterparts (see panels (b) and (d) in Figure 3.3).

Claim-related information at different levels in the NACE-Ins hierarchy The NACE-Ins partitions the companies according to their economic activity, at varying levels of granularity. We compute the category-specific weighted average damage rate and claim frequency at all levels in the hierarchy. Considering the top level in the hierarchy (as explained in Section 3.2), we calculate the weighted average damage rate for category $j = (1, \ldots, J)$ as

$$\bar{Y}_j = \frac{\sum_{i,k,t} w_{ijkt} Y_{ijkt}}{\sum_{i,k,t} w_{ijkt}} \tag{3.6}$$

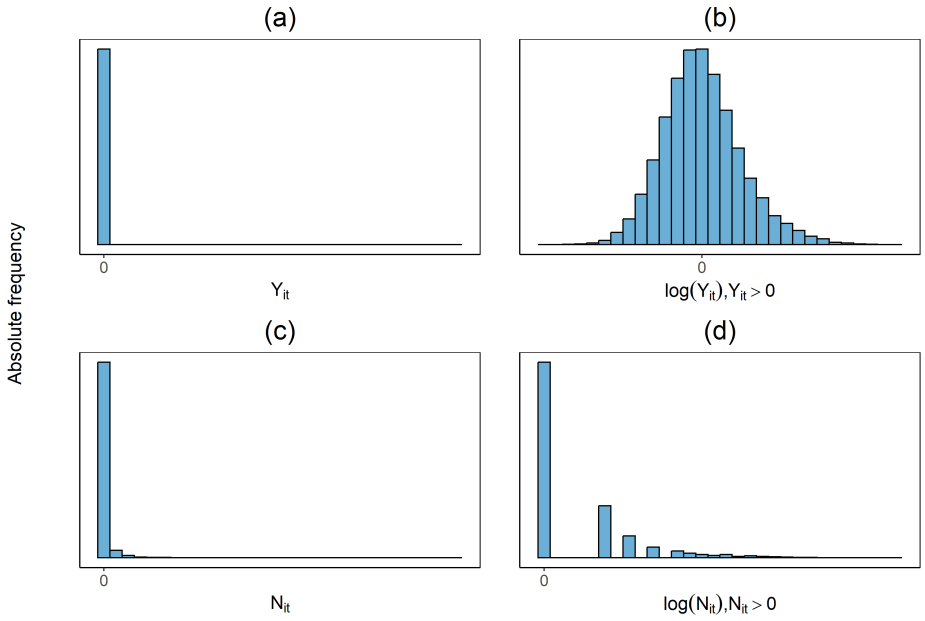
and the expected claim frequency as

$$\bar{C}_j = \frac{\sum_{i,k,t} N_{ijkt}}{\sum_{i,k,t} w_{ijkt}}. \tag{3.7}$$

We then calculate the weighted average damage rate and expected claim frequency for child-category $jk = (j1, \ldots, jK_j)$ within parent-category j as

$$\bar{Y}_{jk} = \frac{\sum_{i,t} w_{ijkt} Y_{ijkt}}{\sum_{i,t} w_{ijkt}} \qquad \text{and} \qquad \bar{C}_{jk} = \frac{\sum_{i,t} N_{ijkt}}{\sum_{i,t} w_{ijkt}}. \tag{3.8}$$

Figure 3.3: Empirical distribution of the individual companies' (a) damage rates Y_{it}; (b) log transformed damage rates Y_{it} for $Y_{it} > 0$; (c) number of claims N_{it}; (d) log transformed N_{it} for $N_{it} > 0$. This figure depicts the Y_{it}'s and N_{it}'s of all available years in our data set.



When we consider more granular levels in the hierarchy, the computation is similar. To calculate these quantities for a specific category, we only use observations that are classified herein.

At different levels in the hierarchy, the empirical distribution of the category-specific weighted average damage rates and expected claims frequencies show a strong right skew (see Appendix B.4). The large range in values hinders the visual comparison of the weighted average damage rates and expected claim frequencies across categories at different levels in the hierarchy. We therefore apply a transformation solely for the purpose of visual comparison. Illustrating the procedure with $\bar{Y}_j$, we first apply the following transformation

$$\bar{\mathcal{Y}}_j = \log(\bar{Y}_j + 0.0001) \tag{3.9}$$

since we have categories for which $\bar{Y}_j = 0$. Hereafter, we cap $\bar{\mathcal{Y}}_j$ using

$$\bar{\mathcal{Y}}_j^c = \max(\min(\bar{\mathcal{Y}}_j,\ Q_3(\bar{\mathcal{Y}}_j) + 1.5\ \text{IQR}(\bar{\mathcal{Y}}_j)),\ Q_1(\bar{\mathcal{Y}}_j) - 1.5\ \text{IQR}(\bar{\mathcal{Y}}_j)) \tag{3.10}$$

where $Q_n(\bar{\mathcal{Y}}_j)$ denotes the n^{th} quantile of $\bar{\mathcal{Y}}_j$ and $\text{IQR}(\bar{\mathcal{Y}}_j) = Q_3(\bar{\mathcal{Y}}_j) - Q_1(\bar{\mathcal{Y}}_j)$ denotes the interquartile range of $\bar{\mathcal{Y}}_j$. The lower and upper bound of $\bar{\mathcal{Y}}_j^c$ correspond to the inner fences of a boxplot (Schwertman et al., 2004). By transforming and capping the quantities, we focus on the pattern seen in the majority of the categories. Additionally, this facilitates the visual comparison across different categories.

Figure 3.4 visualizes the category-specific weighted average damage rates and salary masses. Panel (a) depicts the category-specific weighted averages at all levels in the hierarchy, panel (b) is close-up of the top right portion of panel (a) and panel (c) represents the averages at the `subsection` and `tariff group` level. We use a colour gradient for the weighted average damage rate. The darker the colour, the higher the weighted average damage rate. Further, each ring in the circle corresponds to a specific level in the hierarchy and the level is depicted by the number in the ring. To represent the categories at a specific level, the rings are split into slices proportional to the corresponding summed salary mass. The bigger the slice, the larger the summed salary mass that corresponds to a specific category at the considered level in the hierarchy. To preserve the confidentiality of the data, we randomly assign Greek letters to each of the categories at the `section` level.

At all levels in the hierarchy, there is variation in the category-specific weighted average damage rates. This is demonstrated in Figure 3.4(b), displaying a magnified view of the top right section of Figure 3.4(a). Here, the blue tone varies between the child categories of parent category λ at the `section` level. Additionally, at all levels in the hierarchy, there are categories with a low summed salary mass (e.g. $\gamma, \beta, \upsilon, \delta, \rho, \phi$ at the `section` level). In Figure 3.4, this is depicted by the thin slices. These categories represent only a small part of our portfolio. Furthermore, most of the categories with a low salary mass have a less granular representation in the NACE system, having very few child categories compared to other categories. For example, the parent category ϕ at the `section` level, has a low salary mass and only a limited number of child categories across all levels in the hierarchy of the NACE system.

Figure 3.5 depicts the category-specific expected claim frequency and salary masses. Similarly to Figure 3.4, we use a colour gradient for the claim frequencies and the size of a slice is again proportional to the summed salary mass. Overall,

Figure 3.4: Category-specific weighted average damage rates: (a) at all levels in the hierarchy; (b) of the `section` λ and ϕ, including those of their child categories at all levels in the hierarchy; (c) at the `subsection` and `tariff group` level. (b) is a close-up of the top right part of (a). In this close-up, the width of ϕ at the `section` level and its child categories is increased by a factor 10 to allow for better visual inspection.

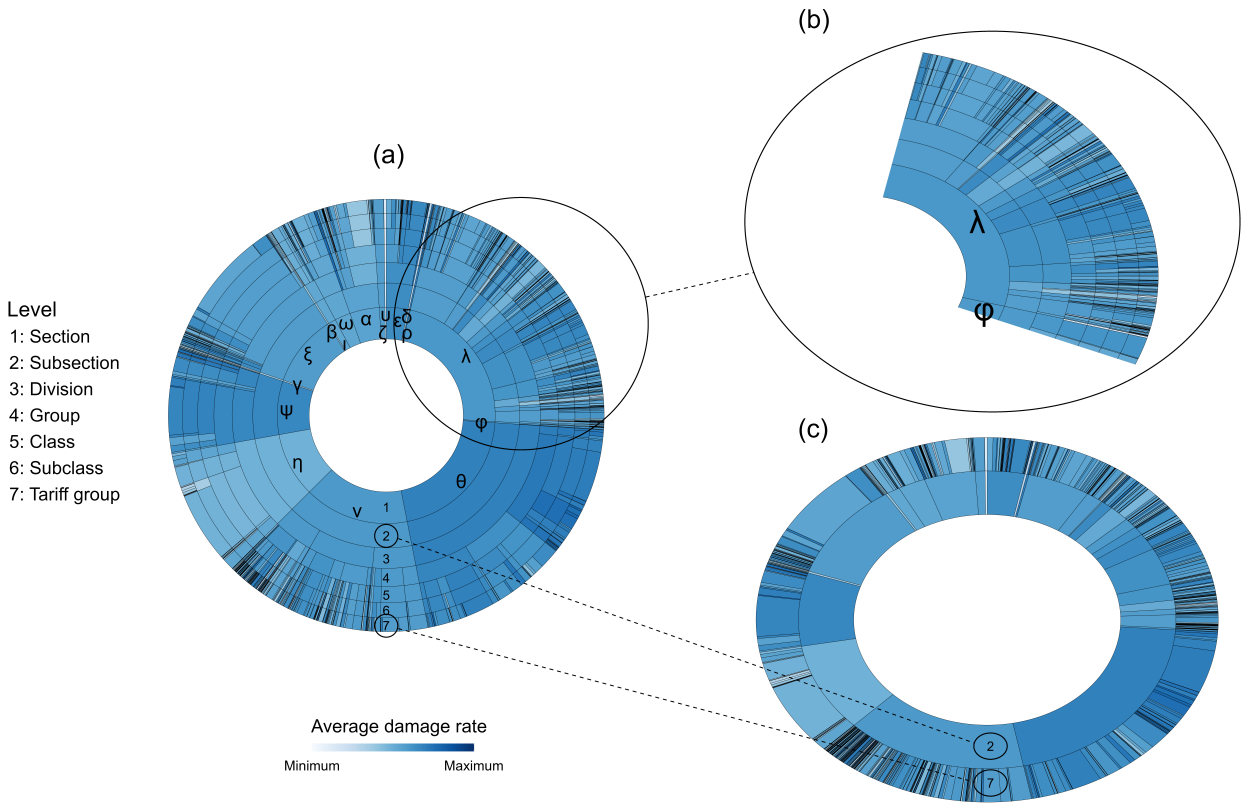
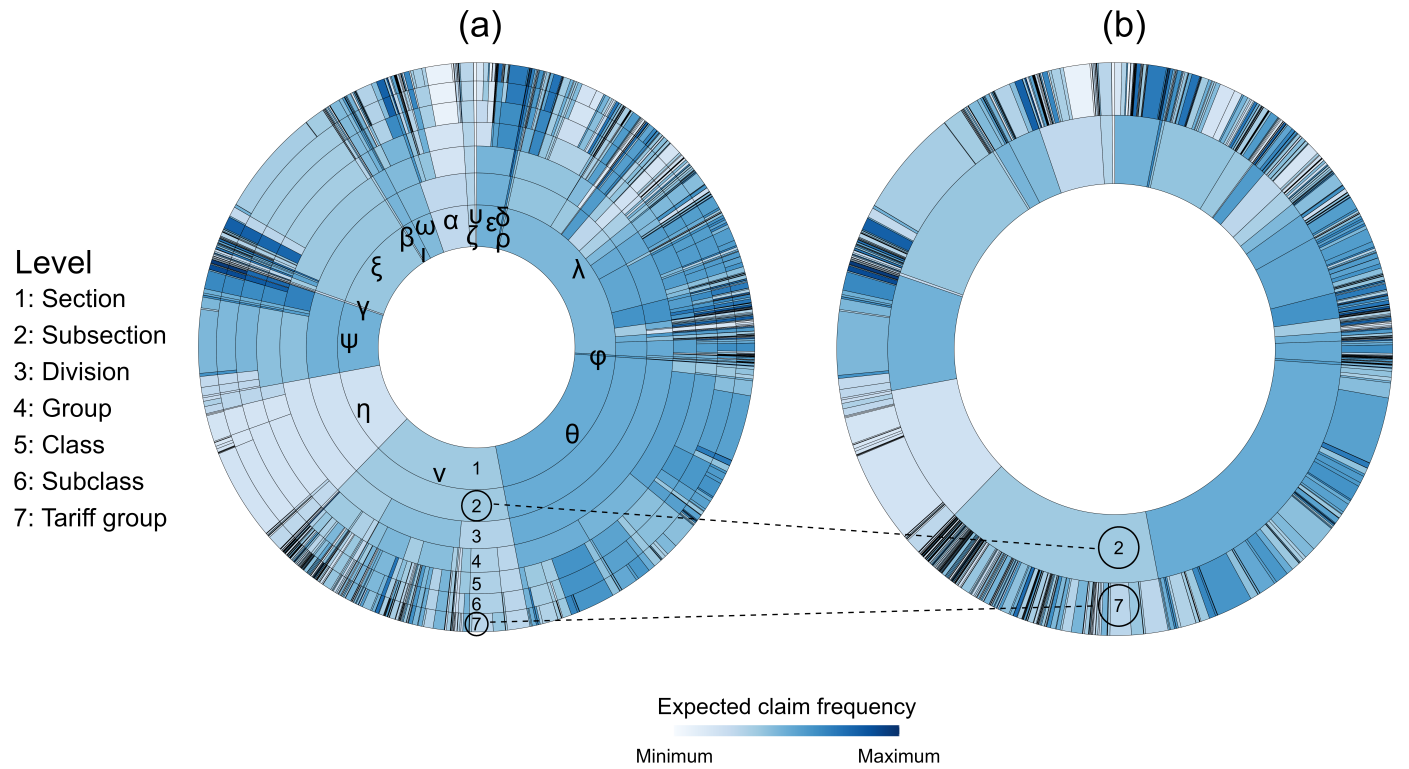


Figure 3.5: Category-specific expected claim frequencies: (a) at all levels in the hierarchy; (b) at the `subsection` and `tariff group` level.



the findings are comparable to Figure 3.4. The category-specific expected claim frequency varies between the categories at all levels in the hierarchy.

Following, we focus only on the `subsection` and `tariff group` level. The category-specific weighted average damage rates and expected claim frequencies at the `subsection` and `tariff group` level are shown in Figure 3.4(c) and Figure 3.5(b), respectively. To illustrate PHiRAT, we group similar categories at these levels in the hierarchy and hereby, reduce the granularity of the hierarchical risk factor.

3.4.2 Engineering features to improve clustering results

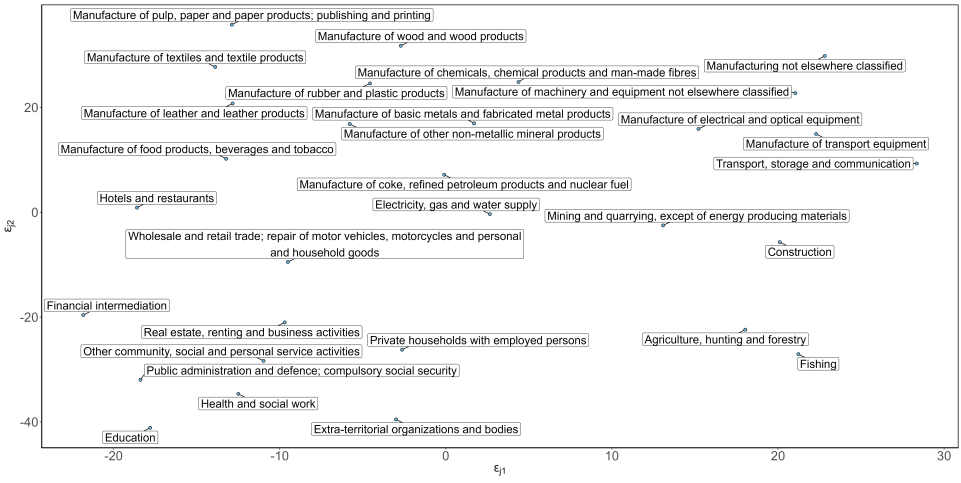
Feature engineering is crucial to obtain a reliable clustering solution through PHiRAT. As discussed in Section 3.2.2, we therefore engineer a set of features to capture the riskiness and the economic activity of the categories. The predicted random effect from the damage rate and claim frequency model expresses the category-specific riskiness at the `subsection` and `tariff group` level. Further, we use LMMs to fit the damage rate random effects models in (3.2) and (3.4). LMMs are less complex, computationally more efficient and are less likely to experience convergence problems compared to GLMMs. Alternatively, we can consider Tweedie GLMMs to model the damage rate. We refer the reader to Campo and Antonio (2023) for a discussion on the effect of the distributional assumption on the response. We use embeddings to encode the category's textual labels that describe the economic activity. In our database, we have a description for every category at the `subsection` and `tariff group` level.

To encode the textual information, we rely on pre-trained encoders. The first pre-trained encoder we use is a Word2Vec model (Mikolov et al., 2013) trained on part of the Google News data set that contains approximately 100 billion words (`https://code.google.com/archive/p/word2vec/`). This encoder is only able to give vector representations of words. As a result, when encoding a sentence for example, we get a separate vector for each word in the sentence. To obtain a single embedding for a sentence, we first remove stop words (e.g. the, and, ...) and then take the element-wise average of the embedding vectors of the individual words (Troxler and Schelldorfer, 2022). We also use the Universal Sentence Encoder (USE) trained on Wikipedia, web news, web question-answer pages and discussion forums (Cer et al., 2018) (`https://tfhub.dev/google/collections/universal-sentence-encoder/1`). There are two different versions, v4 and v5, which are specifically designed to encode greater-than-word length text (i.e. sen-

tences, phrases or short paragraphs). In this chapter, we use both versions.

The pre-trained Word2Vec encoder outputs a 300-dimensional embedding vector and the USEs a 512-dimensional vector. To assess the quality of the resulting embeddings, we inspect whether embeddings of related economic activities lie close to each other in the vector space. We employ the dimension reduction technique t-distributed stochastic neighbour embedding (t-SNE) (Van Der Maaten and Hinton, 2008) to obtain a two-dimensional visualization of the embeddings constructed for different categories at the `subsection` level (see Figure 3.5). We opt to reduce the dimensionality to two dimensions, since it allows for an easy representation of the information in a scatterplot. t-SNE maps the high-dimensional embedding vector $\boldsymbol{e}_j$ for category j into a lower dimensional representation $\boldsymbol{\varepsilon}_j = (\varepsilon_{j1}, \varepsilon_{j2})$ whilst preserving its structure, enabling the visualization of relationships and the identification of patterns and groups. Figure 3.6 visualizes the low-dimensional representation of the embeddings resulting from the pre-trained Word2Vec encoder (see Appendix B.5 for similar figures of USE v4 and v5).

Figure 3.6: Low-dimensional visualization of all embedding vectors at the `subsection` level, resulting from the pre-trained Word2Vec model, encoding the textual labels of the categories (see Figure 3.4). The text boxes display the textual labels. The blue dots connected to the boxes depict the position in the low-dimensional representation of the embeddings.



In Figure 3.6 we see that the embeddings of economically similar activities lie close to each other. All manufacturing related activities are situated at the top of the plot and in the left bottom corner, we have mostly activities that have a social

component (e.g., education). In the right bottom corner, we have activities related to the exploitation of natural resources. Further, the more unrelated the activities are, the bigger the distance between their embedding vectors. For example, *financial intermediation* ($\langle \varepsilon_{j1}, \varepsilon_{j2} \rangle \cong \langle -22.5, -20 \rangle$) and *transport, storage and communication* ($\langle \varepsilon_{j1}, \varepsilon_{j2} \rangle \cong \langle 28, 10 \rangle$) lie at opposite sides of the plot.

3.4.3 Clustering subsections and tariff groups using PHiRAT

We illustrate the effectiveness of PHiRAT by applying it to cluster categories at the `subsection` and `tariff group` level, using the training set. We slightly adjust Algorithm 2 to ensure that each of the resulting categories at the `subsection` and `tariff group` level has sufficient salary mass. The minimum salary mass is based on previous analyses of the insurance company and we need to adhere to this minimum during clustering. Further, as discussed at the end of Section 3.3, we run PHiRAT with all possible combinations of the clustering methods (i.e. k-medoids, spectral clustering and HCA) and internal validation criteria (i.e. Caliński-Harabasz index, Davies-Bouldin index, Dunn index and silhouette index). Per run, a specific combination is used to cluster the categories at all levels in the hierarchy.

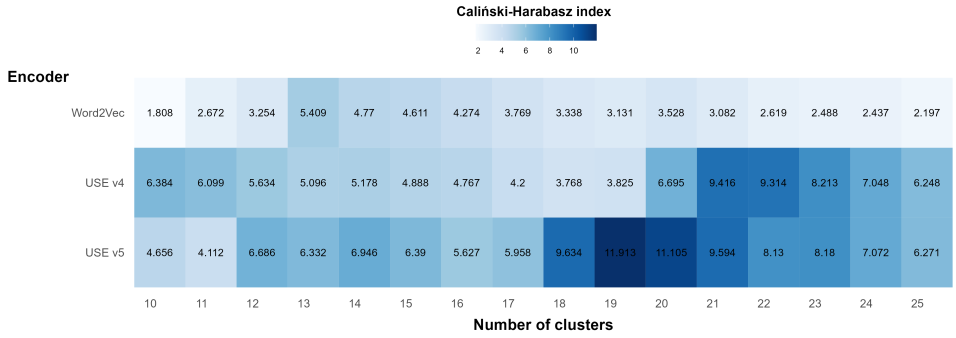
Figure 3.2 visualizes how PHiRAT works its way down the hierarchy. We start at the `subsection` level and construct the feature matrix $\mathcal{F}_1$. Part of $\mathcal{F}_1$ is built using an encoder to capture the textual information (see Section 3.2.2 and Section 3.4.2). In our paper, we consider three different pre-trained encoders: (a) Word2Vec; (b) USE v4 and (c) USE v5. We use these encoders to obtain the embeddings and construct three separate feature matrices: (a) $\mathcal{F}_1^{\text{Word2Vec}}$; (b) $\mathcal{F}_1^{\text{USEv4}}$ and (c) $\mathcal{F}_1^{\text{USEv5}}$. All three feature matrices contain the same predicted random effects vectors ${}_1\widehat{\boldsymbol{U}}^d$ and ${}_1\widehat{\boldsymbol{U}}^f$. The difference between the feature matrices is that the embeddings are encoder-specific. Next, we define a tuning grid for the number of clusters $\mathcal{K}_1 \in (10, 11, \ldots, 24, 25)$. Hence, the lower and upper bound to the number of grouped categories is determined by the minimum and maximum value of the tuning grid, respectively. In combination with the encoder-specific feature matrices, this results in a search grid of three (i.e. the encoder-specific feature matrices) by 16 (i.e. the possible values for the number of clusters). For every combination in this grid, we run the clustering algorithm to obtain a clustering solution and calculate the value of the internal validation criterion. We select the combination that results in the most optimal clustering solution according to the validation measure. Figure 3.7 provides a visualization of this search grid. In this figure, we use k-medoids as

clustering algorithm, the CH index as internal validation measure and we use a colour gradient for the value of the validation criterion. The darker the colour, the higher the value and the better the clustering solution. The x-axis depicts the tuning grid for $\mathcal{K}_1$ and the y-axis the encoder-specific feature matrices. The CH index is highest (= 11.913) for $\mathcal{K}_1 = 19$ in combination with $\mathcal{F}_1^{\text{USEv5}}$. According to the selected validation measure, this specific combination results in the most optimal clustering solution. Hence, we use this clustering solution to group the categories $j = (1, \ldots, J)$ into clusters $j' = (1, \ldots, J')$ (see Figure 3.2). Hereafter, we merge clusters $j' = (1, \ldots, J')$ with neighbouring clusters until each cluster has sufficient salary mass.

Next, we proceed to cluster the categories at the `tariff group` level. Within each cluster j' at the `section` level, we first group consecutive categories (i.e. those with consecutive NACE codes, see Section 3.2.1) at the `tariff group` level to ensure that the salary mass is sufficient for every category. This level is a highly granular representation of the economic activity. As a result, we have several categories with a low number of observations and/or salary mass. By first merging consecutive tariff groups, we ensure that each category has sufficient information and avoid potential convergence problems. As before, we construct three encoder-specific versions of $\mathcal{F}_2$. We then iterate over every parent category j' to group the child categories. At iteration j', we use $\mathcal{F}_{2,\{c:\pi(c)=j'\}}$ as input in a clustering algorithm. At the `tariff group` level, we use the tuning grid $\mathcal{K}_{2,\{c:\pi(c)=j'\}} \in (5, 6, \ldots, \min(K_{j'}, 25))$. Here $K_{j'}$ denotes the total number of child categories of parent category j'. Similarly, the lower and upper bound to the number of grouped child categories is dependent on the minimum and maximum value in the tuning grid. We select the combination of the encoder-specific feature matrix and $\mathcal{K}_{2,\{c:\pi(c)=j'\}}$ that results in the most optimal clustering solution to group the categories $j'k = (j'1, \ldots, j'K_{j'})$, nested within j', into subclusters $j'k' = (j'1, \ldots, j'K'_{j'})$ (see Figure 3.2).

Distance measures and evaluation metrics For k-medoids clustering and HCA we use the angular distance (see Section 3.3.2), as both algorithms require a distance or dissimilarity measure. For spectral clustering we use the angular similarity. We also need to define a distance measure $d(\cdot, \cdot)$ for the selected internal evaluation criteria, except for the CH index. Hereto, we define $d(\cdot, \cdot)$ as the angular distance. We calculate the evaluation criteria using all engineered features. We do not standardize the features as this would transform the space of the embeddings and hereby disrupt the placement of the textual information in the embedding space.

Figure 3.7: Visualization of the search grid. The x-axis depicts the tuning grid $\mathcal{K}_1$ and the y-axis the encoder-specific feature matrix that is used. Here, we use k-medoids for clustering and the CH index as internal validation measure. The CH index is highest ($= 11.913$) for $\mathcal{K}_1 = 19$ in combination with $\mathcal{F}_1^{\text{USEv5}}$, indicating that this combination results in the most optimal clustering solution.



Further, since a large part of the feature vector consists of the high-dimensional embedding vector, too much weight might be given to the similarity in economic activity when choosing the optimal cluster solution. Therefore, we also evaluate a second variation of the internal evaluation criteria. When calculating the criteria, we remove the embedding vectors from the feature matrices. As such, we focus on constructing a clustering solution that is most optimal in terms of riskiness. In this evaluation we use the Euclidean distance for $d(\cdot, \cdot)$. Hence, at the `subsection` level, we define $d(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) = \|\boldsymbol{x}_j - \boldsymbol{x}_{j'}\|_2^2$ where $\boldsymbol{x}_j = (\widehat{U}_j^d, \widehat{U}_j^f)$. Similarly, at the `tariff group` level, $d(\boldsymbol{x}_{jk}, \boldsymbol{x}_{j\ell}) = \|\boldsymbol{x}_{jk} - \boldsymbol{x}_{j\ell}\|_2^2$ and $\boldsymbol{x}_{jk} = (\widehat{U}_{jk}^d, \widehat{U}_{jk}^f)$. In what follows we discuss the results obtained with this approach. Results obtained with the complete feature vector, including the embedding, are given in Appendix B.6.

Implementation We perform the main part of the analysis using the statistical software R (R Core Team, 2019). For k-medoids and HCA, we rely on the `cluster` (Maechler et al., 2022) and `stats` package, respectively. For spectral clustering, we follow the implementation of Ng et al. (2001) and developed our own code. Spectral clustering partially depends on the k-means algorithm to obtain a clustering solution (see Appendix B.2). However, k-means is sensitive to the initialization and can get stuck in an inferior local minimum (Fränti and Sieranoja, 2019). One way to alleviate this issue is by repeating k-means with different initializations and to select the most optimal clustering solution. Hence, to prevent that spectral clustering

results in a suboptimal clustering solution, we repeat the k-means step a 100 times.

3.4.4 Evaluating the clustering solution

The main aim of our procedure is to reduce the cardinality of a hierarchically structured categorical variable, which can then be incorporated as a risk factor in a predictive model to underpin the technical price list (Campo and Antonio, 2023). Ideally, we maintain the predictive accuracy whilst reducing the granularity of the risk factor. As discussed in Section 3.4.3, we employ PHiRAT to group similar categories at the `subsection` and `tariff group` level. We refer to the less granular version of the hierarchical risk factor as the reduced risk factor.

To assess the predictive accuracy of a clustering solution, we fit an LMM

$$E[Y_{ij'k't}|U^d_{j'}, U^d_{j'k'}] = \mu^d + U^d_{j'} + U^d_{j'k'} \tag{3.11}$$

where the reduced risk factor enters the model through the random effects $U_{j'}$ and $U_{j'k'}$. We include the salary mass $w_{ij'k't}$ as weight. We opt for an LMM given its simplicity and computational efficiency (Campo and Antonio, 2023). Next, we calculate the predicted damage rate as $\widehat{Y}_{ij'k't} = \hat{\mu}^d + \widehat{U}^d_{j'} + \widehat{U}^d_{j'k'}$ for the training and test set as introduced in the beginning of Section 3.4.

Benchmark clustering solution To evaluate the clustering solution obtained with the proposed data-driven approach, we need to compare it to a benchmark where we do not rely on PHiRAT. One possibility is to fit (3.11) with the nominal variable composed of the original categories at the `subsection` and `tariff group` level. In our example, fitting this LMM results in negative variance estimates and yields incorrect random effect predictions. Consequently, to obtain a benchmark clustering solution, we start at the `subsection` level and merge consecutive categories until the salary mass is sufficient (e.g. *01 agriculture* and *02 forestry*). Similarly, we index the merged categories using $j' = (1, \ldots, J')$. Hereafter, within each of the (grouped) categories at the `subsection` level, we group consecutive categories (e.g. *142121* and *142122*) at the `tariff group` level. Again, we use the minimum salary mass as defined by the insurance company and we use $k' = (1, \ldots, K'_{j'})$ as an index for the grouped categories. This results in a hierarchical MLF in which we merged adjacent categories with insufficient salary mass. To construct the benchmark model, we fit an LMM with the same specification as in (3.11).

Performance measures Using the predicted damage rates on the training and test set, we compute the Gini-index (Gini, 1921) and loss ratio. The Gini-index assesses how well a model is able to distinguish high-risk from low-risk companies and is considered appropriate for the comparison of competing pricing models (Denuit, Sznajder and Trufin, 2019; Campo and Antonio, 2023). The higher the value, the better the model can differentiate between risks. The Gini-index has a maximum theoretical value of 1. The loss ratio measures the overall accuracy of the fitted damage rates and is defined as $\mathcal{Z}_t^{tot}/\widehat{\mathcal{Z}}_t^{tot}$, where $\mathcal{Z}_t^{tot} = \sum_{i,j',k'} \mathcal{Z}_{ij'k't}$ denotes the total capped claim amount and $\widehat{\mathcal{Z}}_t^{tot} = \sum_{i,j',k'} \widehat{\mathcal{Z}}_{ij'k't}$ the total fitted claim amount. To obtain $\widehat{\mathcal{Z}}_{ij'k't}$, we transform the individual predictions $\widehat{Y}_{ij'k't}$ as (see equation (3.1))

$$\widehat{\mathcal{Z}}_{ij'k't} = \widehat{Y}_{ij'k't} \cdot w_{ij'k't}. \tag{3.12}$$

Due to the balance property (Campo and Antonio, 2023), the loss ratio is one when calculated using the training data set. We therefore compute the loss ratio only for the test set.

Predictive performance Table 3.9 depicts the performance of the different clustering solutions on the training and test sets. In this table, J' denotes the total number of grouped categories at the `subsection` level and $\sum_{j'=1}^{J'} K'_{j'}$ denotes the total number of grouped categories at the `tariff group` level. With the benchmark clustering solution, we end up with a large number of different categories at both hierarchical levels ($18 + 641 = 659$ separate categories in total). Conversely, with our data-driven approach the maximum is 221 (14 + 207; k-medoids with Davies-Bouldin index) and the minimum is 97 (10 + 87; k-medoids with the CH index). Consequently, PHiRAT substantially reduces the number of categories. Moreover, we are able to retain the predictive performance. On both the training and test set, the Gini-indices of nearly all clustering solutions are higher than the Gini-index of the benchmark solution. Hence, we are better able to differentiate between high- and low-risk companies with the clustering solutions. On the training data set, the highest Gini-indices are seen for the k-medoids algorithm using the silhouette index and the spectral clustering algorithm using the Dunn index. On the test set, spectral clustering using the CH index and silhouette index result in the highest Gini-index whereas the benchmark has the lowest Gini-index. However, the loss ratio of most clustering solutions is higher than the loss ratio of the benchmark clustering solution (= 1.006). This indicates that, when we use the reduced risk factor in an LMM, we

generally underestimate the total damage. One exception is the clustering solution resulting from HCA using the Davies-Bouldin index, which has the same loss ratio as the benchmark (= 1.006). In addition, for this solution the Gini-index is among the highest (0.675 on the training and 0.616 on the test set). Using the result from HCA with the Davies-Bouldin index, we are able to reduce the total number of categories to 207 (= 14 + 193). Compared to the benchmark solution, we obtain the same overall predictive accuracy (i.e. the loss ratio is approximately equal) and we can better differentiate between high- and low-risk companies.

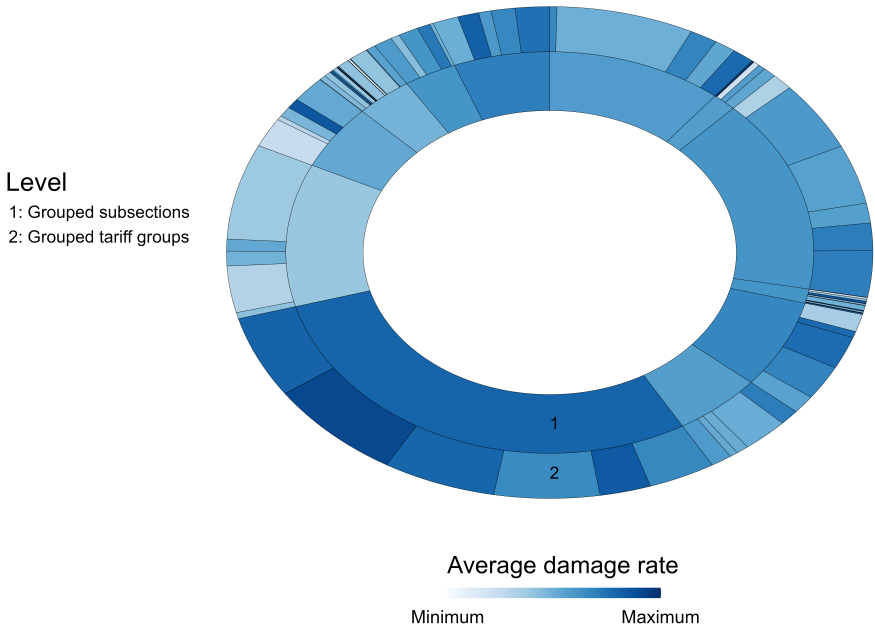
Table 3.9: Predictive performance on the training and test set.

			Training	Test	
	J'	$\sum_{j'=1}^{J'} K'_{j'}$	Gini-index	Gini-index	Loss ratio
Benchmark	18	641	0.658	0.585	1.006
HCA:					
Silhouette index	13	100	0.667	0.596	1.007
Dunn index	15	202	0.656	0.599	1.011
Davies-Bouldin index	14	193	0.675	0.616	1.006
CH index	15	140	0.667	0.614	1.010
k-medoids:					
Silhouette index	12	107	0.678	0.613	1.008
Dunn index	10	130	0.657	0.598	1.010
Davies-Bouldin index	14	207	0.670	0.619	1.010
CH index	10	87	0.676	0.605	1.011
Spectral clustering:					
Silhouette index	12	86	0.673	0.628	1.012
Dunn index	12	143	0.677	0.624	1.010
Davies-Bouldin index	12	166	0.669	0.618	1.010
CH index	12	102	0.670	0.628	1.013

The clustering solution resulting from HCA with the Davies-Bouldin index offers a good balance between reducing granularity and enhancing differentiation whilst preserving predictive accuracy. If, however, a sparse representation and good differentiation is more important than the overall predictive accuracy, the result from spectral clustering with the silhouette index is a better option. The latter clustering solution is visualized in Figure 3.8. This figure is similar to Figure 3.4(a) and depicts the cluster-specific weighted average damage rates at the `subsection` and `tariff group` level. The two figures show that PHiRAT substantially reduces the total number of categories (12 + 86 = 98). Furthermore, spectral clustering with the silhouette index is one of the combinations that results in the best differentiation

between high- and low-risk companies (Gini index is 0.673 and 0.628 on the training and test set, respectively).

Figure 3.8: Cluster-specific weighted average damage rates at the `subsection` and `tariff group` level, when employing PHiRAT with spectral clustering and the silhouette index.



Interpretability After clustering, it is imperative to evaluate the interpretability of the solution. The grouping should provide us with meaningful insights into the underlying structure of the data. Building on the strong predictive performance of the solution resulting from spectral clustering with the silhouette index, we examine this specific clustering solution in more detail.

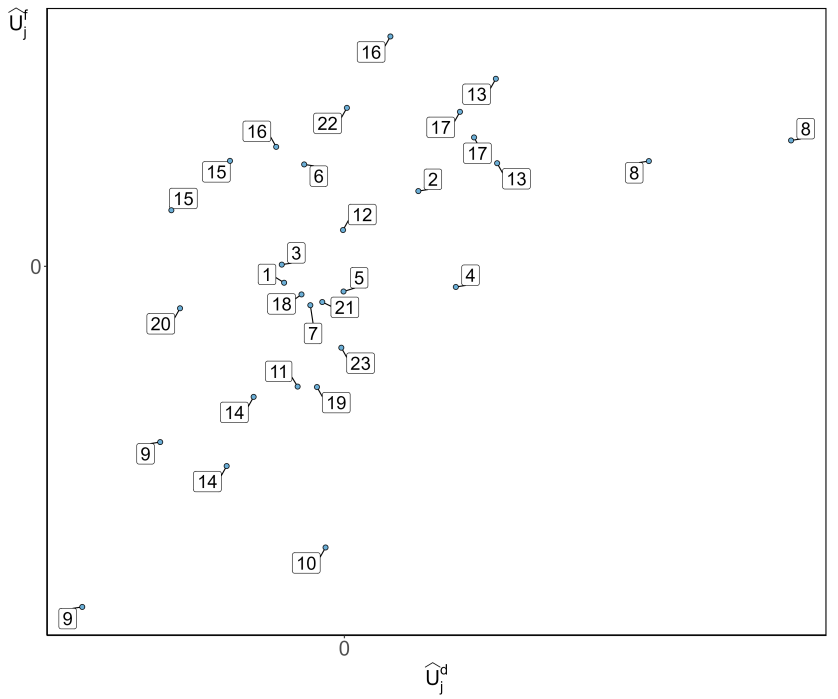
Figure 3.9 visualizes which categories are clustered at the `subsection` level. At this point we have not yet merged neighbouring clusters to ensure sufficient salary mass (see Section 3.4.3). Figure 3.9(a) shows the $\widehat{U}_j^d$ and $\widehat{U}_j^f$ of the fitted damage rate and claim frequency random effects models (see (3.2) and (3.3)). The number in the plot indicates which cluster the categories are appointed to. The description of the category's economic activity is given in Figure 3.9(b). In total we have 24 clusters, 19 of which consist of a single category. Inspecting the clusters in detail, we find

that the riskiness and economic activity of grouped categories are similar. Moreover, categories that are similar in terms of riskiness (i.e. those with similar random effect predictions) but different in economic activity are assigned to different clusters. For example, the $\widehat{U}_j^d$'s and $\widehat{U}_j^f$'s are similar for: a) *manufacture of rubber and plastic products*; b) *agriculture, hunting and forestry*; c) *manufacture of wood and wood products*; d) *manufacture of other non-metallic mineral product* and e) *manufacture of basic metals and fabricated metal products*. These industries are partitioned into three clusters. The category in cluster 2 (i.e. *manufacture of rubber and plastic products*) manufactures organic products. Conversely, the categories in cluster 17 (i.e. *manufacture of other non-metallic mineral product* and *manufacture of basic metals and fabricated metal products*) use inorganic materials and the categories in cluster 13 (i.e. *agriculture, hunting and forestry* and *manufacture of wood and wood products*) utilize products derived from plants and animals.

Figure 3.10 depicts the clustered categories k' at the `tariff group` level, nested within category $j' =$ *manufacture of chemicals, chemical products and man-made fibres* at the `subsection` level. The $\widehat{U}_{j'k}^d$ and $\widehat{U}_{j'k}^f$ of the fitted damage rate and claim frequency random effects model (see (3.4) and (3.5)) are shown in the left panel of Figure 3.10 and the textual information in the right panel. When the text is separated by a semicolon, this indicates that categories are merged before clustering to ensure sufficient salary mass (see Section 3.4.3). Similar to Figure 3.9, both the riskiness and economic activity are taken into account when grouping categories. This is best illustrated for the categories in the red rectangle in Figure 3.10(a). Herein, we have three categories: (a) *pyrotechnic items: manufacture; glue, gelatin: manufacture*; (b) *chemical basic industry* and (c) *pharmaceutical industry*. Of these, *pyrotechnic items: manufacture; glue, gelatin: manufacture* (number 7 in the top right corner of the red rectangle) and *chemical basic industry* (number 3 in the top right corner of the red rectangle) have nearly identical random effect predictions. Notwithstanding, both categories are not grouped. Instead, the category *chemical basic industry* is merged with *pharmaceutical industry* (number 3 in the bottom left corner). The latter two categories manufacture chemical compounds. In addition, companies in *chemical basic industry* mainly produce chemical compounds that are used as building blocks in other products (e.g. polymers). Building blocks (or raw materials) that are needed by companies involved in manufacturing pyrotechnic items, glue and gelatin (e.g. glue is typically made from polymers (Ebnesajjad, 2011)).

Figure 3.9: Visualization of the grouped categories at the `subsection` level (see Figure 3.2): (a) the clustered categories and the original random effect predictions $\widehat{U}_j^d$ and $\widehat{U}_j^f$; (b) the description of the economic activity of the categories.

(a)

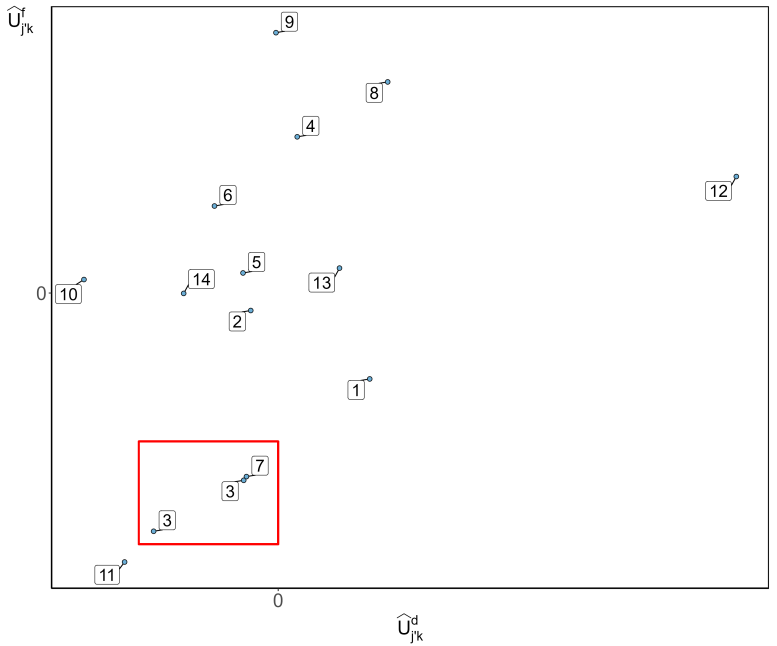


(b)

Cluster	Description
1	Real estate, renting and business activities
2	Manufacture of rubber and plastic products
3	Manufacture of food products, beverages and tobacco
4	Mining and quarrying, except of energy producing materials
5	Manufacture of coke, refined petroleum products and nuclear fuel
6	Manufacture of transport equipment
7	Public administration and defence; compulsory social security
8	Construction
8	Transport, storage and communication
9	Manufacture of pulp, paper and paper products; publishing and printing
9	Hotels and restaurants
10	Extra-territorial organizations and bodies
11	Manufacture of leather and leather products
12	Fishing
13	Agriculture, hunting and forestry
13	Manufacture of wood and wood products
14	Other community, social and personal service activities
14	Private households with employed persons
15	Education
15	Health and social work
16	Manufacture of machinery and equipment not elsewhere classified
16	Manufacturing not elsewhere classified
17	Manufacture of other non-metallic mineral products
17	Manufacture of basic metals and fabricated metal products
18	Manufacture of textiles and textile products
19	Financial intermediation
20	Manufacture of electrical and optical equipment
21	Wholesale and retail trade; repair of motor vehicles, motorcycles and personal and household goods
22	Electricity, gas and water supply
23	Manufacture of chemicals, chemical products and man-made fibres

Figure 3.10: Visualization of the clustered child categories at the `tariff group` level, with parent category *manufacture of chemicals, chemical products and man-made fibres* at the `subsection` level (see Figure 3.2): (a) the clustered categories and the original random effect predictions $\widehat{U}^{d}_{j'k}$ and $\widehat{U}^{f}_{j'k}$; (b) the description of the economic activity of the categories.

(a)



(b)

Cluster	Description
1	Laundry and dishwasher detergents (synthetic): manufacture; maintenance products: manufacture
2	Synthetic products (organic): manufacture; chemical basic product (organic) not defined: manufacture
3	Chemical basic industry
3	Pharmaceutical industry
4	Chemical products for textile and leather: manufacture
5	Gas (industrial): manufacture; dyes, pigments: manufacture
6	Chemical products of oils and fats: manufacture
7	Pyrotechnic items: manufacture; glue, gelatin: manufacture
8	Chemical basic product (inorganic) not defined: manufacture
9	Soap: manufacture
10	Essential oils: manufacture
11	Fertilizers: manufacture; polymers: manufacture
12	Rubber (synthetic): manufacture; paint, varnish, printing ink: manufacture
13	Chemical product (industrial use) not defined: manufacture; writing ink: manufacture; synthetic and artificial fibers: manufacture
14	Perfumes, cosmetics: manufacture

3.5 Discussion

This chapter presents the data-driven PHiRAT approach to reduce a hierarchically structured categorical variable with a large number of categories to its essence, by grouping similar categories at every level in the hierarchy. PHiRAT is a top-down procedure that preserves the hierarchical structure. It starts with grouping categories at the highest level in the hierarchy and proceeds to lower levels. At a specific level in the hierarchy, we first engineer several features to characterize the profile and specificity of each category. Using these features as input in a clustering algorithm, we group similar categories. The procedure stops once we grouped the categories at the lowest level in the hierarchy. When deployed in a predictive model, the reduced structure leads to a more parsimonious model that is easier to interpret, less likely to experience estimation problems or to overfit. Further, the increased number of observations of grouped categories leads to more precise estimates of the category's effect on the response.

Using a workers' compensation insurance portfolio from a Belgian insurer, we illustrate how to employ PHiRAT to reduce the granular structure of the NACE code. Using PHiRAT, we are able to substantially reduce the dimensionality of this hierarchical risk factor whilst maintaining its predictive accuracy. The reduced risk factor allows for better differentiation, has the same overall precision as the original risk factor and the grouping seems to generalize well to out-of-sample data. Moreover, the resulting clusters consist of categories that are similar in terms of riskiness and economic activity. Furthermore, the results show that embeddings are an efficient and effective method to capture textual information.

Our approach results in a clustering solution that can provide insurers with improved insights into the underlying risk structure of a hierarchical covariate. By capturing the key characteristics of the original hierarchically structured risk factor, the reduced risk factor offers a more informative and concise representation of the various risk profiles. Insurers can incorporate the reduced version into their pricing algorithms to better assess the riskiness associated with each category in said hierarchical covariate.

Negative variance estimates in the random effects model and computational limitations with the generalized fused lasso penalty prevent us from constructing a different benchmark model. Hence, future research can examine the robustness of our approach by examining its performance in other data sets or applications and by comparing it with alternative approaches. Additionally, future research can

aim to improve our proposed solution in multiple directions. The final clustering solution depends on the engineered features, the clustering algorithm and the cluster evaluation criterion. Constructing appropriate and reliable features when employing PHiRAT is crucial to obtain a good clustering solution. In this paper, we rely on random effects models to capture the riskiness of the categories. Alternatively, we can characterize the risk profile using entity embeddings (Guo and Berkhahn, 2016). Here, we train a neural network to create entity embeddings that map the categorical values to a continuous vector, ensuring that categories with similar response values are located closer to each other in the embedding space. Additionally, we advise to run PHiRAT using different clustering algorithms and cluster evaluation criteria, since no algorithm consistently outperforms the others (Hennig, 2015; McNicholas, 2016*a*,*b*; Murugesan et al., 2021). Moreover, the robustness of the clustering solution largely depends on the stability of the selected clustering method. Further, NLP is a rapidly evolving field and we only considered three pre-trained encoders. We did not include the pre-trained Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2018), an encoder that is being increasingly used by actuarial researchers (see, for example, Xu et al. (2022); Troxler and Schelldorfer (2022)). Subsequent studies can assess whether BERT-based models result in higher quality embeddings, which in turn leads to better clustering solutions. In addition, researchers can employ alternative approaches to represent the parent-child relationships between categories in hierarchically structured data. Such as Argyrou (2009), for example, who formalizes the hierarchical relations using graph theory and applies a self-organizing map (Kohonen, 1995) to obtain a reduced representation of the hierarchical data.

Chapter 4

An engine to simulate insurance fraud network data

Traditionally, the detection of fraudulent insurance claims relies on business rules and expert judgement which makes it a time-consuming and expensive process (Óskarsdóttir et al., 2022). Consequently, researchers have been examining ways to develop efficient and accurate analytic strategies to flag suspicious claims. Feeding learning methods with features engineered from the social network of parties involved in a claim is a particularly promising strategy (see for example Óskarsdóttir et al. (2022); Van Vlasselaer et al. (2016); Tumminello et al. (2023)). When developing a fraud detection model, however, we are confronted with several challenges. The uncommon nature of fraud, for example, creates a high class imbalance which complicates the development of well performing analytic classification models. In addition, only a small number of claims are investigated and get a label, which results in a large corpus of unlabeled data. Yet another challenge is the lack of publicly available data. This hinders not only the development of new methods, but also the validation of existing techniques. We therefore design a simulation machine that is engineered to create synthetic data with a network structure and available covariates similar to the real life insurance fraud data set analyzed in Óskarsdóttir et al. (2022). Further, the user has control over several data-generating mechanisms. We can specify the total number of policyholders and parties, the desired level of imbalance and the (effect size of the) features in the fraud generating model. As such, the simulation engine enables researchers and practitioners to examine several

methodological challenges as well as to test their (development strategy of) insurance fraud detection models in a range of different settings. Moreover, large synthetic data sets can be generated to evaluate the predictive performance of (advanced) machine learning techniques.

This chapter is based on Bavo D.C. Campo and Katrien Antonio. (2023). An engine to simulate insurance fraud network data. arXiv: 2304.09046. Available at: `https://arxiv.org/abs/2304.09046`.

4.1 Introduction

Fraudulent activity in the insurance industry causes significant financial losses for both insurance companies and policyholders. In non-life insurance, the total yearly cost of fraudulent claims is estimated to be more than $40 billion in the United States (FBI, 2022). For an average family, this leads to an increased yearly premium of $400 to $700 (FBI, 2022). To detect and mitigate fraud, insurance companies implement various anti-fraud measures. Traditionally, insurance companies rely on a combination of business rules and expert judgment to identify fraudulent claims (Óskarsdóttir et al., 2022). The business rules flag suspicious claims, which are then sent to experts who determine whether the claim is fraudulent or not (Warren and Schweitzer, 2018). These in-depth investigations, however, are time-intensive and costly. Researchers have therefore developed insurance fraud detection models combining business rules and analytical techniques to flag the most suspicious claims, which are sent to the experts for further investigation. As such, experts can focus solely on the claims with a high likelihood of fraud and avoid spending precious resources on examining non-fraudulent claims. Insurers predominately rely on analytics to prevent fraud (European Insurance and Occupational Pensions Authority, 2019). Moreover, fraud detection is considered an area for more intense use of big data and analytics in the insurance industry.

Within fraud analytics, researchers rely on a wide range of statistical and machine learning techniques (see Ngai et al. (2011) and Albashrawi (2016) for an overview). In the literature, we find examples of both supervised and unsupervised (machine learning) techniques to construct fraud detection models. Vosseler (2022), for example, developed an unsupervised anomaly detection technique to identify fraudulent insurance claims. Nur Prasasti et al. (2020) constructed fraud detection models using neural networks and tree-based machine learning techniques. The accuracy of such models, however, greatly depends on the input. We typically use

traditional claim characteristics, such as the claim amount, as features in a fraud detection model (Baesens et al., 2015). These characteristics are static whereas the typical features of fraudsters tend to be dynamic (Óskarsdóttir et al., 2022; Gomes et al., 2021; Tumminello et al., 2023). According to Jensen (1997), fraudsters adapt their tactics in response to fraud detection systems and hence, the typical fraudster profile evolves over time.

One particularly promising approach to capture the characteristics of fraudsters is through social network analytics (Van Vlasselaer et al., 2016; Óskarsdóttir et al., 2022; Tumminello et al., 2023). In an insurance context, social network data captures the relationship between claims on the one hand and policyholders and other involved parties (e.g., garage, broker, expert) on the other hand. By analyzing the social network structure of reported claims, insurers can unravel patterns and relationships among policyholders and claims that are indicative of fraud. Moreover, this approach potentially uncovers organized schemes of collaborating fraudsters who are trying to hide their tracks. Criminals often commit crimes in groups to increase rewards and to decrease the risk of detection (Reiss, 1988; Andresen and Felson, 2009). Furthermore, in organized crime, such as fraud, social connections play a crucial role since these connections are based on trust and provide access to co-offenders and opportunities (van Koppen et al., 2010). As such, social network analytics can assist in identifying enduring relationships between fraudsters even as overt characteristics undergo changes.

Nonetheless, publicly available data on insurance fraud is scarce, in particular data with a social network structure. This makes it difficult for researchers to test, validate and improve existing fraud detection methods. Moreover, the lack of data hinders the reproducibility of research findings and the discovery of novel methodologies (Baesens, 2023). The main reason for the limited availability is the sensitive nature of the data (Lopez-Rojas et al., 2015). Insurance data sets contain confidential information about the insurance company and its policyholders. In consequence, the data used to develop and validate fraud detection methods and models is almost never shared. Researchers in fraud analytics are confronted with several inherent methodological challenges when developing analytic models for fraud detection (Baesens, 2023). Investigating suspicious claims is a time-intensive and expensive process, which results in only few claims being labeled (i.e. whether the claim is fraudulent or non-fraudulent) (Warren and Schweitzer, 2018). In addition, due to fraud being uncommon, data sets are often characterized by a severe class imbalance (see, for example Óskarsdóttir et al. (2022); Gomes et al. (2021); Subudhi

and Panigrahi (2020)). Another challenge is the continuous development of machine learning techniques (Baesens, 2023). To assess whether these perform better or on par with existing techniques, we need to evaluate competing models or techniques in similar conditions. That is, using the same (type of) data set.

One way to address the scarcity of publicly available data, is by using a simulation engine to generate synthetic data that mimics the structure of the original data set (Lopez-Rojas et al., 2015). Simulation engines enable researchers to perform benchmark studies to investigate the properties and performance of various statistical and machine learning techniques (Morris et al., 2019; Khondoker et al., 2016). Additionally, synthetic data facilitates the development of new methods and stimulates the reproducibility of research. Recently, simulation engines have gained considerable attention in actuarial science. Gabrielli and Wüthrich (2018) developed a simulation engine to generate individual claim histories of non-life insurance claims and So et al. (2021) devised an engine to generate synthetic telematics data. However, both engines employ neural networks trained on a single real insurance data set to generate synthetic data that closely mirrors the original data. Conversely, the simulation machine of Avanzi et al. (2021) does not emulate a single data set when generating individual non-life insurance claims. Instead, it allows users to generate a diverse set of scenarios which vary in complexity.

In this chapter, we design a simulation machine that generates synthetic insurance fraud network data. The simulation engine is inspired by and mimics the structure and properties of the non-life motor insurance data used in Óskarsdóttir et al. (2022), which contains both traditional claim and policy(holders) characteristics as well as social network features. When generating the synthetic data, the user has control over several data-generating mechanisms. Both policyholder, contract-specific and claim characteristics as well as the dependence between them can be adjusted. Further, when simulating the number of claims, the individual claim costs and the claim labels (i.e. fraudulent or non-fraudulent), the user can specify the (effect size of the) features that are used in the data-generating model. Specific characteristics of the social network structure and fraud investigation process can be adjusted as well. In addition, the size of the resulting data set and the required level of class imbalance can be set by the user. Hereby, the simulation engine provides researchers with a powerful and valuable tool for evaluating and improving the performance of fraud detection methods across various scenarios. The simulation engine's ability to produce large data sets makes it ideal for machine learning techniques that require large amounts of data. Furthermore, researchers and practitioners can use the engine

to test their (development strategy of) insurance fraud detection models and to investigate the performance of a wide range of analytic methods in tackling the challenges inherent to fraud data sets, e.g. the severe class imbalance and large number of missing labels. Additionally, by examining a specific method or model in these scenarios, researchers can gain a better understanding of its strengths and limitations.

This chapter is structured as follows. In Section 4.2, we discuss the fraud cycle, existing analytic fraud detection strategies and provide a comprehensive overview of social network analytics for fraud detection strategies. Further, we address several challenges that are an integral part of fraud detection research. Section 4.3 delves into the design of the simulation engine and explains how a synthetic data set is generated. Section 4.4 showcases the simulation engine's capabilities by generating and exploring two different types of synthetic data sets. In the first type we introduce a social network effect when simulating the claim labels and in the second type, we omit the social network effect. Additionally, using the artificial data, we provide a practical demonstration of the development and evaluation of a fraud detection model. We conclude the chapter with Section 4.5.

4.2 Fighting fraud with data analytics: strategies, techniques and challenges

In this section, we provide an overview of some conventional and emerging approaches to fraud detection. We highlight the role of analytics in uncovering fraudulent activities and discuss the various challenges that researchers and practitioners face in this domain.

4.2.1 Uncovering fraud: traditional and analytic approaches

In insurance, policyholders file a claim to request a financial compensation for a covered loss. The insurance company retains all relevant information on past and current claims in a database, typically stored in a tabular format. The data set encompasses claim, policyholder, and contract-specific attributes, collectively referred to as traditional claim characteristics. Certain claims, however, are illegitimate and detecting fraudulent claims is essential for insurance companies to prevent financial losses and to protect their policyholders. Hereto, insurers adopt either a traditional, a data-driven or a combined strategy to flag suspicious claims.

Expert-based fraud detection The traditional approach to detect fraud is expert-based (Baesens et al., 2015). This approach is two-fold. First, the insurance company flags certain claims as suspicious. To flag suspicious claims, companies rely on a set of business rules that are based on insights from previous investigations. Second, once a claim is flagged as suspicious, the claim is passed to an expert, who conducts an in-depth investigation to determine whether the claim is fraudulent or not (Warren and Schweitzer, 2018). Hereafter, newly obtained insights from the investigation are used to adjust the procedure for identifying suspicious claims. This is known as the fraud detection cycle.

The expert-based approach, however, has some notable shortcomings (Baesens et al., 2015). It is highly dependent on the manual input and expertise of the expert. In addition, investigating claims is a time-intensive and expensive process. Moreover, the dynamic nature of fraud requires that the rule base to flag suspicious claims needs to be continuously monitored, improved and updated. To address these drawbacks, researchers have developed alternative approaches to detect fraud in a more automated manner (Baesens et al., 2015; Ngai et al., 2011; Albashrawi, 2016; Barman et al., 2016). Notwithstanding, even with the alternative approaches, the inclusion of expert knowledge and input is critical to the success of the fraud detection system.

Fraud analytics The limitations of the expert-based approach prompted researchers to develop data-driven methodologies to detect fraud. In the literature, either supervised or unsupervised learning techniques or a combination of both are employed. Within fraud detection, we use unsupervised learning techniques to identify anomalous behavior (Baesens et al., 2015). There is a plethora of anomaly detection techniques available and Hilal et al. (2022) provides an extensive overview of anomaly detection techniques to detect financial fraud. In our paper, we focus on supervised learning methods which learn from historical, labeled data. There are numerous supervised techniques available that can be utilized to construct a fraud detection model, ranging from logistic regression models to neural networks. A comprehensive literature review of the supervised techniques applied in financial fraud detection can be found in Ngai et al. (2011); Barman et al. (2016); Albashrawi (2016).

One way to tackle insurance fraud detection is by treating it as a binary classification problem. Here, our response variable Y_i can take on only two values: 0 (non-fraud) or 1 (fraud). Further, each claim i has a corresponding covariate vector

$\boldsymbol{x}_i$. The values herein correspond to a set of features that provide information on the policyholder, contract, claim, network and any other relevant features that can assist in identifying fraudulent claims. The general equation of a predictive classification model is

$$P[Y_i = 1|\boldsymbol{x}_i] = f(\boldsymbol{x}_i) \tag{4.1}$$

where $P[Y_i = 1|\boldsymbol{x}_i]$ denotes the probability that claim i is fraudulent given covariate vector $\boldsymbol{x}_i$ and $f(\boldsymbol{x}_i)$ denotes the predictive model. Logistic regression is one of the most popular models to estimate (4.1) (Ngai et al., 2011; Baesens et al., 2015; Barman et al., 2016; Albashrawi, 2016). Other commonly employed techniques include tree-based learners (Kho and Vea, 2017; Roy and George, 2017) and neural networks (Srivastava et al., 2016; Ghobadi and Rohani, 2016).

To develop a fraud detection model, we rely on historical, labeled data of past observed fraud behavior (Baesens et al., 2015). This historical data is commonly derived from the expert judgment of previously investigated claims and serves as the foundation for constructing an effective fraud detection model. By training the fraud detection model on labeled claims, we aim to find hidden patterns that allow us to identify new fraudulent claims.

4.2.2 Enriching traditional claim characteristics with social network data

Both the expert-based and fraud analytics approach commonly rely on traditional claim characteristics stored in a tabular data set. To go beyond this tabular structure, we can rely on social network analytics which extracts information from the relational structure in the data set. As such, we augment the database with supplementary information on the social network structure of the claim and the involved parties. The involved parties are typically the policyholder and the experts involved in the claim (Óskarsdóttir et al., 2022). Certain contracts involve the active participation of brokers, hereby incorporating them into the network structure. Further, depending on the type of insurance, other parties may be present as well. In motor insurance, for example, we commonly also have the auto repair shop that repaired the vehicle (hereafter referred to as the garage). In constructing the network structure, Óskarsdóttir et al. (2022) take a holistic view by integrating information across multiple lines of business. In this chapter, we focus exclusively on the social network structure within one specific insurance product.

Figure 4.1(a) depicts a toy example of a social network consisting of seven claims

and seven involved parties. In this figure, the edges symbolize the connections between the claims and the parties. The claims and parties are represented by circles and the claims in the network are color-coded. Green claims correspond to non-fraudulent claims and fraudulent claims are colored red. There is a notable cluster of fraudulent claims (i.e. c_5, c_6 and c_7) that are strongly interconnected. Party p_7 is connected to all three fraudulent claims and might be the central figure in the criminal network. Via the claims, p_7 is connected to the fraudsters p_5 and p_6. In this example, all fraudsters are connected to each other via one or more fraudulent claims.

To obtain a mathematical representation of the network data, we use a bipartite network of nodes $C \cup P$ and edges E. C denotes the set of all claims and P the set of all parties in the network. E is the set of edges that connect the nodes in C to the nodes in P. The bipartite network $G = (C \cup P, E)$ is undirected (i.e. there is no direction in the edges). We use c_i to denote an individual claim, where $i \in (1, \ldots, n_C)$ and n_C is the total number of nodes in C. p_j denotes an individual party. Here, $j \in (1, \ldots, n_P)$ where n_P is the total number of nodes in P. The network's edges are represented in a weight matrix $\boldsymbol{W}$ of dimension $n_P \times n_C$. Each individual edge carries a certain weight w_{ij} that reflects the strength of the relationship between claim i and party j. If $w_{ij} > 0$, claim c_i is connected to party p_j. We have an unweighted network when all nonzero values in $\boldsymbol{W}$ are equal to one.

We refer to the set of nodes, connected to node c_i via a path of exactly k edges, as the k^{th} order neighborhood of c_i and we denote it as $\mathcal{N}_{c_i}^k$. Hence, the first-order neighborhood of a claim c_i consists of all involved parties

$$\mathcal{N}_{c_i}^1 = \{p_j | w_{ij} \neq 0\} \tag{4.2}$$

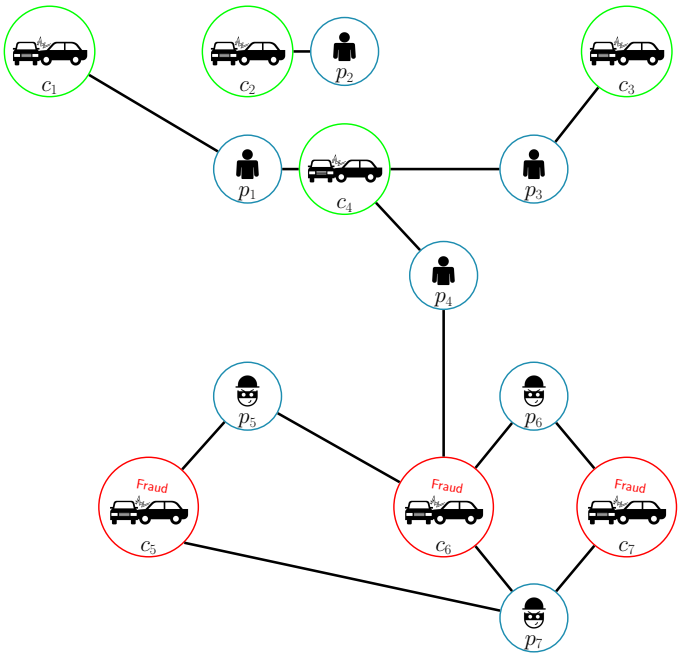
and the second-order neighborhood of c_i

$$\mathcal{N}_{c_i}^2 = \{c_k | p_j \in \mathcal{N}_{c_k}^1 \wedge w_{kj} \neq 0\} \setminus c_i. \tag{4.3}$$

are all the claims connected to the parties in $\mathcal{N}_{c_i}^1$. In an unweighted network, we refer to the number of nodes in a node's first order neighborhood as the degree of the node. For claim c_i, we denote this as d_i and d_j refers to the degree of party p_j. The degree of all claims is summarized in a $n_C \times n_C$ diagonal matrix $\boldsymbol{D}_C$, where $(\boldsymbol{D}_C)_{ii} = d_i \; \forall \; i$. Similarly, the $n_P \times n_P$ diagonal matrix $\boldsymbol{D}_P$ contains the degrees of all parties.

Figure 4.1: A toy example of a social network in an insurance context.

(a) Visualization of the relationships between the claims and the involved parties.



(b) The weight matrix $\boldsymbol{W}$ corresponding to the social network example.

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	
p_1	1	0	0	1	0	0	0	2
p_2	0	1	0	0	0	0	0	1
p_3	0	0	1	1	0	0	0	2
p_4	0	0	0	1	0	1	0	2
p_5	0	0	0	0	1	1	0	2
p_6	0	0	0	0	0	1	1	2
p_7	0	0	0	0	1	1	1	3
	1	1	1	3	2	4	2	d_j / d_i

In this toy example of a social network, we use an unweighted network. Figure 4.1(b) represents the corresponding $\boldsymbol{W}$ that captures the connections in the example. The strong interconnectedness between fraudsters and fraudulent claims is also reflected in $\boldsymbol{W}$. In the lower right corner of $\boldsymbol{W}$ we notice a distinct cluster, which predominantly consists of fraudulent nodes and which reveals a web of fraudsters. We observe another group of connected claims in the upper left corner of $\boldsymbol{W}$. This cluster consists mostly of legitimate claims. Furthermore, there are only a few links connecting fraudulent and non-fraudulent nodes, indicating limited relationships between the two groups.

Homophily One of the fundamental concepts in a network-based fraud detection approach is the concept of homophily (Baesens et al., 2015; Óskarsdóttir et al., 2022). This refers to the tendency of people to form social connections with individuals that are similar to themselves in some way (Newman, 2010). Translated to an insurance context, this means that fraudulent claims are predominantly linked to other fraudulent claims, while non-fraudulent claims tend to be connected to other non-fraudulent claims. Moreover, fraudulent and non-fraudulent claims exhibit a weaker degree of connection with each other.

To assess whether there are patterns of homophily present in the network, we compute the dyadicity and heterophilicity of the network (Park and Barabási, 2007; Baesens et al., 2015). Dyadicity measures the connectedness between nodes with the same label. The higher the dyadicity, the more densely connected the same-label nodes are, compared to what is expected based on a random network configuration. Conversely, heterophilicity assesses the degree of interconnection between nodes with different labels. Networks exhibit high heterophilicity when nodes with different labels show higher interconnectedness compared to what is expected by chance.

In a fraud context, the investigated claims can be labeled as fraudulent (1) or non-fraudulent (0). Claims that are uninvestigated have no label and are referred to as unlabeled. Hence, there are three different labels present in the data set. If our focus is on the identification of dense networks of fraudulent claims, we can adopt a one-versus-all classification strategy and group the unlabeled with the non-fraudulent claims. We denote the total number of fraudulent claims as ${}_1n_C$ and ${}_0n_C$ denotes the total number of non-fraudulent and unlabeled claims. Further, $n_C = {}_1n_C + {}_0n_C$. In this example, we have three types of relationships between claims or so-called dyads. That is: fraudulent claims connected to other fraudulent claims (1 - 1); fraudulent claims linked to non-fraudulent, unlabeled claims (1 - 0)

and non-fraudulent, unlabeled claims connected to non-fraudulent, unlabeled claims (0 - 0). We use m_{11}, m_{10} and m_{00} to refer to the total number of dyads of each kind present in the network and $|E| = m_{11} + m_{10} + m_{00}$, where $|E|$ denotes the number of edges. ρ denotes the probability that two nodes are connected and is empirically calculated in the network as

$$\rho = \frac{2|E|}{n_C(n_C - 1)}. \tag{4.4}$$

If nodes are randomly connected to other nodes irrespective of their labels, the expected values of m_{11} and m_{10} equal (Baesens et al., 2015)

$$\overline{m}_{11} = \frac{{}_1n_C({}_1n_C - 1)\rho}{2} \quad \text{and} \quad \overline{m}_{10} \quad = {}_1n_C(n_C - {}_1n_C)\rho. \tag{4.5}$$

We then calculate the dyadicity $\mathscr{D}$ and heterophilicity $\mathscr{H}$ of the network as

$$\mathscr{D} = \frac{m_{11}}{\overline{m}_{11}}, \tag{4.6}$$

$$\mathscr{H} = \frac{m_{10}}{\overline{m}_{10}}. \tag{4.7}$$

When $\mathscr{D} > 1$, the network is dyadic and fraudulent nodes are more densely connected to each other compared to what we expect by chance and $\mathscr{D} \approx 1$ corresponds to a random network configuration. Here, $\approx$ denotes approximately equal to. We have a heterophobic network if $\mathscr{H} < 1$, indicating that fraudulent claims have fewer connections to non-fraudulent claims than what is expected by chance. In a random network configuration, $\mathscr{H} \approx 1$. In our fictive example depicted in Figure 4.1, the network is dyadic ($\mathscr{D} = 2.5$) and heterophobic ($\mathscr{H} = 0.28$). Consequently, we can infer that our network exhibits homophily as it displays both dyadicity ($\mathscr{D} > 1$) and heterophobia ($\mathscr{H} < 1$). In this toy example, engineering features from the social network potentially enables us to identify collaborating fraudsters that try to hide their tracks.

BiRank algorithm In a homophilic network (i.e. where $\mathscr{D} > 1$ and $\mathscr{H} < 1$), we can potentially uncover fraud by inspecting claims that are closer and more densely connected to known fraudulent claims. To evaluate the proximity to fraudulent claims, a suitable metric is needed. One effective approach is to employ the BiRank algorithm (He et al., 2017). This algorithm is an extension of the personalized PageRank algorithm (Page et al., 1999) and is specifically designed for bipartite networks. Using BiRank, we rank claims with respect to their exposure to known

fraudulent claims (Óskarsdóttir et al., 2022).

The scores of nodes c_i and p_j are calculated iteratively as

$$c_i = \sum_{j=1}^{n_P} w_{ij} p_j \quad \text{and} \quad p_j = \sum_{i=1}^{n_C} w_{ij} c_i$$

where c_i and p_j denote the scores of nodes c_i and p_j, respectively. To ensure convergence and stability, the scores are normalized using

$$c_i = \sum_{j=1}^{n_P} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} p_j \quad \text{and} \quad p_j = \sum_{i=1}^{n_C} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} c_i. \tag{4.8}$$

This normalization lessens the contribution of high-degree nodes and gives better quality results (He et al., 2017). To steer the scoring towards fraudulent claims, we incorporate query vectors in the scoring process. These query vectors encode our prior belief on the nodes' importance. We use $\boldsymbol{c}^0$ and $\boldsymbol{p}^0$ to denote the query vectors for the claims and parties, respectively. Further, c_i^0 and p_j^0 represent the individual entries in the vectors $\boldsymbol{c}^0$ and $\boldsymbol{p}^0$. We adjust (4.8) to

$$c_i = \alpha \sum_{j=1}^{n_P} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} p_j + (1-\alpha) c_i^0 \quad \text{and} \quad p_j = \beta \sum_{i=1}^{n_C} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} c_i + (1-\beta) p_j^0. \tag{4.9}$$

where $\alpha \in [0,1]$ and $\beta \in [0,1]$ are adjustable parameters. The values for α and β regulate the relative emphasis given to the network structure and the query vector. We rewrite (4.9) in matrix form

$$\mathbf{c} = \alpha \boldsymbol{S}\mathbf{p} + (1-\alpha)\mathbf{c}^0 \quad \text{and} \quad \mathbf{p} = \beta \boldsymbol{S}^T\mathbf{c} + (1-\beta)\mathbf{p}^0. \tag{4.10}$$

Here, $\boldsymbol{S} = \boldsymbol{D}_C^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}_P^{-\frac{1}{2}}$ denotes the symmetrically normalized weight matrix. We start the algorithm by randomly initializing the ranking vectors $\boldsymbol{c}$ and $\boldsymbol{p}$. Hereafter, we iteratively compute the node scores until convergence.

We encode information about known fraudulent claims into the query vector $\boldsymbol{c}^0$ to rank the nodes' scores towards fraudulent claims. When the claim is fraudulent, we set $\boldsymbol{c}_i^0 = 1$ and $\boldsymbol{c}_i^0 = 0$ otherwise. We define $\mathbf{p}^0 \equiv \mathbf{0}$, since only claims can be fraudulent and not parties. We set $\beta = 1$ since we do not include prior information on the parties. We adjust (4.10) to

$$\mathbf{c} = \alpha S\mathbf{p} + (1-\alpha)\mathbf{c}^0 \quad \text{and} \quad \mathbf{p} = S^T\mathbf{c}.$$

The iterative procedure to compute the fraud scores is summarized in Algorithm 3. The BiRank algorithm stops when the difference between two successive iterations is below a certain threshold or when we exceed the maximum number of iterations.

Algorithm 3: BiRank algorithm for computing fraud scores in a network of insurance claims and parties (Óskarsdóttir et al., 2022). Adapted from Algorithm 1 in He et al. (2017). We omit the query vector $\mathbf{p}^0$ and set $\beta = 1$.

Input: Weight matrix $\boldsymbol{W}$, query vector $\mathbf{c}^0$ and hyperparameter $\alpha = 0.85$;
Output: Ranking vectors $\mathbf{c}$ and $\mathbf{p}$;
Symmetrically normalize $\boldsymbol{W}$: $\boldsymbol{S} = \boldsymbol{D}_P^{-\frac{1}{2}} \boldsymbol{W} \boldsymbol{D}_C^{-\frac{1}{2}}$;
Randomly initialize $\mathbf{c}$ and $\mathbf{p}$;
while *stopping criteria is not met* **do**
  $\mathbf{c} \leftarrow \alpha S\mathbf{p} + (1-\alpha)\mathbf{c}^0$;
  $\mathbf{p} \leftarrow \boldsymbol{S}^T\mathbf{c}$;
return $\mathbf{c}$ *and* $\mathbf{p}$;

Network featurization Next, we engineer several social network features from the network structure, the labels and the scores resulting from the BiRank algorithm. These features capture the information that is embedded in the network of the claims and can be integrated into a tabular dataset alongside the individual characteristics of each claim. The social network features then represent an additional source of information that we can use in our fraud analytics models (see Section 4.2.1). We divide the network features into two groups, the fraud-score based features and the neighborhood based features (see Table 4.1). The results from the BiRank algorithm are used to compute the fraud-score based features. Here, we look at the claim's fraud score and the distribution of the fraud scores in, for instance, its first and second order neighborhood. To summarize these distributions, we can rely on (robust) central tendency measures such as the median or midmean (Tukey, 1977). Further, we engineer neighborhood based features that capture the surrounding network structure of each claim. Here, we can compute the size of a claim's first and second neighborhood for instance.

4.2.3 Challenges within fraud analytics

When developing an analytic model for fraud detection, we encounter several challenges that are inherent to research on fraud. One of the main challenges is the infrequent nature of fraud, which leads to highly imbalanced data sets (Baesens

Table 4.1: Fraud-score and neighborhood based features, partially based on the feature engineering process from Óskarsdóttir et al. (2022).

	Name	Order	Description
Fraud-score	`scores0`	0	The node's fraud score as determined via BiRank
	`n1.q1`	1	The first quartile of the empirical distribution of the fraud scores in the node's first order neighborhood
	`n1.med`	1	The median of the empirical distribution of the fraud scores in the node's first order neighborhood
	`n1.midmean`	1	The midmean (or interquartile mean) of the empirical distribution of the fraud scores in the node's first order neighborhood
	`n2.q1`	2	The first quartile of the empirical distribution of the fraud scores in the node's second order neighborhood
	`n2.med`	2	The median of the empirical distribution of the fraud scores in the node's second order neighborhood
	`n2.midmean`	2	The midmean (or interquartile mean) of the empirical distribution of the fraud scores in the node's second order neighborhood
Neighborhood	`n1.size`	1	The number of nodes in the node's first order neighborhood
	`n2.size`	2	The number of nodes in the node's second order neighborhood
	`n2.ratioFraud`	2	The number of known fraudulent claims in the node's second order neighborhood divided by `n2.size`
	`n2.ratioNonFraud`	2	The number of known non-fraudulent claims in the node's second order neighborhood divided by `n2.size`
	`n2.binFraud`	2	A binary value indicating whether there is a known fraudulent claim in the node's second order neighborhood

et al., 2015; Jensen, 1997; West and Bhattacharya, 2016; Thabtah et al., 2020). This imbalance creates a bias towards the majority class for certain analytic techniques, resulting in compromised fraud detection performance. Within fraud research, we commonly tackle the class imbalance problem by employing resampling techniques such as under- and over-sampling, SMOTE or ROSE (Baesens, 2023; Subudhi and Panigrahi, 2020; Sundarkumar and Ravi, 2015; Van Vlasselaer et al., 2016; Óskarsdóttir et al., 2022). A second challenge is the dynamic nature of fraud (Baesens et al., 2015; Baesens, 2023; West and Bhattacharya, 2016). To remain undetected, fraudsters continuously adapt their behavior and tactics. Consequently, it is essential to detect fraud as soon as possible and to tweak or rebuild the fraud detection model when necessary. Hereto related is the computational efficiency of the fraud detection model (Baesens et al., 2015; West and Bhattacharya, 2016) which represents yet another challenge. With the rapid advancement of machine learning, new methods are constantly emerging, adding to the ever-growing repertoire of techniques available. Further, given the high cost of fraud, it is crucial to detect fraudulent activities instantly. Existing analytic methods for fraud detection are predominantly evaluated based on their accuracy, often overlooking the aspect of computational efficiency. For an accurate and reliable comparison of competing methods, it is essential to evaluate their performance on a set of benchmark data sets. Most studies, however, do not share their data sets due to their sensitive nature. The lack of publicly available data hinders the reproducibility of research (Baesens, 2023) and gives rise to a fifth challenge (Pourhabibi et al., 2020; West and Bhattacharya, 2016). The sixth and final challenge arises from the disproportionate misclassification cost and the specific performance criteria to evaluate the model (West and Bhattacharya, 2016; Baesens, 2023). Analytic fraud models are commonly assessed using performance measures that evaluate the predictive performance. Notwithstanding, wrongly classifying claims has financial implications. Depending on the allocated budget for the fraud investigation process, wrongly classifying a fraudulent claim as legitimate can be considerably more expensive than the reverse. Consequently, it might be more appropriate to compare models in terms of their monetary performance (Baesens, 2023).

4.3 Simulation engine

Our proposed simulation engine addresses one of the key challenges within fraud research. That is, the limited availability of publicly accessible data. In this section,

we provide an in-depth overview of the data generation process and the architecture of the simulation engine. We outline the sequential steps taken to generate a realistic and representative insurance fraud network data set.

The resulting synthetic data set is structured in a tabular format, with each row representing a unique claim and its corresponding attributes. The columns in the data set capture items such as policyholder characteristics, traditional claim features, involved parties and social network features. These are the attributes typically used for fraud analytics. Further, for each claim we have two different types of labels. The first is the true label of the claim, indicating if it is fraudulent or not. This label is determined by the fraud generating model. The second type of label is the outcome of the fraud investigation, which can either take on the value fraudulent, non-fraudulent or uninvestigated. This variable is typically the one we have available in insurance fraud data sets.

Implementation The simulation engine is available as open-source R package on Github at `https://github.com/BavoDC/iFraudSimulator`. An extensive overview of the default configuration is described in Appendix C.1 as well as in the package's documentation.

Architecture We generate a synthetic, tabular data set in seven consecutive steps (see Figure 4.2). We start by generating the policyholder characteristics (step 1). Hereafter, we simulate the contract-specific attributes per policyholder (step 2). We use the policyholder and contract-specific features as input for our data-generating claim frequency model and generate the number of claims per contract (step 3). Next, we simulate the individual claim amounts using a data-generating claim severity model (step 4). Similarly to step 3, we use the policyholder and contract-specific characteristics as input for the data-generating model. Subsequently, we combine all simulated claims and their characteristics into a tabular data set and we proceed with the observations that have at least one claim. In step 5, we generate the network structure of the claims by connecting each claim to different types of parties. Next, we engineer the social network features and generate the claim labels which represent the ground truth of the claim (i.e. fraudulent or non-fraudulent). We replicate the fraud investigation process in step 6. Hereby, we generate the label that is commonly available in fraud data sets (see Section 4.2). This label expresses whether the claim has been investigated for fraud and what the outcome of that investigation was. We conclude the synthetic data generation with step 7 where we

merge all simulated data.

The simulation engine offers a range of customizable features. In all seven steps, several parameters can be adjusted. Moreover, it allows for dependencies between certain features. For example, we can specify a negative dependence between the age and the value of the car. Consequently, older cars will be characterized by a lower value. By allowing for dependencies between features, the synthetic data captures more realistic and nuanced relationships among variables and more closely mirrors real-world data sets. Following, we discuss the seven consecutive steps in detail.

4.3.1 Policyholder and contract-specific characteristics

Prior to the synthetic data generation, the user can adjust several parameters that determine the characteristics of the synthetic data (see Appendix C.1 for the default configuration). Hence, the user has full control over the data-generating mechanics. One of the adjustable parameters is the number of policyholders n_{ph}, which determines the size of the resulting data set. By default, $n_{ph} = 10\,000$. We use $i = 1, \ldots, n_{ph}$ as an index for the policyholders. In addition to n_{ph}, the user can also specify the total number of experts n_{exp}, garages n_{gar}, brokers n_{bro} and other person(s) involved in the claim n_{per}. These parameter values will govern the size of the social network.

We start the synthetic data generation by simulating the policyholder characteristics for n_{ph} policyholders (step 1 in Figure 4.2). Here, we generate features such as the age and gender of the policyholder and the number of contracts. We use $\texttt{NrContractsPH}_i$ to denote the number of contracts of policyholder i and use $j = 1, \ldots, \texttt{NrContractsPH}_i$ as an index for the contracts. Per policyholder, we also generate the number of years since the inception of the first contract and refer hereto as the exposure w_i. By default, we set the average exposure to five years and the maximum to 20 years. Additionally, we simulate the contract-specific exposure w_{ij}. In the synthetic data set, we consolidate the multiple years of coverage into a single contract to simplify the data structure and analysis. That is, by default we allow $w_{ij} > 1$. Table 4.2 gives an overview of the different attributes that are generated. The first column in this table depicts the variable name, the second the variable type and the third column contains the feature description. The last column of Table 4.2 specifies which generator is used to simulate the feature values. For certain features, the user can specify the range of the feature values. When generating the data, we ensure that all simulated values fall within this prespecified range (see

Figure 4.2: Roadmap of the simulation engine.

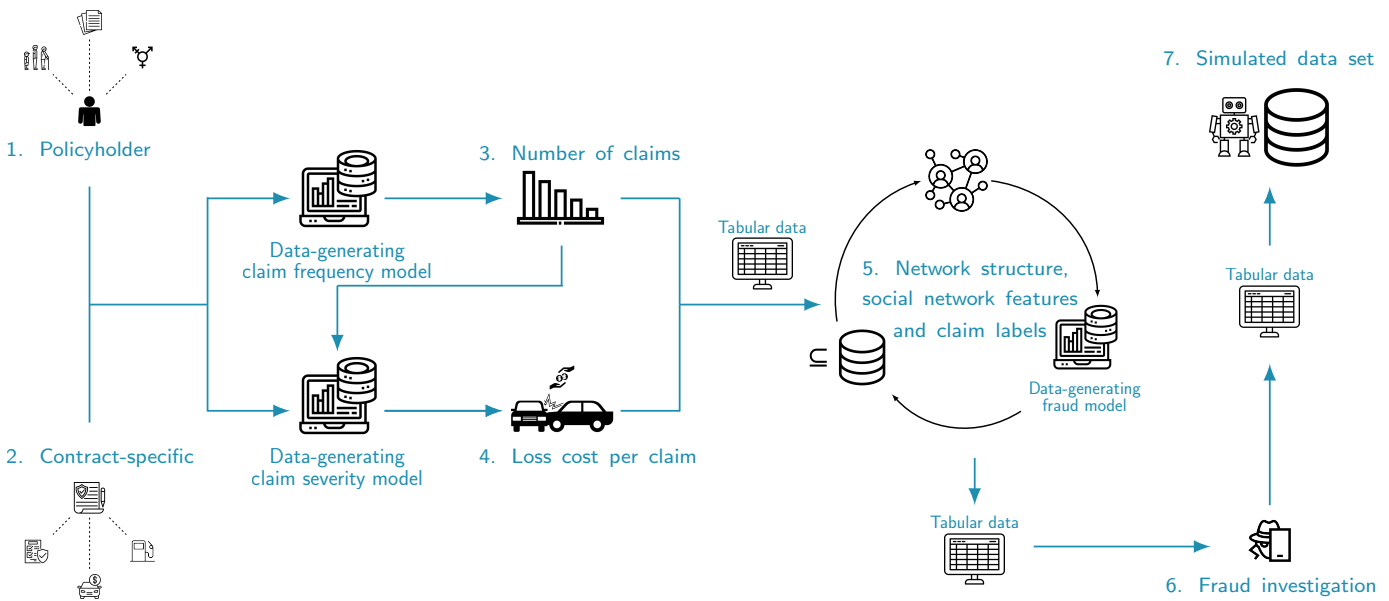


Table 4.2: The policyholder and contract-specific characteristics, along with the generator used to simulate the feature values.

	Variable	Type	Description	Generator
Policyholder	`IDPH`	Continuous	Unique ID to identify the policyholder (index $i = 1, \ldots, n_{ph}$)	
	`AgePH`	Continuous	Age of the policyholder in years. Default range is [18, 80]	$\mathcal{N}(40, 15)$
	`GenderPH`	Categorical	Gender of the policyholder: `female` ($u_i \leqslant 0.28$), `male` ($u_i > 0.29$) or `non-binary` ($0.28 \leqslant u_i \leqslant 0.29$)	$u_i \sim U(0, 1)$
	`ExpPH`	Continuous	Time since inception of the first contract of the policyholder, in years	$\mathcal{N}(5, 1.5)$
	`RateNrContracts`	Continuous	Rate parameter λ_i for generating the number of contracts	$\lambda_i = 0.25(1.05 - 2.5 \times 10^{-6} \times \texttt{AgePH}_i + 0.0025 \times \texttt{AgePH}_i^2 - 2.65 \times 10^{-5} \times \texttt{AgePH}_i^3)$
	`NrContractsPH`	Ordinal	Number of contracts. Default range is [1, 5]	$\text{Poi}(\lambda_i)$
Contract-specific	`ContractID`	Continuous	Unique ID to identify the contract (index $j = 1, \ldots,$ $\texttt{NrContractsPH}_i$)	
	`ExpPHContracts`	Continuous	Duration or exposure of the contract, in years	if $\texttt{NrContractsPH}_i > 1$: $\texttt{ExpPH}_i - \text{U}(0, \texttt{ExpPH}_i/2)$ else: $\texttt{ExpPHContracts}_{ij} = \texttt{ExpPH}_i$
	`AgeCar`	Continuous	Age of the vehicle in years	$\max(\mathcal{N}(7.5, \sqrt{5}), \texttt{ExpPHContracts}_{ij})$
	`OrigValueCar`	Continuous	Original value of the vehicle	$\text{Exp}(\lambda_i/\texttt{NrContractsPH}_i)\upsilon$
	`ValueCar`	Continuous	Current value of the car	$\texttt{OrigValueCar}_{ij}(1-\delta)^{\texttt{AgeCar}_{ij}}$
	`Coverage`	Categorical	Type of coverage provided by the insurance company: `TPL` = only third party liability, `PO` = partial omnium = TPL + limited material damage, `FO` = full omnium = TPL + comprehensive material damage.	$\text{Multinomial}(1, \pi_{\text{TPL}}, \pi_{\text{PO}}, \pi_{\text{FO}})$ (see Appendix C.3)
	`Fuel`	Categorical	Type of fuel of the vehicle: `Gasoline/LPG/Other` (0) or `Diesel` (1)	Bernoulli(0.3)
	`BonusMalus`	Ordinal	Level occupied in bonus-malus scale of the insurance company	$\lfloor \mathcal{G}(1, 1/3) \rfloor$
Claim	`ClaimAge`	Integer	Number of months from beginning of contract to the date of the incident	$\min(\lfloor \text{Exp}(0.25) \rfloor, \lfloor \texttt{ExpPHContracts}_{ij} * 12 \rfloor)$
	`ClaimDate`	Continuous	Number of years between the start of the contract and the claim's filing date	$\max(U(0, \texttt{ExpPHContracts}_{ij}), \texttt{ClaimAge}_{ijk}/12)$
	`Police`	Categorical	Whether police was called when the incident happened: `no` (0) or `yes` (1)	Bernoulli(0.25)
	`nPersons`	Integer	Number of other persons involved in the claim (see Section 4.3.3). Range is [0, 5]	$S \xleftarrow{\pi_p} x$

$\mathcal{N}$ denotes the normal distribution, U the uniform distribution, Poi the Poisson distribution, $\mathcal{G}$ the Gamma distribution and Exp denotes the exponential distribution. $\upsilon = 25 \times 10^3$ if $\texttt{GenderPH}_i = \texttt{male}$, $\upsilon = 20 \times 10^3$ if $\texttt{GenderPH}_i = \texttt{female}$ and $\upsilon = 22.5 \times 10^3$ if $\texttt{GenderPH}_i = \texttt{non-binary}$. δ is the depreciation rate (see Table 4.3). We use $\lfloor \cdot \rfloor$ to depict the floor function. $S \xleftarrow{\pi_p} x$ denotes that a random sample x is drawn from the set $S = (0, 1, 2, 3, 4, 5)$ with corresponding probability $\pi_p = (0.025, 0.6, 0.2, 0.1, 0.1, 0.025)$.

Appendix C.2). Hereby, we avoid generating implausible or invalid values. For the policyholder's age, for example, the default range is [18, 80]. Consequently, none of the policyholders will be younger than 18 (the legal driving age in many countries) or older than 80. Once all policyholder characteristics are generated, we proceed to simulate the contract-specific features such as the age of the car and the type of coverage (see Table 4.2).

Dependence structure The simulation engine allows to specify a dependency between different features. To generate the dependency, we rely on copulas (Denuit et al., 2005). In our simulation engine, we restrict ourselves to the bivariate Ali-Mikhail-Haq (AMH) (Kumar, 2010) and Frank copula (Denuit et al., 2005). We use θ to denote the parameter that controls the dependence.

Table 4.3 presents an overview of the dependencies and the method used to incorporate them. Within insurance, we commonly have variables that are correlated (Goldburd et al., 2016). Consequently, by allowing for dependencies, we can create a more realistic data set. For example, in real life data sets we commonly observe that older cars are worth less compared to newer cars. In our engine, we incorporate this negative dependency between the age of the car and its value by using a Frank copula with $\theta = -25$.

Table 4.3: Overview of the dependencies between the variables.

Variables	Dependency
`AgePH` and `GenderPH`	Weak negative dependence, introduced using AMH copula with $\theta = -0.15$
`AgePH` and `ExpPH`	Weak positive dependence, introduced using AMH copula with $\theta = 0.15$
`AgePH` and `NrContracts`	Convex function (see Table 4.2) and positive dependence between `AgePH` and `NrContracts`, introduced using AMH copula with $\theta = 0.95$
`AgeCar` and `OrigValueCar`	Negative dependence, introduced using Frank copula with $\theta = -25$
`OrigValueCar` and `ValueCar`	The depreciation rate $\delta = 0.15$ for cars whose original value $< 30\,000$ and $\delta = 0.075$ when the original value $\geqslant 30\,000$ (see Table 4.2 and Storchmann (2004)).
`Coverage` and `ValueCar`, `AgeCar`, `AgePH`	A dependency is introduced using a multinomial logistic regression model (see Appendix C.3)

4.3.2 Claim frequency and claim severity

Next, we proceed to simulating the number of claims N_{ij} for policyholder i on contract j and the individual claim costs L_{ijk}. We use $k = (1, \ldots, N_{ij})$ as an index for the claims. Hereto, we employ the frequency-severity approach (Ohlsson and Johansson, 2010; Frees et al., 2014) where we model the claim frequency and claim severity separately. To simulate the number of claims and the claim costs as a function of the policyholder and contract-specific characteristics, we rely on the generalized linear model (GLM) framework (McCullagh and Nelder, 1999). We use a Poisson GLM with log link as the data-generating model for the number of claims (Ohlsson and Johansson, 2010; Quijano Xacur and Garrido, 2015)

$$N_{ij} \sim \text{Poi}(w_{ij} \exp({}_{cf}\boldsymbol{x}_{ij}^{\top} \boldsymbol{\beta}_{cf})). \tag{4.11}$$

To generate N_{ij}, we take a random draw from $\text{Poi}(w_{ij} \exp({}_{cf}\boldsymbol{x}_{ij}^{\top} \boldsymbol{\beta}_{cf}))$, with ${}_{cf}\boldsymbol{x}_{ij}$ the covariate vector and $\boldsymbol{\beta}_{cf}$ is the corresponding parameter vector. The exposure w_{ij} of the contract is included as an offset term and the subscript cf stands for claim frequency. In our simulation engine, the user can specify which features should be included in ${}_{cf}\boldsymbol{x}_{ij}$ and the features' effect size can be adjusted via $\boldsymbol{\beta}_{cf}$. As such, the user can control the relation between the N_{ij}'s and the policyholder and contract-specific characteristics (see Appendix C.4 for the default specification of the claim frequency model).

Hereafter, we generate the claim-specific characteristics (see Table 4.2). For example, we simulate the duration in months since the beginning of the contract until the date of the incident. Hereby, we create the claim-specific information that is typically available in fraud insurance data sets and that is used in fraud detection models (see, for example, Óskarsdóttir et al. (2022)).

Next, we proceed with generating the cost L_{ijk} of claim k under contract j of policyholder i. The data-generating model for the claim amounts L_{ijk} is driven by a gamma GLM with log link

$$L_{ijk} \sim \mathcal{G}(\alpha, \alpha / \exp({}_{cs}\boldsymbol{x}_{ij}^{\top} \boldsymbol{\beta}_{cs} + N_{ij}\zeta)) \tag{4.12}$$

where $\mathcal{G}$ denotes the gamma[1] distribution and α the shape parameter, which we set to 0.25. The subscript cs stands for claim severity and the parameter ζ controls

[1]For the gamma distribution, we use the parameterization with the density function $f(x) = \tau^{\alpha} x^{\alpha-1} \exp(-\tau x)/\Gamma(\alpha)$ where $\tau_i = \alpha / \exp({}_{cs}\boldsymbol{x}_{ij}^{\top} \boldsymbol{\beta}_{cs} + N_{ij}\zeta)$ and $\Gamma(\cdot)$ denotes the gamma function.

the dependency between the claim frequency and claim severity (Frees et al., 2011; Garrido et al., 2016). Within the frequency-severity approach, we commonly assume that $\zeta = 0$ (i.e. the claim frequency and claim severity are independent). We specify 50 as a lower limit for L_{ijk} to prevent generating implausible low claim amounts. Consequently, when $L_{ijk} < 50$ we replace it with a randomly drawn value from $U(50, 150)$ to ensure that a low yet realistic claim amount is generated. As with the data-generating claim frequency model (see equation (4.11)), we can specify which features are included in ${}_{cs}\boldsymbol{x}_{ij}$ as well as their effect size via $\boldsymbol{\beta}_{cs}$ (see Appendix C.4 for the default model specification).

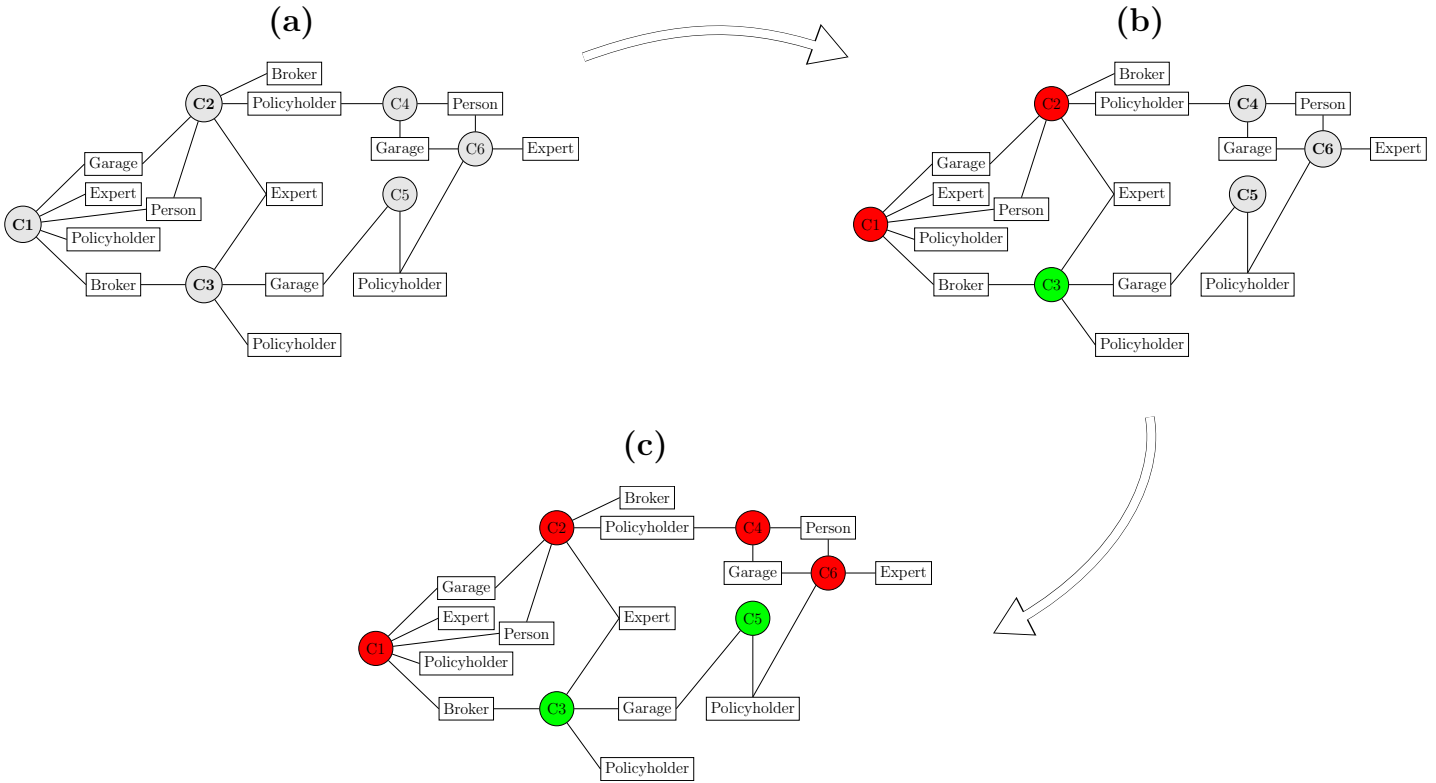
4.3.3 Constructing the social network structure and simulating fraudulent claims

Our next objective is to generate the social network structure that links claims to parties and to other claims. Within motor insurance each claim is linked to a policyholder and a garage. Other parties involved in the claim may include brokers and persons other than the policyholder. Experts are involved in the process only when the claim amount exceeds a certain threshold (KBC Brussels, n.d.). Insurance companies commonly handle minor losses without the involvement of an expert to inspect the damage or injury. Our goal is to enhance the simulated claims with a network structure similar to the one depicted in Figure 4.3(a). In this figure, the circles depict claims and the rectangles parties.

To accomplish this, we create a set A_p for each party type, such as the set of garages A_g, brokers A_b, experts A_e, and other persons A_o. These sets represent the specific parties of each type within the larger set of all possible parties, denoted as $P = A_g \cup A_b \cup A_e \cup A_o$. The size of a specific set A_p is determined by the corresponding user-specified parameter (see Section 4.3.1 and Appendix C.1). For example, if the number of garages n_{gar} is set to 150, the simulation engine will create a set A_g of 150 unique garages. For each claim, we then randomly select one (or multiple when we connect the claim to other persons, see Table 4.2) member from A_p to link the claim to a specific party. By repeating this procedure for each type of party, we create a social network structure where every claim is connected to different (types of) parties (see Figure 4.3(a)). As a rule, we do not link the claim to an expert when $L_{ijk} < 250$ (KBC Brussels, n.d.).

Next, we proceed with generating the claim label. The data-generating fraud

Figure 4.3: Example of a social network structure, which illustrates the desired connectivity we want to obtain in our synthetic data set. Each claim is linked to specific parties, and as a result, claims that share the same party are connected to each other in the network. The rectangles depict the parties and the circles the claim. Red claims are fraudulent claims, green claims legitimate claims and the gray claims represent unlabeled claims.



model is a logistic regression model

$$Y_{ijk} \sim \text{Bern}\left(\pi_{ijk}\right) \quad \text{and} \quad \pi_{ijk} = \frac{e^{{}_0\beta_f + {}_f\boldsymbol{x}_{ijk}^\top \boldsymbol{\beta}_f}}{1 + e^{{}_0\beta_f + {}_f\boldsymbol{x}_{ijk}^\top \boldsymbol{\beta}_f}}. \tag{4.13}$$

Here, Bern denotes a Bernoulli distribution and Y_{ijk} is a binary variable indicating if the k^{th} claim of the j^{th} contract of policyholder i is fraudulent ($Y_{ijk} = 1$) or not ($Y_{ijk} = 0$). The subscript f stands for fraud and ${}_0\beta_f$ is the intercept term. The relation between the fraud label and the features is adjusted via the covariate vector ${}_f\boldsymbol{x}_{ijk}$ and the parameter vector $\boldsymbol{\beta}_f$ (see Appendix C.5 for the default specification). In the default model specification, there are no interaction terms or non-linear effects. To introduce interactions between covariates, we include the interactions in the model formula (equation (4.13)) while non-linear effects can be incorporates using splines. (Wood, 2011, 2017).

We generate the claim labels in an iterative manner and this process is visualized in Figure 4.3. In this figure, the label of the claims is color-coded. Gray stands for unlabeled, red for fraudulent and green for non-fraudulent. Panel (a) represents the network at initialization, when all claims are unlabeled. We have no fraud-related information at this point and hence, no values for the social network features that rely on this information (e.g., the ratio of known fraudulent claims in the second order neighborhood). Consequently, at initialization, we remove all fraud-score and neighborhood based features (see Table 4.1) from ${}_f\boldsymbol{x}_{ijk}$ and $\boldsymbol{\beta}_f$ in (4.13). Next, we take a random subset, equal in size to 1% for example, of all simulated claims. By deliberately taking a small subset of the data, we ensure that only a limited proportion of the claim labels is generated without effect of the fraud-score and neighborhood based features. In Figure 4.3(a), this subset consists of claims C1, C2 and C3. To generate the claim labels, we take a random draw from Bern(π_{ijk}) (see (4.13)). As such, we generate our first set of labeled claims (see Figure 4.3(b)). This enables us to compute the values for the fraud-score and neighborhood based features. Consequently, from here on out, we include these features in ${}_f\boldsymbol{x}_{ijk}$ and $\boldsymbol{\beta}_f$. To generate the labels of the remaining claims, we again take a random subset of unlabeled claims. In Figure 4.3(b), this subset corresponds to claims C5 and C6. The size of this random subset can be set by the user. By default, this is equal to 10% of all simulated claims. We combine this subset with the unlabeled claims in the 2^{nd} order neighborhood of the fraudulent claims in the previous iteration (i.e. C4 in Figure 4.3(b)). In doing so, we ensure that every subset includes unlabeled claims

that are connected to fraudulent claims and that we propagate fraud through the network. We engineer the social network features for all claims in the subset and we take random draws from Bernoulli(π_{ijk}) to generate the claim labels (Figure 4.3(c)).

Algorithm 4 is a generalization of the process illustrated in Figure 4.3. We use this iterative algorithm to simulate the claim labels Y_{ijk} and each iteration consists of three steps. First, we take a random subset of the unlabeled claims and we combine this subset with all unlabeled claims in the 2^{nd} order neighborhood of the fraudulent claims in the previous iteration. Second, we engineer the social network features for all claims in the subset. Third, we take random draws from Bernoulli(π_{ijk}) to generate the claim labels (Figure 4.3(c)). This concludes one iteration and we repeat the algorithm until all claims are labeled.

Algorithm 4: Iterative algorithm to simulate the claim labels

Model: $Y_{ijk} \sim \text{Bern}(\pi_{ijk})$
Initialization: Remove fraud-score and neighborhood based features from $({}_f\boldsymbol{x}_{ijk}, \boldsymbol{\beta}_f)$ in the first iteration and generate the initial claim labels using (4.13)
repeat
1. Take a subset of the simulated database: a random sample of unlabeled claims combined with the unlabeled claims in 2^{nd} order neighborhood of fraudulent claims in the previous iteration;
2. Construct the social network features for the claims in this subset;
3. Generate the claim label using the data-generating logistic regression model in (4.13) ;

until *all claims are labeled*;

The user can specify which features are included in ${}_f\boldsymbol{x}_{ijk}$ and determine their effect size in $\boldsymbol{\beta}_f$. Consequently, the user has the flexibility to activate or deactivate specific feature effects and to control their strength. By including social network features in ${}_f\boldsymbol{x}_{ijk}$ and via the specified effect size in $\boldsymbol{\beta}_f$, for example, we determine to which extent the network exhibits patterns of homophily. The greater the corresponding effect size in $\boldsymbol{\beta}_f$, the more densely connected fraudulent claims will be. Conversely, we can turn off the social network effect by omitting the social network features from ${}_f\boldsymbol{x}_{ijk}$. Further, we can set the desired level of class imbalance. Hereto, our simulation engine determines which value for ${}_0\beta_f$ results in the target class imbalance (see Appendix C.5 for detailed information).

4.3.4 Replicating the expert-based fraud detection approach

In a real life fraud data set, we typically have historical, labeled data that is the result of an investigation by a fraud expert (see Section 4.2.1). In our simulation engine, we replicate the two steps of this fraud detection approach to obtain these labels. First, we flag claims as suspicious based on a set of business rules which can be defined by the user. Hence, an alert will be triggered for claims that meet the criteria outlined in the business rules. By default, we flag claim k under contract j of policyholder i as suspicious if it satisfies one of the following criteria: a) the claim is filed within one year of the most recent claim (i.e. $\texttt{ClaimDate}_{ijk} - \texttt{ClaimDate}_{ij(k-1)} \leqslant 1$); b) the individual claim amount $L_{ijk} > 75\%$ of $\texttt{ValueCar}_{ijk}$ or c) the cumulative claim amount $\sum_{l=1}^{k} L_{ijl} > 200\%$ of $\texttt{ValueCar}_{ijk}$. In reality, these claims are passed to an expert who performs an in-depth investigation. Following the investigation, the expert judgement determines whether the claim is legitimate or not. We simulate the expert judgement in the second step as follows. For a claim that is flagged by the business rules in step one, we first look at its ground truth label Y_{ijk}. If $Y_{ijk} = \texttt{non-fraudulent}$, we take a random draw from $Y_{ijk}^{\texttt{expert}} \sim \text{Bern}(0.01)$. Hence, when the claim is legitimate, we have a 99% probability that the expert will label the claim as non-fraudulent. Conversely, if $Y_{ijk} = \texttt{fraudulent}$, we randomly draw from $Y_{ijk}^{\texttt{expert}} \sim \text{Bern}(0.99)$. Thus, for fraudulent claims, there is a 99% probability that the expert will classify them as fraudulent as well. Further, claims that are not flagged by the business rules obtain the label `uninvestigated`. Hereby, we create all three labels that are typically available in an insurance fraud data set: `non-fraudulent`, `fraudulent` or `uninvestigated`. By following the procedure as outlined above, we acknowledge and reflect the inherent missing information and uncertainties that exist in real-life data. That is, the expert-based approach is not entirely infallible (Baesens et al., 2015). Claims that are judged to be non-fraudulent by the investigation may in reality be fraudulent and vice versa. In addition, we acknowledge and replicate the phenomenon of having a substantial proportion of uninvestigated claims that are unlabeled. However, with a sensitivity and specificity of 99% we assume nearly perfect judgement. To allow for other scenarios, we allow users to define both the sensitivity and specificity of the expert judgment. As such, users can regulate the precision of the expert.

4.4 Generating synthetic fraud network data: illustrations

In this section, we illustrate the capabilities of our simulation engine. We specifically highlight the impact of social network features on the resulting synthetic data sets. Hereto, we generate and analyze two different types of data sets. One where we include a moderately strong social network effect and one where we exclude it. Additionally, we provide an illustrative example of the construction and evaluation of a fraud detection model using a synthetically generated data set. We explore to which extent the constructed model is able to identify fraudulent claims that are not investigated and labeled by the expert.

4.4.1 The impact of social network features on the synthetic data

We generate two different types of data sets. In the first type of data set we introduce a moderately strong social network effect in the claim label generation (see Section 4.3.3). We denote this type of data set as $\mathcal{D}^{Network}$. Table 4.4 depicts the specification of the effect sizes used in the simulation of $\mathcal{D}^{Network}$. We include various types of features to generate a realistic and representative synthetic data set. These features encompass the policyholder, claim-specific, and social network characteristics. In order to replicate the social dynamics of fraud, we assign a strong effect size for the social network features `n1.size`, `n2.size` and `n2.ratioFraud`. Hereby, we create a synthetic data set where the network structure exhibits patterns of homophily. Conversely, in the second type of data set $\mathcal{D}^{Non-network}$, we exclude all network-related features from the data-generating fraud model (see Table 4.4). As such, we create a data set where fraud is not influenced by social interactions or network dynamics. The claim label generation is solely driven by policyholder and claim-specific characteristics. The selected set of policyholder and claim-specific features is identical in $\mathcal{D}^{Network}$ and $\mathcal{D}^{Non-network}$, as well as the effect sizes of these features (see Table 4.4).

For both types of data sets, the number of policyholders is set to 200 000 and the target class imbalance (i.e. the ratio of the number of fraudulent claims to the total number of claims) to 2%. All other settings remain at their default values (see Appendix C.1). We generate 100 data sets of each type and explore the distribution of the claim labels across these simulated data sets. Hereto, we calculate the frequency

Table 4.4: Specification of the data-generating fraud model in $\mathcal{D}^{Network}$ and $\mathcal{D}^{Non-network}$. We generate the claim label Y_{ijk} by taking a random draw from Bern (π_{ijk}) where $\pi_{ijk} = \exp({}_0\beta_f + {}_f\boldsymbol{x}_{ijk}^\top {}_f\boldsymbol{\beta}_f)(1+\exp({}_0\beta_f + {}_f\boldsymbol{x}_{ijk}^\top {}_f\boldsymbol{\beta}_f))^{-1}$.

	β_f	
Feature	$\mathcal{D}^{Network}$	$\mathcal{D}^{Non-network}$
Policyholder:		
`AgePH`	-2.00	-2.00
`NrContractsPH`	-1.50	-1.50
Claim-specific:		
`ClaimAmount`	0.20	0.20
`ClaimAge`	-0.35	-0.35
Social network:		
`n1.size`	2.00	0
`n2.size`	-2.00	0
`n2.ratioFraud`	3.00	0

Table 4.5: The average, minimum and maximum frequency and relative frequency (%) of the ground truth and expert-based claim labels across the 100 synthetic data sets.

		Frequency (%)		
		Average	Minimum	Maximum
$\mathcal{D}^{Network}$	Y_{ijk}:			
	- Fraudulent	2 175.37 (2.01%)	2 121.00 (1.97%)	2 219.00 (2.06%)
	- Non-fraudulent	105 937.66 (97.99%)	105 103.00 (97.94%)	106 655.00 (98.03%)
	$Y_{ijk}^{\texttt{expert}}$:			
	- Fraudulent	284.28 (0.26%)	260.00 (0.24%)	318.00 (0.30%)
	- Non-fraudulent	9 149.38 (8.46%)	8 952.00 (8.30%)	9 431.00 (8.67%)
	- Uninvestigated	98 679.37 (91.27%)	98 021.00 (91.08%)	99 428.00 (91.45%)
$\mathcal{D}^{Non-network}$	Y_{ijk}:			
	- Fraudulent	2 175.84 (2.01%)	2 139.00 (1.98%)	2 209.00 (2.04%)
	- Non-fraudulent	105 921.40 (97.99%)	104 978.00 (97.96%)	106 789.00 (98.02%)
	$Y_{ijk}^{\texttt{expert}}$:			
	- Fraudulent	293.64 (0.27%)	254.00 (0.24%)	335.00 (0.31%)
	- Non-fraudulent	9 137.37 (8.45%)	8 924.00 (8.29%)	9 454.00 (8.72%)
	- Uninvestigated	98 666.23 (91.28%)	97 826.00 (91.01%)	99 366.00 (91.45%)

and relative frequency of the Y_{ijk} categories (i.e., `fraudulent` and `non-fraudulent`) and $Y_{ijk}^{\texttt{expert}}$ (i.e., `fraudulent`, `non-fraudulent`, and `uninvestigated`) in each data set. The average, minimum, and maximum values for both the frequency and relative frequency are computed and presented in Table 4.5. In all synthetic data sets, the empirical class imbalance is nearly identical to the target class imbalance. The minimum class imbalance in $\mathcal{D}^{Network}$ is 1.97% and the maximum 2.06%. In $\mathcal{D}^{Non-network}$, the minimum is 1.98% and the maximum is 2.04%. The class

imbalance is also reflected in the expert judgement. Only a small fraction of claims are subject to investigation (approximately 9%), and among those investigated, only a minority are found to be fraudulent. Further, the empirical distributions of both Y_{ijk} and $Y_{ijk}^{\texttt{expert}}$ are similar across all simulated data sets.

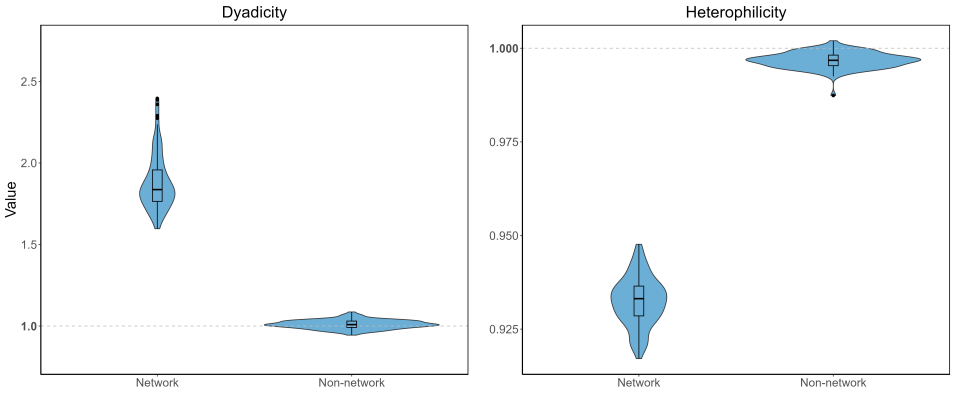
Empirical distribution of the features Figures 4.5 and 4.6 present the empirical distribution of the features included in the data-generating fraud models (see Table 4.4) of one synthetically generated data set. Figure 4.5 displays the features' empirical distribution in a simulated data set $\mathcal{D}^{Network}$. Figure 4.6 shows this in a synthetic data set $\mathcal{D}^{Non-network}$. The empirical distribution of policyholder and claim-specific features is different across fraudulent and non-fraudulent claims in both types of data sets. For example, fraudulent claims are mostly associated with younger policyholders (top left plot on Figures 4.5 and 4.6). Further, in $\mathcal{D}^{Network}$, the difference in the empirical distribution between fraudulent and non-fraudulent claims is also present for the social network features `n1.size`, `n2.size` and `n2.ratioFraud`. This suggests that the claim labels are linked to these features in $\mathcal{D}^{Network}$. In contrast, the empirical distributions of the social network features do not differ in $\mathcal{D}^{Non-network}$, indicating that there is no association between these features and the claim label.

Homophily Figure 4.4 illustrates the dyadicity $\mathscr{D}$ and heterophilicity $\mathscr{H}$ observed in the simulated data sets (see Section 4.2.2). In $\mathcal{D}^{Network}$, the fraudulent claims are more densely connected to each other compared to what we expect by chance ($\mathscr{D} > 1$). In addition, fraudulent claims have fewer connections to non-fraudulent claims relative to what we expect by chance ($\mathscr{H} < 1$). In comparison, we observe no patterns of homophily in $\mathcal{D}^{Non-network}$. Both the dyadicity ($\mathscr{D} \approx 1$) and heterophilicity ($\mathscr{H} \approx 1$) correspond to values indicative of a random network configuration.

Effect size of the features Next, we estimate the coefficient vector $\boldsymbol{\beta}_f$ in each synthetic data set. We fit the following logistic regression model

$$\begin{aligned}\text{logit}(E[Y_{ijk}]) = {}& {}_0\beta_f + {}_1\beta_f \texttt{AgePH}_i + {}_2\beta_f \texttt{NrContractsPH}_{ij} + {}_3\beta_f \texttt{ClaimAmount}_{ijk} \\ & + {}_4\beta_f \texttt{ClaimAge}_{ijk} + {}_5\beta_f \texttt{n1.size}_{ijk} + {}_6\beta_f \texttt{n2.size}_{ijk} \\ & + {}_7\beta_f \texttt{n2.ratioFraud}_{ijk}.\end{aligned} \tag{4.14}$$

Figure 4.4: The empirical distribution of the dyadicity and heterophilicity in the synthetically generated $\mathcal{D}^{Network}$ and $\mathcal{D}^{Non-network}$ data sets.



where i refers to the policyholder, j to the contract and k to the claim. This model is the same as the data-generating fraud model (see Table 4.4). Figure 4.7 depicts the empirical distribution of the estimated coefficient vector $\widehat{\boldsymbol{\beta}}$ across the 100 simulated data sets. Panel (a) shows the estimates obtained from $\mathcal{D}^{Network}$ and panel (b) from $\mathcal{D}^{Non-network}$. In both types of data sets, we observe some minor deviations from the specified effect size for most features, reflecting sampling variability. Further, the variability of the estimates is relatively small. Hence, we are able to accurately replicate the specified effect size for the different features, with only minor deviations due to sampling variability. The deviation from the specified effect size and variability, however, is substantially larger for the social network feature `n2.ratioFraud`. This feature represents the proportion of fraudulent claims in a claim's second order neighborhood. The estimated effect size of `n2.ratioFraud` is lower than its value as specified in $\boldsymbol{\beta}_f$. This is most likely attributable to the iterative growth in the number of fraudulent claims (see Algorithm 4), leading to a deviation in the estimated effect size from the originally specified value. In the first iterations, there are only a few instances of fraudulent claims. Hence, most of the unlabeled claims will have similar values for `n2.ratioFraud`. As the number of iterations increases, there will be a progressive increase in the proportion of fraudulent claims (see Appendix C.6). Accordingly, there will be more distinct feature values for `n2.ratioFraud`. Thus, the empirical distribution of `n2.ratioFraud` alters with each iteration.

Figure 4.5: Illustration of the features' empirical distribution in a synthetically generated $\mathcal{D}^{Network}$.

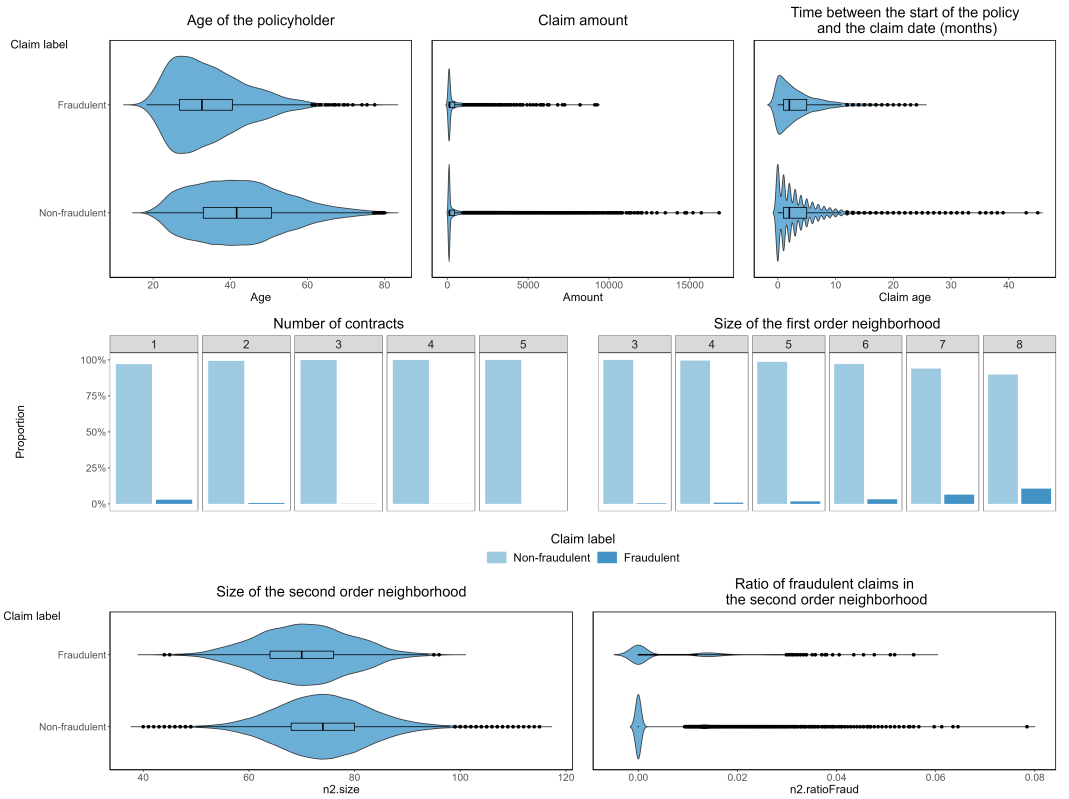


Figure 4.6: Illustration of the features' empirical distribution in a synthetically generated $\mathcal{D}^{Non-network}$.

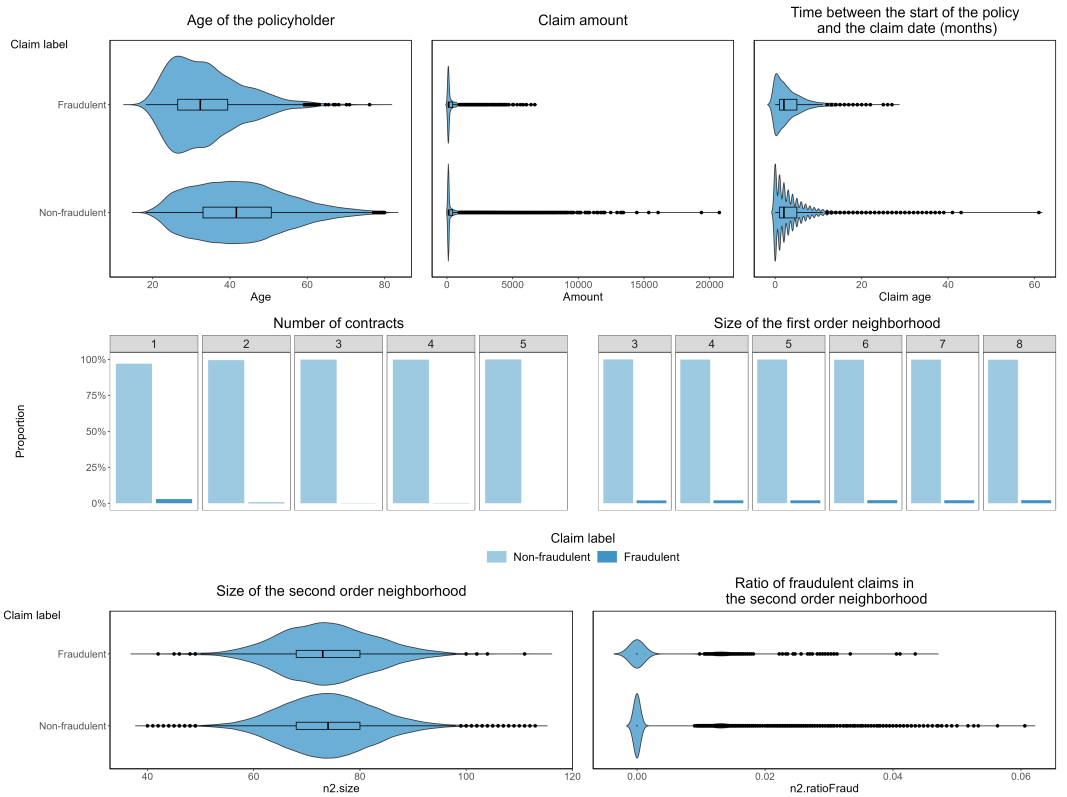
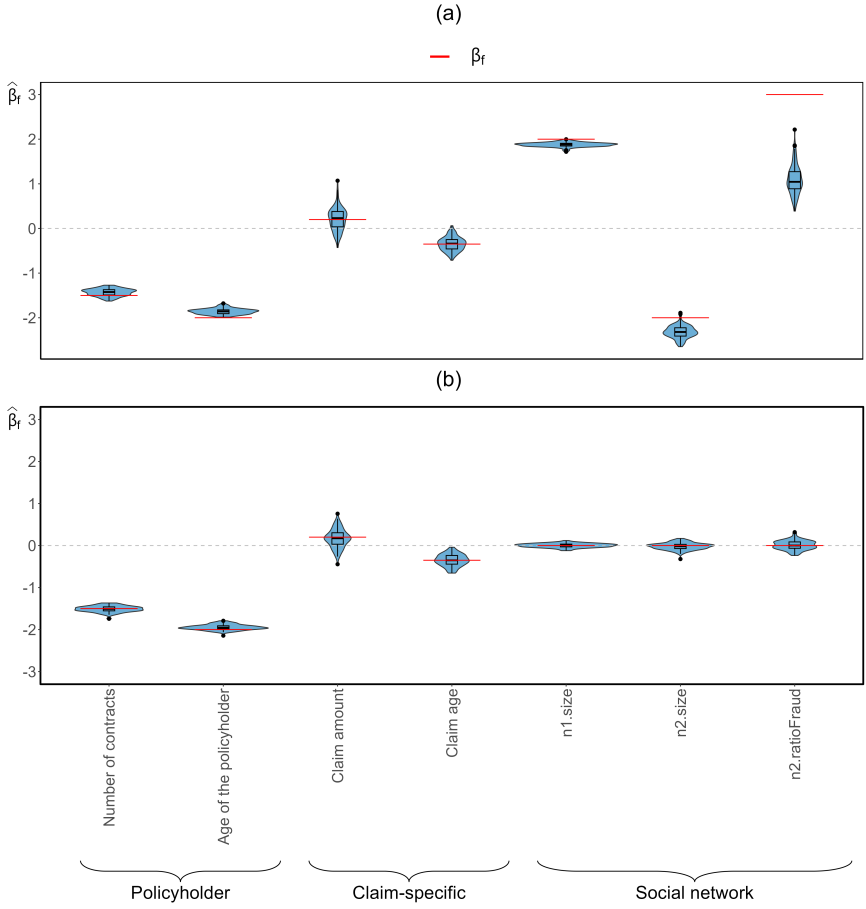


Figure 4.7: Empirical distribution of the coefficient estimates across the (a) 100 simulated data sets $\mathcal{D}^{Network}$ and (b) 100 simulated data sets $\mathcal{D}^{Non-network}$. The red lines on the plot depict the features' effect size as specified in $\boldsymbol{\beta}_f$.



4.4.2 Exploring the capabilities of the simulation engine: evaluating a fraud detection model's effectiveness

In this section, we illustrate the development and validation of a fraud detection model using a supervised learning technique (see Section 4.2). We proceed with the 100 synthetic data sets $\mathcal{D}^{Network}$. In each simulated data set we construct a fraud detection model by fitting a logistic regression model to the investigated claims (i.e.

those that are investigated and labeled by the expert, see Section 4.3.4). We rely on logistic regression given its robustness to imbalanced class sizes (Oommen et al., 2011; Marques et al., 2013; van den Goorbergh et al., 2022). Table 4.5 illustrates that said imbalance is present in each synthetic data set. To examine the added value of social network analytics when a network effect is present, we define two distinct model specifications. For model 1, we only include policyholder and claim-specific features

$$\begin{aligned}\text{logit}(E[Y_{ijk}]) = {}& {}_0\beta_f + {}_1\beta_f \texttt{AgePH}_i + {}_2\beta_f \texttt{NrContractsPH}_{ij} + {}_3\beta_f \texttt{ClaimAmount}_{ijk} \\ & + {}_4\beta_f \texttt{ClaimAge}_{ijk}.\end{aligned} \tag{4.15}$$

Next, we extend model 1 by incorporating social network features as well, resulting in model 2

$$\begin{aligned}\text{logit}(E[Y_{ijk}]) = {}& {}_0\beta_f + {}_1\beta_f \texttt{AgePH}_i + {}_2\beta_f \texttt{NrContractsPH}_{ij} + {}_3\beta_f \texttt{ClaimAmount}_{ijk} \\ & + {}_4\beta_f \texttt{ClaimAge}_{ijk} + {}_5\beta_f \texttt{n1.size}_{ijk} + {}_6\beta_f \texttt{n2.size}_{ijk} \\ & + {}_7\beta_f \texttt{n2.ratioFraud}_{ijk}.\end{aligned} \tag{4.16}$$

To assess the predictive performance of the fraud detection models, we rely on the area under the receiver operating characteristic curve (AUC) (Hanley and McNeil, 1982) and the top decile lift (TDL) (Lemmens and Croux, 2006). The AUC measures how well the model differentiates between fraudulent and non-fraudulent claims. An AUC of 0.5 corresponds to a random model and a perfect model has an AUC of 1. The TDL measures the extent to which a model surpasses a random model in detecting fraudulent claims. We calculate the TDL by dividing the proportion of fraudulent claims among the top 10% of claims with the highest predicted probability by the relative frequency of fraudulent claims in the data set. The TDL of a random model is equal 1. The higher the TDL, the better the model performance.

In each synthetic data set, we examine the in- and out-of-sample predictive performance of the fitted model. Hereto, we use the model fit to compute the probability of fraud for claims in the in- and out-of-sample data set

$$\pi_{ijk} = \frac{e^{{}_0\hat{\beta}_f + {}_f\boldsymbol{x}_{ijk}^\top \widehat{\boldsymbol{\beta}}_f}}{1 + e^{{}_0\hat{\beta}_f + {}_f\boldsymbol{x}_{ijk}^\top \widehat{\boldsymbol{\beta}}_f}}. \tag{4.17}$$

The in-sample data set consists of the investigated claims (approximately 9% of all claims are investigated, see Table 4.5). Here, we use the labels of $Y_{ijk}^{\texttt{expert}}$ as

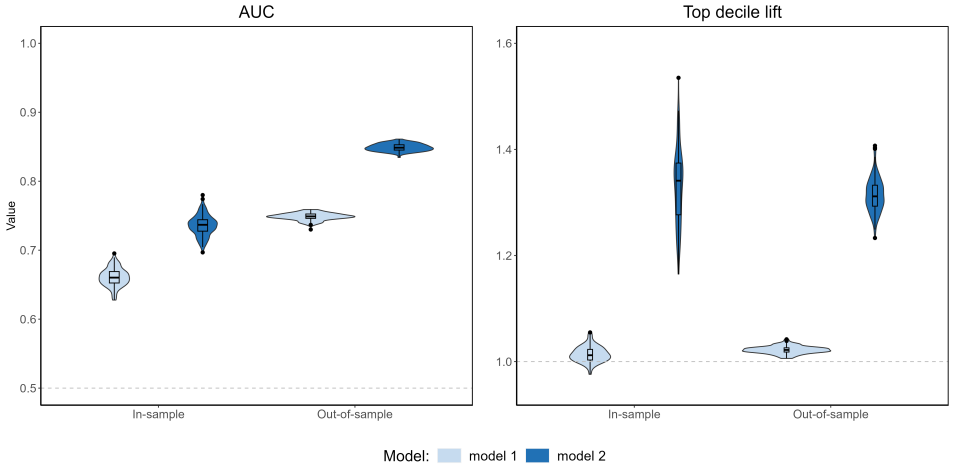
outcome when computing the performance measures. The out-of-sample data set contains all uninvestigated claims. In our synthetic data set, we have the advantage of having access to the ground truth label Y_{ijk} of the uninvestigated claims, which is not available in real-life data sets. We calculate the out-of-sample AUC and TDL using Y_{ijk}. As such, we evaluate to which extent our model is able to generalize and detect fraud in the unlabeled claims.

Figure 4.8 depicts the in- and out-of-sample predictive performance of model 1 and model 2. In terms of AUC and TDL, model 2 consistently outperforms model 1 in both the in- and out-of-sample evaluations. Consequently, by incorporating social network features in addition to the traditional claim characteristics, we enhance the model's ability to identify fraudulent claims. Furthermore, the TDL of model 1 approaches one in all simulated data sets, indicating that the model performs no better than random chance in identifying fraudulent claims within the top 10% of predicted probabilities. In comparison, the TDL of model 2 is substantially larger than one. Model 2 also retains its predictive performance on the out-of-sample data sets. Hence, by training the model on the investigated claims, we can effectively capture the distinct patterns exhibited by fraudulent claims. One seemingly contradictory finding, however, is that the out-of-sample AUCs are higher than the in-sample AUCs. This discrepancy in performance is likely due to the different labels used for model evaluation. For the in-sample comparison, we rely on the expert judgment labels $Y_{ijk}^{\texttt{expert}}$. Conversely, for the out-of-sample comparison we use the ground-truth labels Y_{ijk}. This variation in label sources may contribute to the observed differences in model performance. In addition, the in-sample data sets are 10 times smaller than the the out-of-sample data sets. Consequently, the in-sample data sets exhibit more variability. Further, a small number of the investigated claims will be false positives or false negatives (see Section 4.3.4). When we fit the models with the ground truth-label Y_{ijk} instead of $Y_{ijk}^{\texttt{expert}}$, the in-sample performance is higher compared to the out-of-sample performance (see Appendix C.7).

4.5 Discussion

In this chapter, we present a powerful and flexible toolbox to generate synthetic insurance fraud network data. The simulation engine consists of seven consecutive steps which enable us to generate a complete and complex data set. The engine generates policyholder characteristics, contract-specific features, the number of claims and individual claim costs, and the claim labels (fraudulent or non-fraudulent). To

Figure 4.8: Distribution of the performance measures across the 100 simulated data sets $\mathcal{D}^{Network}$. The grey line in the plots corresponds to the performance of a random model. The in-sample performance is evaluated using the labels of the investigated claims. The out-of-sample performance is assessed using the ground truth label Y_{ijk} of the uninvestigated claims.



ensure that the simulated data accurately reflects the real-world scenario, the fraud investigation process is also replicated. The generated data from each step is combined to produce a final synthetic database that can be used for various purposes. In generating the synthetic data, the user has control over various data-generating mechanisms.

The simulation engine can produce diverse scenarios to meet different research needs. We showcase this ability by generating two distinct types of data sets, one where the social network effect is present during the claim label generation and one where it is absent. Our results highlight the toolbox's capability to simulate synthetic data according to the user-defined parameters. Our simulation engine accurately generates the desired class imbalance as well as the specified effect sizes of the covariates (including the social network features). As such, we are able to generate data sets that closely mirror real-life insurance fraud data sets in motor insurance.

Researchers can utilize our simulation engine to conduct benchmark studies, aimed at addressing (methodological) challenges posed by insurance fraud. For instance, future research can focus on the evaluation of sampling techniques to handle the high class imbalance and the performance of learning methods in combination

with said sampling techniques.

Chapter 5

Conclusions and outlook

This thesis focuses on the use of hierarchical and network data in predictive models constructed for various insurance applications, ranging from detecting fraudulent insurance claims to insurance pricing. Recent technological innovations have not only facilitated the acquisition and storage of large data sets, but have also greatly enhanced the performance of various predictive modeling techniques. Notwithstanding, several outstanding challenges remain. The research conducted in this thesis provides a comprehensive exploration of two important challenges in insurance analytics and proposes data-driven solutions to tackle them.

5.1 Hierarchical MLFs

The first two chapters are dedicated to incorporating hierarchical MLFs into an insurance pricing model. In Chapter 2, we show that the random effects approach is an efficient strategy to handle hierarchical MLFs. We provide a comprehensive overview and comparison of existing estimation methods. Additionally, we present a data-driven procedure to construct an insurance pricing model when both hierarchically structured and contract-specific risk factors are available. Further, our results indicate that the Tweedie distribution is particularly well-suited for modeling and predicting damage rates.

The proposed approach, however, is confined to regression-type random effects models. Consequently, it shares the same drawbacks as GLMs. Variables have to be explicitly selected, as well as interaction terms and non-linear transformations. This is not the case for machine learning techniques, which partially explains why,

in general, machine learning methods have a higher predictive performance than the traditional GLMs. Currently, there exist a few machine learning techniques that incorporate a random effects component. Sela and Simonoff (2012) devised a framework and estimation method for tree-based methods with random effects. Similarly, Avanzi et al. (2023), developed a combination of neural networks with a random effects component. To the best of our knowledge, there is no existing research on the performance of random effects machine learning techniques when confronted with hierarchically structured risk factors. Hence, this is a possible first path for future research. Subsequent studies can provide insights into whether random effects machine learning methods share the same limitations as other random effects models (e.g. the existence of negative variance estimates). Furthermore, the random effects approach can be compared with alternative encoding methods, such as target encoding (Micci-Barreca, 2001) or entity embeddings from neural networks (Guo and Berkhahn, 2016).

In certain situations, the random effects approach is not feasible or appropriate. This is the main topic of Chapter 3, where we provide an algorithm to reduce a hierarchically structured risk factor to its essence. Using a combination of feature engineering, clustering techniques and cluster evaluation criteria, the algorithm groups similar categories at every level in the hierarchy. As such, the quality of the final clustering solution is dependent on the latter components. Future research can examine whether different features, clustering techniques and cluster evaluation criteria lead to better clustering solutions. In addition, our algorithm works top-down. Alternatively, researchers can design an algorithm that works bottom-up and that is able to group child categories of different parent categories.

5.2 Social network data

The third and final chapter tackles the scarcity of publicly available insurance fraud network data by introducing a simulation engine. Our engine is designed to generate a wide range of scenarios that closely resemble real-life data sets. By enabling users to specify the parameters of the data-generating mechanisms, we give them control over the characteristics of the resulting synthetic data set. Additionally, we demonstrate that the engine is able to generate synthetic data according to parameters as defined by the user.

The simulation engine is designed as a tool for various research endeavors. Consequently, a range of potential future research topics present itself, ranging from

methodological studies to hands-on case studies. Our aim is to stimulate research that examines several of the (methodological) challenges inherent to insurance fraud, such as the class imbalance and handling missing data. Alternatively, synthetically generated data sets can be used to examine and compare the accuracy in detecting fraudulent claims of several predictive modeling techniques. As such, the simulation engine can also serve as a tool to preselect a range of promising models that can be tested on real-life data. Further, the simulation engine provides a means to explore potential strategies for managing the dynamic and temporal nature of fraudulent activities. By adjusting parameters within the simulation and mimicking the fraudsters' attempt to remain undetected by altering their tactics, researchers can explore how different models respond to shifts in fraud dynamics. For example, by changing the composition of the included covariates and the covariates' effect size in the data generating fraud model. This can potentially lead to the identification of effective approaches for managing the evolving nature of fraudulent behavior.

Appendix Chapter 2

A.1 Jewell's hierarchical model: variance estimators

In our analysis, we make use of the estimators proposed by Ohlsson (2005)

$$\begin{aligned}
\sigma^2 &= \frac{1}{\sum_j \sum_k (T_{jk}-1)} \sum_{i,j,k,t} w_{ijkt}(Y_{ijkt} - \bar{Y}_{\cdot jk\cdot})^2, \\
\sigma_B^2 &= \frac{\sum_j \sum_k w_{\cdot jk\cdot}(\bar{Y}_{\cdot jk\cdot} - \bar{Y}_{\cdot j\cdot\cdot})^2 - \hat{\sigma}^2 \sum_j (K_j - 1)}{w_{\cdot\cdot\cdot\cdot} - \sum_j \frac{\sum_k w_{\cdot jk\cdot}^2}{w_{\cdot j\cdot\cdot}}}, \\
\sigma_I^2 &= \frac{\sum_j z_{j\cdot}(\bar{Y}^z_{\cdot j\cdot\cdot} - \bar{Y}^z_{\cdot\cdot\cdot\cdot})^2 - \hat{\sigma_B}^2 (J-1)}{z_{\cdot\cdot} - \frac{\sum_j z_{j\cdot}^2}{z_{\cdot\cdot}}},
\end{aligned} \tag{1}$$

where

$$\bar{Y}_{\cdot j\cdot\cdot} = \frac{\sum_k w_{\cdot jk\cdot}\bar{Y}_{\cdot jk\cdot}}{\sum_k w_{\cdot jk\cdot}} \quad \text{and} \quad \bar{Y}^z_{\cdot\cdot\cdot\cdot} = \frac{\sum_j z_{j\cdot}\bar{Y}^z_{\cdot j\cdot\cdot}}{\sum_j z_{j\cdot}}. \tag{2}$$

In the above equations, T_{jk} denotes the number of observations in group (j,k), K_j the number of branches in industry j and J the number of industries.

To estimate μ, we use

$$\hat{\mu} = \frac{\sum_j q_j \bar{Y}^z_{\cdot j\cdot\cdot}}{\sum_j q_j}. \tag{3}$$

A.2 Random effect estimates

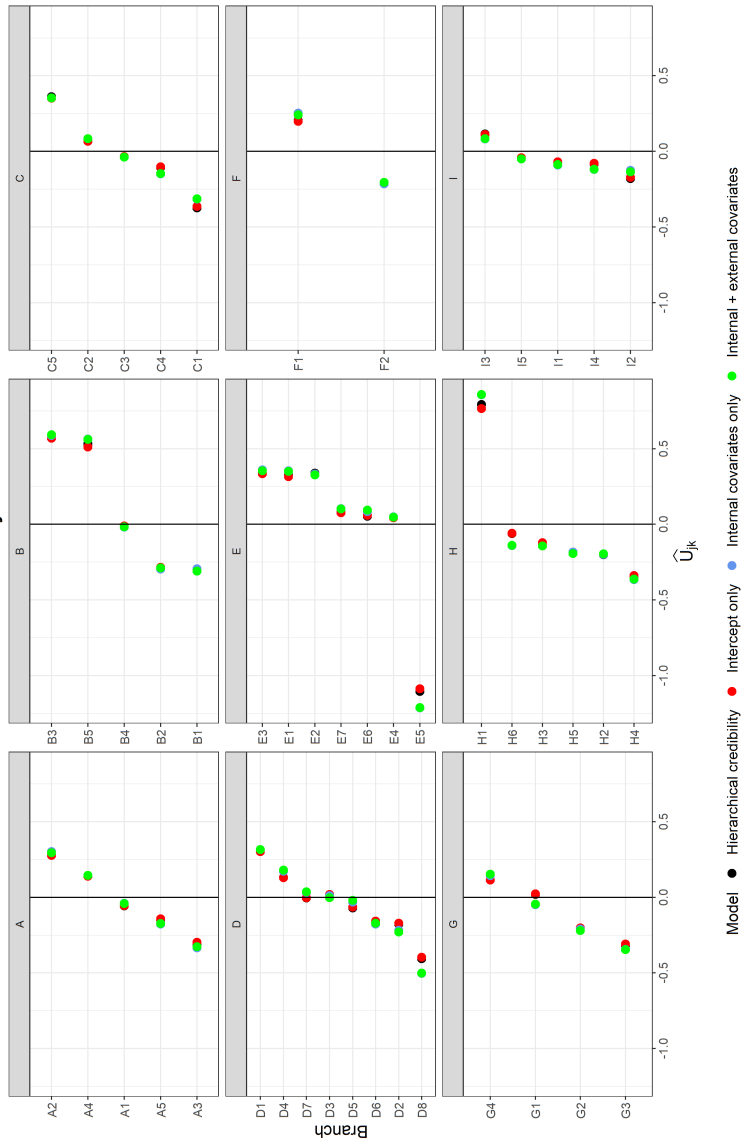


Figure A.1: LMM: Random effects estimates of the branches within industries.

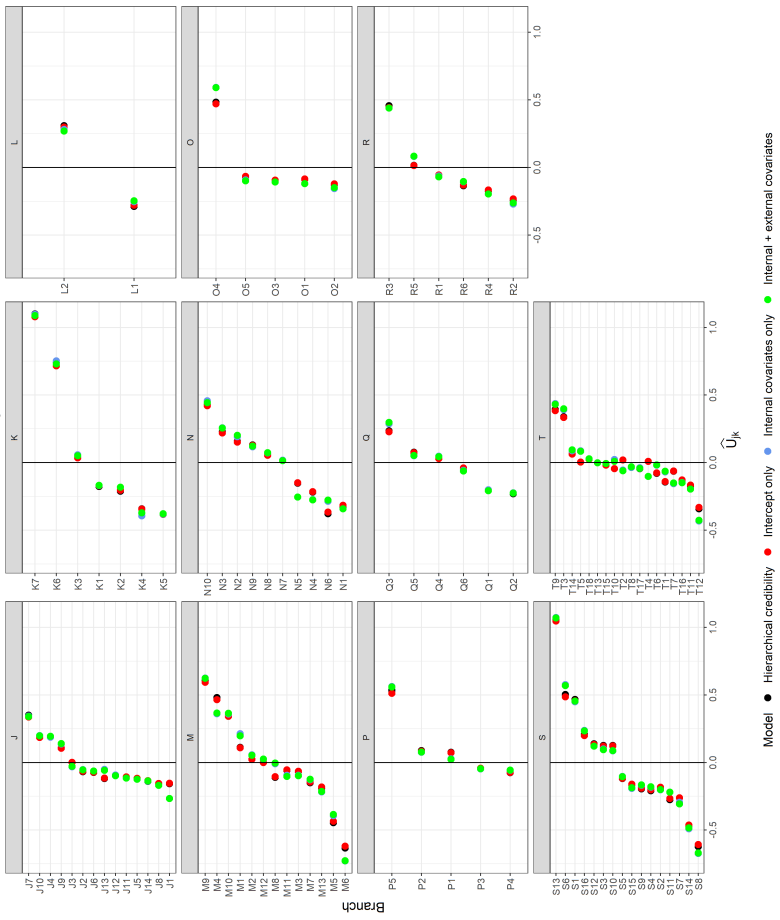


Figure A.2: LMM: Random effects estimates of the branches within industries (*continued*).

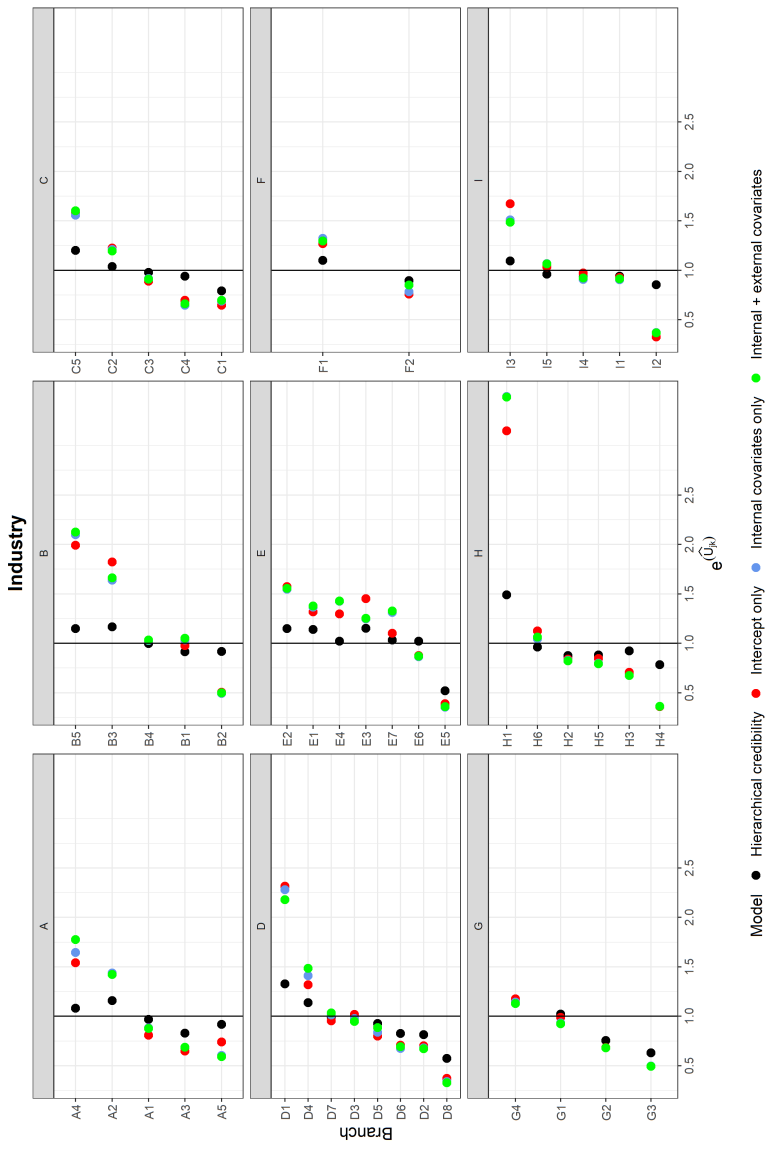


Figure A.3: Tweedie GLMM: Random effects estimates of the branches within industries.

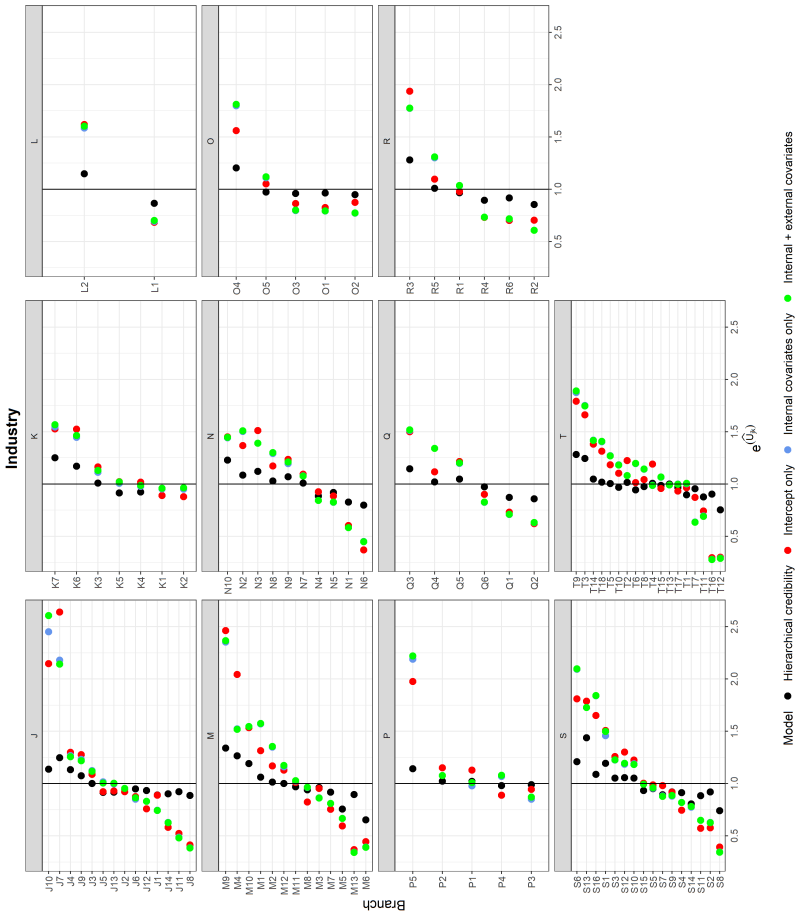


Figure A.4: Tweedie GLMM: Random effects estimates of the branches within industries (*continued*).

Appendix Chapter 3

B.1 Distance and (dis)similarity metrics

Using clustering algorithms, we aim to divide a set of data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_J$ into J' homogeneous groups such that observations in each cluster j' are more similar to each other compared to observations of other clusters $j^{'} \neq j'$. Consequently, most clustering algorithms rely on distance or (dis)similarity metrics between all pairwise observations.

The most commonly used distance metric between two vectors $\boldsymbol{x}_j$ and $\boldsymbol{x}_j$ is the squared Euclidean distance

$$d_e(\boldsymbol{x}_j, \boldsymbol{x}_j) = \|\boldsymbol{x}_j - \boldsymbol{x}_j\|_2^2. \tag{4}$$

Here, $\|\boldsymbol{x}_j\|_2 \coloneqq \sqrt{x_{j1}^2 + \cdots + x_{jn_f}^2}$ and n_f denotes the number of features considered. The squared Euclidean distance can be converted to the Gaussian similarity measure

$$s_g(\boldsymbol{x}_j, \boldsymbol{x}_j) = \exp\left(\frac{-\|\boldsymbol{x}_j - \boldsymbol{x}_j\|_2^2}{\sigma^2}\right) \tag{5}$$

which ranges from 0 (i.e. dissimilar) to 1 (i.e. identical). σ is a scaling parameter set by the user (Ng et al., 2001; Poon et al., 2012). When σ is small, the distance needs to be close to 0 to result in a high similarity measure. Conversely, for high σ, even large distances will result in a value close to 1.

Ideally, vectors that lie close to each other are characterized by a low distance $d(\cdot, \cdot)$ and a high similarity measure $s(\cdot, \cdot)$. Euclidean based distance/similarity measures, however, are not appropriate to capture the similarities between embeddings (Kogan

et al., 2005). Within NLP, the cosine similarity

$$s_c(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) = \frac{\boldsymbol{x}_j^\top \boldsymbol{x}_{j'}}{\|\boldsymbol{x}_j\|_2 \cdot \|\boldsymbol{x}_{j'}\|_2} \tag{6}$$

is therefore most often used to measure the similarity between embeddings (Mohammad and Hirst, 2012; Schubert, 2021). The cosine similarity ranges from -1 (opposite) to 1 (similar). In cluster analysis, however, we generally require the (dis)similarity measure to range from 0 to 1 (Everitt et al., 2011; Kogan et al., 2005; Hastie et al., 2009). In this case, we can use the angular similarity

$$s_a(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) = 1 - \frac{\cos^{-1}(s_c(\boldsymbol{x}_j, \boldsymbol{x}_{j'}))}{\pi} \tag{7}$$

which is restricted to [0, 1]. Hereto related is the angular distance

$$d_a(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) = \frac{\cos^{-1}(s_c(\boldsymbol{x}_j, \boldsymbol{x}_{j'}))}{\pi}. \tag{8}$$

The angular distance is a proper distance metric since it satisfies the triangle inequality $d(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) \leqslant d(\boldsymbol{x}_j, \boldsymbol{x}_z) + d(\boldsymbol{x}_z, \boldsymbol{x}_{j'})$ for any z (Schubert, 2021; Phillips, 2021). Conversely, the distance measure based on the cosine similarity does not satisfy this inequality.

B.2 Clustering algorithms

K-means clustering With k-means clustering (MacQueen et al., 1967), we group the J categories into J' clusters $(C_1, \ldots, C_{J'})$ by minimizing

$$\underset{(C_1,\ldots,C_{J'})}{\arg\min} \sum_{j'=1}^{J'} \sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'}) = \underset{(C_1,\ldots,C_{J'})}{\arg\min} \sum_{j'=1}^{J'} \sum_{\boldsymbol{x}_j \in C_{j'}} \|\boldsymbol{x}_j - \boldsymbol{c}_{j'}\|_2^2 \tag{9}$$

where $\boldsymbol{c}_{j'}$ denotes the cluster centre or centroid of cluster $C_{j'}$. $\boldsymbol{c}_{j'}$ is the sample mean of all $\boldsymbol{x}_j \in C_{j'}$

$$\boldsymbol{c}_{j'} = \frac{1}{n_{j'}} \sum_{\boldsymbol{x}_j \in C_{j'}} \boldsymbol{x}_j. \tag{10}$$

where $n_{j'}$ denotes the number of observations in cluster $C_{j'}$. Hence, with (9) we minimize the within-cluster sum of squares.

K-means is only suited for numeric features, is sensitive to outliers, has several

local optima and the results are sensitive to the initialization (Kogan et al., 2005; Everitt et al., 2011; Ostrovsky et al., 2012; Hastie et al., 2009).

K-medoids clustering Contrary to k-means, k-medoids clustering (Kaufman and Rousseeuw, 1990*b*) uses an existing data point $\boldsymbol{x}_j$ as cluster centre. In addition, the distance measure in k-medoids clustering is not restricted to the Euclidean distance (Rentzmann and Wuthrich, 2019; Hastie et al., 2009). It can be used with any distance or dissimilarity measure. With k-medoids clustering we minimize

$$\underset{(C_1,\ldots,C_{J'})}{\arg\min} \sum_{j'=1}^{J'} \sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'}) \tag{11}$$

where $\boldsymbol{c}_{j'}$ is the observation for which $\sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'})$ is minimal (Struyf et al., 1997). This observation is most central within cluster $C_{j'}$ and is called a medoid.

K-medoids is applicable to any feature type and is less sensitive to outliers. Nonetheless, it still suffers from local optima and is sensitive to the initialization (Onan, 2017; Yu et al., 2018).

Spectral clustering In spectral clustering, we represent the data using an undirected similarity graph $G = \langle V, E \rangle$, where $V = (v_1, \ldots, v_j, \ldots, v_J)$ stands for the set of vertices and E denotes the set of edges (Hastie et al., 2009; von Luxburg, 2007; Wierzchoń and Kłopotek, 2019). The weight of the edges are represented using a $J \times J$ similarity matrix $\mathcal{S}$ which contains all pairwise similarities $s(\cdot, \cdot) \geqslant 0$ between the observations. The diagonal entries in the $\mathcal{S}$ matrix are equal to zero. Vertices v_j and $v_{j'}$ are connected if $s(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) > 0$. Hereby, we reformulate clustering as a graph-partitioning problem. We want to partition the graph such that edges within a group j' have high weights and edges between different groups $j' \neq j^{'}$ have low weights.

To represent the degree of the vertices, we set up a diagonal matrix $\mathcal{D}$ with diagonal elements $(j, j) = \sum_{j=1}^{J} s(\boldsymbol{x}_j, \boldsymbol{x}_{j'})$. We use $\mathcal{D}$ to transform the similarity matrix to the Laplacian matrix $\mathcal{L} = \mathcal{D} - \mathcal{S}$. Next, we compute the J eigenvectors $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_J)$ of $\mathcal{L}$. To cluster the observations in J' groups, we use the J' eigenvectors $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{J'})$ corresponding to the smallest eigenvalues and stack these in columns to form the matrix $U \in \mathbb{R}^{J \times J'}$. In U, the j^{th} row corresponds to the original

observation $\boldsymbol{x}_j$. In the ideal case of J' true clusters, $\mathcal{L}$ is a block diagonal matrix

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_1 & & & \\ & \mathcal{L}_2 & & \\ & & \ddots & \\ & & & \mathcal{L}_{J'} \end{bmatrix} \tag{12}$$

when the vertices are ordered according to their cluster membership. Here, $\mathcal{L}_{j'}$ is the block corresponding to cluster j'. In this situation, $\mathcal{L}$ has J' eigenvectors with eigenvalue zero and these eigenvectors are indicator vectors (i.e. the values are 1 for a specific cluster j' and 0 for clusters $j' \neq j'$) (Hastie et al., 2009; Poon et al., 2012; von Luxburg, 2007). This allows us to easily identify the J' groups. Consequently, in a second step, we apply k-means to U and the results hereof determine the clustering solution.

We commonly refer to $\mathcal{L}$ as the unnormalized Laplacian. It is, however, preferred to use a normalized version of $\mathcal{L}$ (von Luxburg et al., 2004; von Luxburg et al., 2008). One way to normalize $\mathcal{L}$ is by applying the following transformation to $\mathcal{L}$

$$\mathcal{D}^{-1/2}\mathcal{L}\mathcal{D}^{-1/2} = I - \mathcal{D}^{-1/2}\mathcal{S}\mathcal{D}^{-1/2}. \tag{13}$$

Spectral clustering can be used with any feature type and is less sensitive to initialization issues, outliers and local optima (Verma and Meila, 2003; von Luxburg, 2007). Moreover, spectral clustering is specifically designed to identify non-convex clusters (for every pair of points inside a convex cluster, the connecting straight line segment is within this cluster). Conversely, k-means, k-medoids and HCA generally do not work well with non-convex clusters (Hastie et al., 2009; von Luxburg, 2007). When compared to other clustering algorithms, spectral clustering often has a better overall performance (Murugesan et al., 2021; Rodriguez et al., 2019). Notwithstanding, spectral clustering is sensitive to the employed similarity metric (Haberman and Renshaw, 1996; von Luxburg, 2007; de Souto et al., 2008).

Hierarchical clustering analysis Contrary to the other clustering methods, hierarchical clustering analysis (HCA) does not start from a specification of the number of clusters. Instead, it builds a hierarchy of clusters which can be either top-down or bottom-up (Hastie et al., 2009). In the agglomerative or bottom-up approach, each observation is initially assigned to its own cluster and we recursively

merge clusters into a single cluster. When merging two clusters, we select the pair for which the dissimilarity is smallest. Conversely, in the divisive or top-down approach, we start with all observations assigned to one cluster and at each step, the algorithm recursively splits one of the existing clusters into two new clusters. Here, we split the cluster that results in the largest between-cluster dissimilarity. Consequently, in both approaches we need to define how to measure the dissimilarity between two clusters. With complete-linkage, for example, the distance between two clusters $C_{j'}$ and $C_{j'}$ is defined as the maximum distance $d(\cdot, \cdot)$ between two observations in the separate clusters

$$d_{CL}(C_{j'}, C_{j'}) = \max_{\substack{x_j \in C_{j'}, \\ x_j \in C_{j'}}} (d(x_j, x_j)). \tag{14}$$

Conversely, with single-linkage we define the distance between two clusters as

$$d_{SL}(C_{j'}, C_{j'}) = \min_{\substack{x_j \in C_{j'}, \\ x_j \in C_{j'}}} (d(x_j, x_j)). \tag{15}$$

Several other methods are available and we refer the reader to Hastie et al. (2009) for an overview. The visualization of the different steps in HCA is often referred to as a dendrogram. It plots a tree-like structure which shows how the clusters are formed at each step in the algorithm. To partition the data into J' clusters, we cut the dendrogram horizontally at the height that results in J' clusters.

Due to its design, HCA is less sensitive to initialization issues and local optima in comparison to k-means. In addition, we can employ HCA with any type of feature and HCA with single-linkage is more robust to outliers (Everitt et al., 2011; Timm, 2002). The disadvantage of HCA is that divisions or fusions of clusters are irrevocable (Kaufman and Rousseeuw, 1990*a*; Kogan et al., 2005). Once a cluster has been split or merged, it cannot be undone.

B.3 Internal cluster evaluation criteria

In the aforementioned clustering techniques, J' can be considered a tuning parameter that needs to be carefully chosen from a range of different (integer) values. Hereto, we require a cluster validation index to select that J' which results in the most optimal clustering solution. We divide the cluster validation indices into two groups, internal and external (Liu et al., 2013; Everitt et al., 2011; Wierzchoń and Kłopotek, 2019; Halkidi et al., 2001). Using external validation indices, we evaluate the clustering criterion with respect to the true partitioning (i.e. the actual assignment of the

observations to different groups is known). Conversely, we rely on internal validation indices when we do not have the true cluster label at our disposal. Here, we evaluate the compactness and separation of a clustering solution. The compactness indicates how dense the clusters are and compact clusters are characterized by observations that are similar and close to each other. Clusters are well separated when observations of different clusters are dissimilar and far from each other. Consequently, we employ internal validation indices to choose that J' which results in compact clusters that are well separated (Liu et al., 2013; Everitt et al., 2011; Wierzchoń and Kłopotek, 2019).

Several internal validation indices exist and each index formalizes the compactness and separation of the clustering solution differently. An extensive overview of internal (and external) validation indices is given in Liu et al. (2013) and Wierzchoń and Kłopotek (2019). Vendramin et al. (2010) conducted an extensive comparison of the performance 40 internal validation criteria using 1080 data sets. These data sets were grouped into four categories: a) a low number of features (i.e. $\in (2, 3, 4)$); b) a high number of features (i.e. $\in (22, 23, 24)$); c) a low number of true clusters (i.e. $\in (2, 4, 6)$) and d) a high number of true clusters (i.e. $\in (12, 14, 16)$). The authors concluded that the silhouette and Caliński-Harabasz indices are superior compared to other validation criteria. These indices are well-known within cluster analysis (Wierzchoń and Kłopotek, 2019; Govender and Sivakumar, 2020; Vendramin et al., 2010). Nonetheless, the results of Vendramin et al. (2010) do no necessarily generalize to our data set. We therefore include two additional, commonly used criteria: the Dunn-index and Davies-Bouldin index.

Caliński-Harabasz index The Caliński-Harabasz (CH) index (Caliński and Harabasz, 1974; Liu et al., 2013) is defined as the ratio of the average between- to the within-sum of squares

$$\frac{\sum_{j'=1}^{J'} n_{j'} \|\boldsymbol{c}_{j'} - \boldsymbol{c}\|_2^2/(J'-1)}{\sum_{j'=1}^{J'} \sum_{\boldsymbol{x}_j \in C_{j'}} \|\boldsymbol{x}_j - \boldsymbol{c}_{j'}\|_2^2/(J-J')} \tag{16}$$

where $\boldsymbol{c}$ denotes the global centre of all observations. We compute $\boldsymbol{c}$ as

$$\boldsymbol{c} = \frac{1}{J} \sum_{j=1}^{J} \boldsymbol{x}_j. \tag{17}$$

The higher this index, the more compact and well-separated the clustering solution. This index is also known as the Pseudo F-statistic. The results of Vendramin et al. (2010) suggest that the CH index performs better when the number of features is high and the number of true clusters is low.

Davies-Bouldin index The Davis-Bouldin index is defined as (Davies and Bouldin, 1979)

$$\frac{1}{J'}\sum_{j'=1}^{J'}\max_{j',\,j'\neq j'}\left(\frac{\frac{1}{n_{j'}}\sum_{\boldsymbol{x}_j\in C_{j'}}d(\boldsymbol{x}_j,\boldsymbol{c}_{j'})+\frac{1}{n_{j'}}\sum_{\boldsymbol{x}_j\in C_{j'}}d(\boldsymbol{x}_j,\boldsymbol{c}_{j'})}{d(\boldsymbol{c}_{j'},\boldsymbol{c}_{j'})}\right). \tag{18}$$

The numerator in (18) captures the compactness of clusters $C_{j'}$ and $C_{j'}$ and dense clusters are characterized by low values. With the denominator, we measure the distance between the centroids of clusters $C_{j'}$ and $C_{j'}$ and this signifies how well separated the clusters are. Hence, low ratios are indicative of dense clusters that are well separated. By taking the maximum ratio for a specific cluster $C_{j'}$ we take the worst scenario possible.

According to Vendramin et al. (2010), the Davis-Bouldin index performs better for data sets with fewer features and this finding was more pronounced for data sets with a low number of true clusters.

Dunn-index The Dunn-index (Dunn, 1974) is defined as the ratio of the minimum distance between the clusters to the maximum distance within clusters

$$\min_{1\leqslant j'\leqslant J'}\left(\min_{\substack{1\leqslant j'\leqslant J'\\ j\neq j}}\left(\frac{\min\limits_{\substack{x_j\in C_{j'}\\ x_j\in C_{j'}}}d(x_j,x_j)}{\max\limits_{1\leqslant\kappa\leqslant J'}\left\{\max\limits_{x_j,x_j\in C_\kappa}d(x_j,x_j)\right\}}\right)\right) \tag{19}$$

The higher this index, the better the clustering solution. Several variants exist of the Dunn-index (see, for example, Vendramin et al. (2010)) and we focus on the original formulation as given in (19). In Vendramin et al. (2010), the Dunn-index performed reasonably when focusing on the difference between the true and selected number of clusters. Notwithstanding, of all four indices considered in this chapter, it has the lowest performance.

Silhouette index For a specific observation $\boldsymbol{x}_j \in C_{j'}$, we define the average dissimilarity of $\boldsymbol{x}_j$ to all other observations in cluster $C_{j'}$ as

$$a(\boldsymbol{x}_j) = \frac{1}{n_{j'} - 1} \sum_{\boldsymbol{x}_j \in C_{j'}, j \neq j} d(x_j, x_j) \tag{20}$$

and the average dissimilarity of $\boldsymbol{x}_j$ to all observations in cluster $C_{j'}$ as

$$e(\boldsymbol{x}_j) = \frac{1}{n_{j'}} \sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{x}_j). \tag{21}$$

We compute $e(\boldsymbol{x}_j)$ for all clusters $C_{j'} \neq C_{j'}$ and calculate

$$b(\boldsymbol{x}_j) = \min_{C_{j'} \neq C_{j'}} e(\boldsymbol{x}_j). \tag{22}$$

We call $b(\boldsymbol{x}_j)$ the neighbour of $\boldsymbol{x}_j$ as it is the closest observation of another cluster. We calculate the silhouette value $s(\boldsymbol{x}_j)$ as

$$s(\boldsymbol{x}_j) = \frac{b(\boldsymbol{x}_j) - a(\boldsymbol{x}_j)}{\max(a(\boldsymbol{x}_j), b(\boldsymbol{x}_j))}. \tag{23}$$

$s(\boldsymbol{x}_j)$ indicates how well an observation $\boldsymbol{x}_j$ is clustered and from the definition, it follows that $-1 \leqslant s(\boldsymbol{x}_j) \leqslant 1$. For clusters with a single observation, we set $s(\cdot) = 0$. Values close to one indicate that the observation has been assigned to the appropriate cluster, since the smallest between dissimilarity $b(\boldsymbol{x}_j)$ is much larger than the within dissimilarity $a(\boldsymbol{x}_j)$ (Rousseeuw, 1987). Conversely, when $s(\boldsymbol{x}_j)$ is close to -1, $\boldsymbol{x}_j$ lies on average closer to the neighbouring cluster than to its own cluster and this suggests that this observation is not assigned to the appropriate cluster. We calculate the average silhouette width

$$\tilde{s} = \frac{1}{J} \sum_{j=1} s(\boldsymbol{x}_j). \tag{24}$$

to evaluate how good the clustering solution is. Higher $\tilde{s}$'s are associated with a better clustering solution.

In the study of Vendramin et al. (2010), the silhouette index had the most robust performance with regard to the different evaluation scenarios. The other evaluation criteria were more sensitive to the dimensionality of the data and the true number of clusters.

B.4 Empirical distribution of the category-specific weighted average damage rates and expected claim frequencies

Figure B.5: Empirical distribution of the category-specific weighted average damage rate at different levels in the hierarchy. One large value is removed to obtain a better visualization.

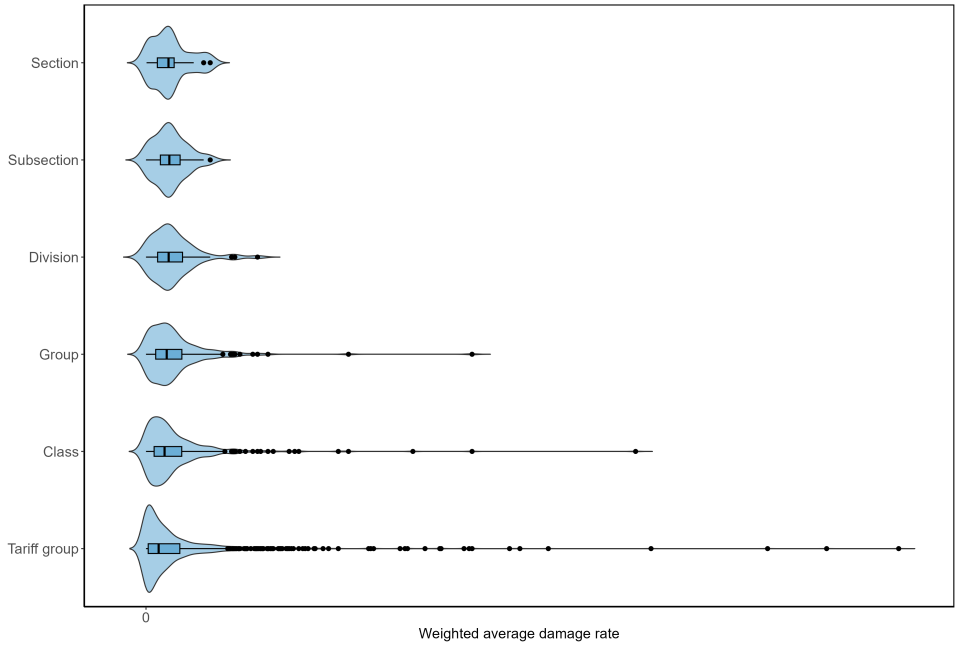
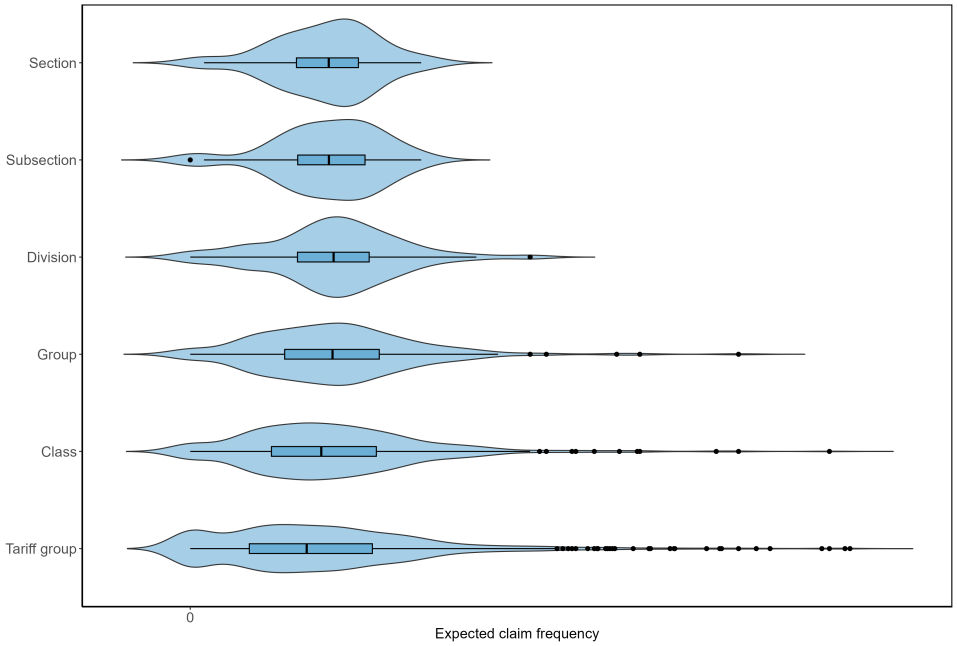


Figure B.6: Empirical distribution of the category-specific expected claim frequency at different levels in the hierarchy. Two large values are removed to obtain a better visualization.



B.5 Low-dimensional representation of the embedding vectors

Figure B.7: Low-dimensional visualization of all embedding vectors, resulting from the pre-trained USE v4 encoder, constructed for different categories at the `subsection` level. The text boxes display the textual labels. The blue dots connected to the boxes depict the position in the low-dimensional representation of the embeddings.

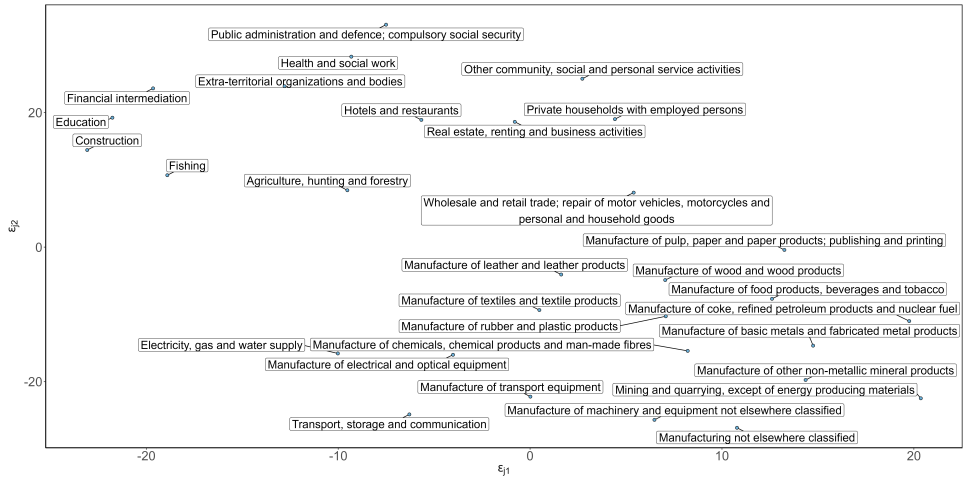
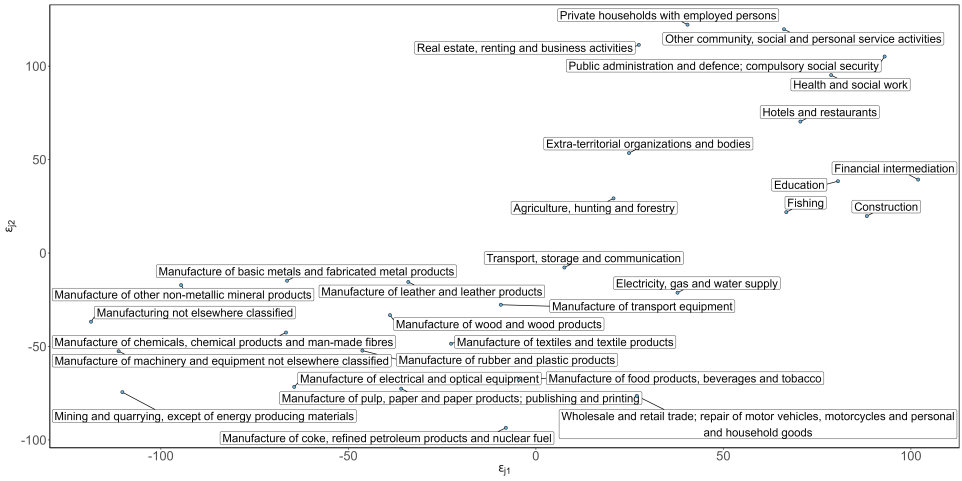


Figure B.8: Low-dimensional visualization of all embedding vectors, resulting from the pre-trained USE v5 encoder, constructed for different categories at the `subsection` level. The text boxes display the textual labels. The blue dots connected to the boxes depict the position in the low-dimensional representation of the embeddings.



B.6 Predictive performance when using the angular distance matrix $\mathcal{D}$ for the cluster evaluation criteria

Table B.1: Predictive performance on the training and test set, when the internal evaluation criterion is calculated using the angular distance and the complete feature vector.

			Development	Validation	
	J'	$\sum_{j'=1}^{J'} K'_{j'}$	Gini-index	Gini-index	Loss ratio
Benchmark	18	641	0.658	0.585	1.006
HCA:					
Silhouette index	7	137	0.669	0.625	1.008
Dunn index	14	322	0.662	0.604	1.008
Davies-Bouldin index	14	316	0.670	0.613	1.007
CH index	13	157	0.658	0.604	1.011
k-medoids:					
Silhouette index	6	135	0.655	0.593	1.008
Dunn index	14	315	0.656	0.576	1.006
Davies-Bouldin index	14	303	0.674	0.621	1.011
CH index	10	125	0.658	0.627	1.010
Spectral clustering:					
Silhouette index	8	150	0.670	0.599	1.009
Dunn index	14	299	0.663	0.585	1.008
Davies-Bouldin index	16	323	0.668	0.583	1.006
CH index	16	224	0.671	0.619	1.008

Appendix Chapter 4

C.1 Default configuration of the simulation engine

The default settings for the database, policyholder, contract-specific and claim characteristics are given in Table C.2. The default data-generating claim frequency and claim severity models are given in Appendix C.4. In Appendix C.5, we specify the default data-generating fraud model.

We define the level of class imbalance as the ratio of the number of fraudulent claims to the total number of claims. Further, `ExcludeParties` is a parameter that allows to exclude certain types of parties from the network. We include all types of parties by default.

Table C.2: Default configuration of the variables in the simulation engine.

	Variable	Description	Default value/generator
Data set	`TargetPrev`	Target level of class imbalance	`0.01`
	`NrPH`	Number of policyholder	`10000`
	`NrExperts`	Number of unique experts	$\lfloor 0.01$`NrPH`$\rfloor$
	`NrBrokers`	Number of unique brokers	$\lfloor 0.01$`NrPH`$\rfloor$
	`NrGarages`	Number of unique garages	$\lfloor 0.03$`NrPH`$\rfloor$
	`NrPersons`	Number of unique persons	1.5`NrPH`
	`ExcludeParties`	Type of party to exclude	`Expert`
Policyholder	`AgePH`	Age of the policyholder in years, default range is [18, 80]	$\mathcal{N}(40, 15)$
	`GenderPH`	Gender of the policyholder, default settings are `female` if $u_i \leqslant 0.28$, `male` if $u_i > 0.29$ and `non-binary` otherwise	$u_i \sim U(0, 1)$
	`ExpPH`	Exposure of the policyholder in years, default range is [0, 20]	$\mathcal{N}(5, 1.5)$
	`RateNrContracts`	Rate parameter λ_i for generating `NrContractsPH`	$\lambda_i = 0.25(1.05 - 2.5 \times 10^{-6} \times$ `AgePH`$_i$ $+$ $0.0025 \times$ `AgePH`$_i^2 - 2.65 \times 10^{-5} \times$ `AgePH`$_i^3$)
	`NrContractsPH`	Number of contracts. Default range is [1, 5]	$\text{Poi}(\lambda_i)$
Contract-specific	`ExpPHContracts`	Exposure corresponding to the contract	`ExpPH`$_i -$ $\text{U}(0,$ `ExpPH`$_i/2)$ if `NrContractsPH`$_i > 1$, else `ExpPHContracts`$_{ij}$ = `ExpPH`$_i$
	`AgeCar`	Age of the vehicle in years	$\max(\mathcal{N}(7.5, \sqrt{5}),$ `ExpPHContracts`$_{ij})$
	`OrigValueCar`	Original value of the vehicle	$\text{Exp}(\lambda_i/\text{NrContractsPH}_i)$
	`ValueCar`	Current value of the car	`OrigValueCar`$_{ij}(1 - \delta^{\text{a}})^{\text{AgeCar}_{ij}}$
	`Coverage`	Type of coverage provided by the insurance company (see Appendix C.3)	$\text{Multinomial}(1, \pi_{\text{TPL}}, \pi_{\text{PO}}, \pi_{\text{FO}})$
	`Fuel`	Type of fuel of the vehicle	Bernoulli(0.3)
	`BonusMalus`	Level occupied in bonus-malus scale of the insurance company	$\min(\lfloor \mathcal{G}(1, 1/3) \rfloor, 22)$
Claim	`ClaimAge`	Number of months from the contract's inception to the date of the incident	$\lfloor \text{Exp}(0.25) \rfloor$
	`ClaimDate`	Number of years between the start of the contract and the claim's filing date	$\max(U(0,$ `ExpPHContracts`$_{ij})$, `ClaimAge`$_{ijk}/12)$
	`Police`	Whether police was called when the incident happened	Bernoulli(0.25)
	`nPersons`	Number of people involved in the claim, range is [0, 5]	$S \xleftarrow{\pi_p} x$, $S = (0, 1, 2, 3, 4, 5)$ and $\pi_p = (0.025, 0.6, 0.2, 0.1, 0.1, 0.025)$

[a] If `OrigValueCar` < 30000, $\delta = 0.15$ and $\delta = 0.075$ otherwise. Hence, more expensive cars have a lower depreciation rate.

C.2 Limiting the range of feature values

The default range for `AgePH` is $[18, 80]$. Values generated outside of this range are redistributed among the integer values falling inside the range, proportional to the frequency of the values within this range. Hereafter, we take a random draw from $U(0, 1)$ and add this value to the integer to obtain a numeric value.

We specify $[0, 20]$ as the default range for `ExpPH`. Furthermore, `AgePH - ExpPH` cannot be smaller than the user-specified minimum for `AgePH`. This would imply that the contract started before the policyholder is legal age of driving. Hereto, we define $\texttt{MaxExp}_i$ = $\texttt{AgePH}_i$ - `MinAge`. For values outside the prespecified range, we redraw a value from $U(\texttt{MinExp}, \texttt{MaxExp}_i)$.

We define $[1, 5]$ as default range for `NrContractsPH`. Values outside this range are rounded to the closest boundary.

C.3 Simulating type of coverage

The type of coverage is a nominal variable with three levels. We rely on a multinomial regression model to generate the type of coverage as a function of `ValueCar`, `AgeCar` and `AgePH`. The general form of the multinomial logistic regression model (Agresti, 2013) is

$$\log\left(\frac{\pi_j(\boldsymbol{x}_i)}{\pi_J(\boldsymbol{x}_i)}\right) = \boldsymbol{x}_i^\top \boldsymbol{\beta}_j, \quad j = (1, \ldots, J-1),$$

where $\pi_j(\boldsymbol{x}_i) = P(\texttt{Coverage}_i = j|\boldsymbol{x}_i)$ denotes the probability that $\texttt{Coverage}_i$ equals category j. $\texttt{Coverage}_i$ denotes the response variable's value for observation i and here, we use $j = (1, \ldots, J)$ as an index for the categories of the response variable. We use category J as reference category. $\boldsymbol{x}_i$ denotes the covariate vector and $\boldsymbol{\beta}_j$ is the parameter vector for category j. For notational simplicity, we assume that $\boldsymbol{x}_i$ is fixed for all categories $j = (1, \ldots, J-1)$. Further, $\sum_{j=1}^{J} \pi_j(\boldsymbol{x}_i) = 1 \ \forall \ i \in (1, \ldots, N)$ where N denotes the total number of observations.

The category-specific probability for category $j = (1, \ldots, J-1)$ is calculated as

$$\pi_j(\boldsymbol{x}_i) = \frac{e^{\boldsymbol{x}_i^\top \boldsymbol{\beta}_j}}{1 + \sum_{h=1}^{J-1} e^{\boldsymbol{x}_i^\top \boldsymbol{\beta}_h}}$$

and we calculate this probability for reference category J as

$$\pi_J(\boldsymbol{x}_i) = \frac{1}{1 + \sum_{h=1}^{J-1} e^{\boldsymbol{x}_i^\top \boldsymbol{\beta}_h}}.$$

In our simulation engine, $\texttt{Coverage}_i \in (\texttt{TPL}, \texttt{PO}, \texttt{FO})$ and we define

$$\begin{aligned} \boldsymbol{\beta}_{\texttt{TPL}} &= (\log(0.50), \log(1.25), \log(0.25)), \\ \boldsymbol{\beta}_{\texttt{PO}} &= (\log(1.25), \log(0.75), \log(1.05)), \\ \boldsymbol{\beta}_{\texttt{FO}} &= (\log(1.50), \log(0.75), \log(1.25)). \end{aligned} \tag{25}$$

The covariate vector $\boldsymbol{x}_i$ consists of the normalized values for the value of the car, age of the car and age of the policyholder. Given a variable $a = (a_1, \ldots, a_i, \ldots, a_N)$, we normalize in $[-1, 1]$ using

$$2\frac{(a - \min(a))}{\max(a) - \min(a)} - 1. \tag{26}$$

Hereafter, we calculate the probabilities $(\pi_{\text{TPL}}(\boldsymbol{x}_i), \pi_{\text{PO}}(\boldsymbol{x}_i), \pi_{\text{FO}}(\boldsymbol{x}_i))$ for all observations. We generate values for the type of coverage by taking random draws from Multinomial$(1, \pi_{\text{TPL}}, \pi_{\text{PO}}, \pi_{\text{FO}})$.

Using the default values (see equation (25)), the probability of signing up for a full omnium is larger for expensive, relatively new cars and older policyholders. Similarly, the probability of taking out a partial omnium is higher for expensive, relatively new cars but here the effect of age is less strong. Young policyholders with an inexpensive, older car have a higher probability to take out a policy with only third party liability.

C.4 Claim frequency and claim severity model

Both the claim frequency and claim severity model are based on the results in Henckaerts et al. (2018). In this paper, the authors fit a claim frequency and claim severity model on a motor insurance portfolio from a Belgian insurer. Further, (Henckaerts et al., 2018) used a data-driven method to bin the continuous variables `AgePH`, `AgeCar` and `BonusMalus` into categorical variables. These bins are given in Table C.3. We denote these binned versions as `AgePHBin`, `AgeCarBin` and `BonusMalusBin`. By default, we use these binned versions in the data-generating claim frequency and claim severity model (see Table C.4).

Table C.3: Bins of the continuous variables as used in Henckaerts et al. (2018).

Variable	Bins
`AgePHBin`	$[18, 26]; (26, 30]; (30, 36]; (36, 50]; (50, 60]; (60, 65]; (65, 70]; (70, 80]$
`AgeCarBin`	$[0, 5]; (5, 10]; (10, 20]; (20, \max(\texttt{AgeCar})]$
`BonusMalusBin`	$[0, 1); [1, 2); [2, 3); [3, 7); [7, 9); [9, 11); [11, 22]$

Table C.4: Default specification of the claim frequency and claim severity model.

Variable	β_{cf}	β_{cs}
(Intercept)	-2.18	6.06
`AgePH`:		
[18,26] (reference)		
(26,30]	log(0.85)	log(0.85)
(30,36]	log(0.75)	log(0.75)
(36,50]	log(0.70)	log(0.85)
(50,60]	log(0.60)	log(0.85)
(60,65]	log(0.55)	log(1.15)
(65,70]	log(0.60)	log(1.25)
(70, max(`AgePH`)]	log(0.70)	log(1.50)
`Coverage`:		
`TPL` (reference)		
`PO`	-0.12	-0.16
`FO`	-0.11	0.11
`AgeCarBin`:		
(0, 5] (reference)		
(5,10]	log(0.90)	0
(10,20]	log(0.80)	0
(20, max(`AgeCar`)]	log(0.60)	0
`Fuel`:		
`Gasoline/LPG/Other` (reference)		
`Diesel`	log(1.19)	0
`BonusMalusBin`:		
[0,1) (reference)		
[1,2)	0.12	0.10
[2,3)	0.18	0.15
[3,7)	0.34	0.15
[7,9)	0.48	0.15
[9,11)	0.54	0.20
[11,22]	0.78	0.30

C.5 Data generating fraud model and class imbalance

Table C.5 depicts the default specification of the data-generating fraud model. The column names indicate which features are included, while the values represent the corresponding value in $\boldsymbol{\beta}_f$. Further, in the data-generating model we use the normalized version of the features `ClaimAmount`, `ClaimAge`, `n1.size`, `n2.size`, `AgePH` and `n2.ratioFraud` (see (26)). Hereby, we bring all features to the same scale. This ensures that the features' effect sizes, as specified in $\boldsymbol{\beta}_f$, are comparable. Furthermore, at the end of every iteration in Algorithm 4, we normalize both `n2.ratioFraud` and `n2.ratioNonFraud`. We do so since the network grows with every step in the algorithm. By normalizing these features, we aim to mitigate the influence of fluctuating values across iterations (also see Appendix C.6).

Table C.5: Default specification of the data-generating fraud model.

	`ClaimAmount`	`ClaimAge`	`n1.size`	`n2.size`	`NrContractsPH`	`AgePH`	`n2.ratioFraud`
$\boldsymbol{\beta}_f$	0.20	-0.35	2.00	-2.00	-1.50	-2.00	3.00

Further, the simulation engine allows us to specify the desired level of class imbalance p_t. We achieve this by employing the following approach in the third step of Algorithm 4, where we generate the claim labels

$$Y_{ijk} \sim \text{Bern}\left(\pi_{ijk}\right) \quad \text{and} \quad \pi_{ijk} = \frac{e^{{}_0\beta_f + {}_f\boldsymbol{x}_{ijk}^\top \boldsymbol{\beta}_f}}{1 + e^{{}_0\beta_f + {}_f\boldsymbol{x}_{ijk}^\top \boldsymbol{\beta}_f}}. \tag{27}$$

Before generating Y_{ijk}, we set a seed for the random number generator to ensure reproducibility. We achieve the desired level of imbalance p_t by optimizing

$$\min_{{}_0\beta_f} |p_t - {}_ap| \tag{28}$$

where

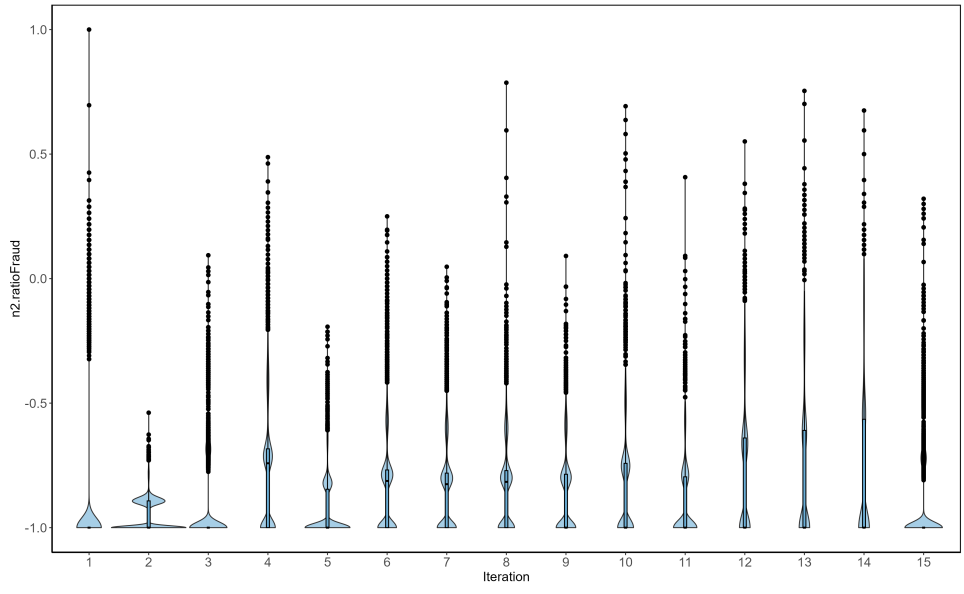
$${}_ap = \frac{\sum_{i,j,k} I(Y_{ijk} = \texttt{fraudulent})}{\sum_{i,j,k} I(Y_{ijk} = \texttt{fraudulent} \text{ OR } Y_{ijk} = \texttt{non-fraudulent})} \tag{29}$$

denotes the actual level of class imbalance in the synthetic data set (using all available claim labels). Here, $I(\cdot)$ represents the indicator function.

C.6 Distribution values n2.ratioFraud

Figure C.9 depicts the features values of `n2.ratioFraud` across the different iterations of Algorithm 4 when generating the claim labels in a synthetic data set. The horizontal axis depicts the iteration and the vertical axis the feature values. Per iteration, we show the empirical distribution of `n2.ratioFraud` in the random subset (see Algorithm 4). In the first iterations, we have a small number of fraudulent claims. As a consequence, most observations have similar feature values for `n2.ratioFraud`. The scarcity of distinct values is evident from the compactness of the violin plots. With each iteration, the number of fraudulent claims grows, resulting in an increase in distinct values for `n2.ratioFraud`. In Figure C.9, this is reflected by the increase in width of the violin plots.

Figure C.9: Distribution of the values of `n2.ratioFraud` in each iteration. Per iteration, the violin plot depicts the density of `n2.ratioFraud` in the subset.



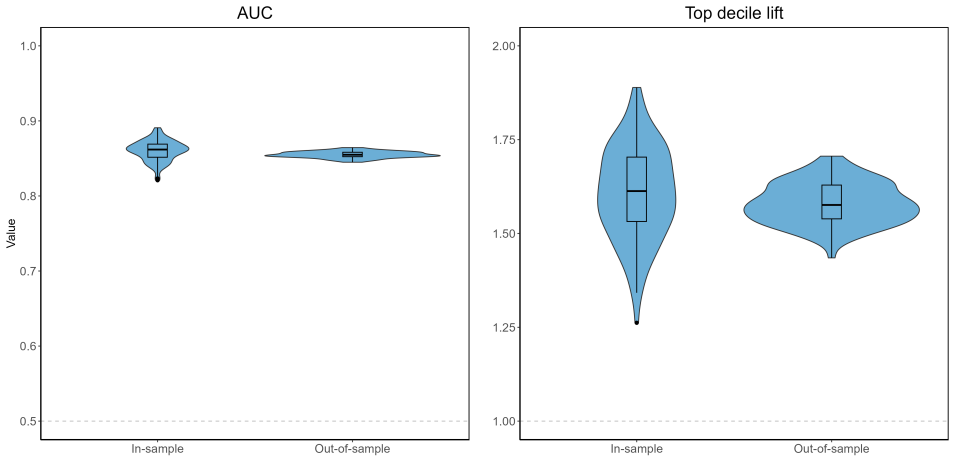
C.7 Predictive performance in the synthetic data sets

To obtain insights into the results in Section 4.4, we first fit the following model to the investigated claims

$$\begin{aligned}\text{logit}(E[Y_{ijk}]) = \ & {}_0\beta_f + {}_1\beta_f \texttt{AgePH}_i + {}_2\beta_f \texttt{NrContractsPH}_{ij} + {}_3\beta_f \texttt{ClaimAmount}_{ijk} \\ & + {}_4\beta_f \texttt{ClaimAge}_{ijk} + {}_5\beta_f \texttt{n1.size}_{ijk} + {}_6\beta_f \texttt{n2.size}_{ijk} \\ & + {}_7\beta_f \texttt{n2.ratioFraud}_{ijk}.\end{aligned} \tag{30}$$

Hence, here we use the ground-truth Y_{ijk} instead of the expert judgement $Y_{ijk}^{\texttt{expert}}$ as the response variable. Hereafter, we examine the performance on the investigated (i.e. the in-sample data set) and uninvestigated (i.e. the out-of-sample data set) claims. Figure C.10 depicts the performance on the in- and out-of-sample data sets. Both the AUC and TDL indicate that the fitted models are substantially better than a random model. In addition, compared to the in-sample data sets, the performance is slightly lower on the out-of-sample data sets. Hence, the model is able to capture the underlying relationships between the predictors and the target variable.

Figure C.10: Distribution of the performance measures in $\mathcal{D}^{Network}$. The grey line in the plots corresponds to the performance of a random model. The in-sample performance is evaluated using the ground truth label Y_{ijk} of the investigated claims. The out-of-sample performance is assessed using the ground truth label Y_{ijk} of the uninvestigated claims.



List of Figures

List of Tables

Bibliography

Agresti, A. (2013). *Categorical data analysis.*. 3rd ed. edn. Hoboken: Wiley.

Ahmad, A., Ray, S. K. and Aswani Kumar, C. (2019). Clustering mixed datasets by using similarity features. *in* 'Sustainable Communication Networks and Application'. Lecture Notes on Data Engineering and Communications Technologies. Cham: Springer International Publishing. pp. 478–485.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control* **19**(6), 716–723.

Albashrawi, M. (2016). Detecting financial fraud using data mining techniques: A decade review from 2004 to 2015. *Journal of Data Science* **14**(3), 553–570.

Andresen, M. A. and Felson, M. (2009). The impact of co-offending. *British Journal of Criminology* **50**(1), 66–81.

Antonio, K. and Beirlant, J. (2007). Actuarial statistics with generalized linear mixed models. *Insurance, Mathematics & Economics* **40**(1), 58–76.

Antonio, K., Frees, E. and Valdez, E. (2010). A multilevel analysis of intercompany claim counts. *ASTIN Bulletin* **40**(1), 151–177.

Argyrou, A. (2009). Clustering hierarchical data using self-organizing map: A graph-theoretical approach. *in* 'Advances in Self-Organizing Maps'. Vol. 5629 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. Berlin, Heidelberg. pp. 19–27.

Arora, S., May, A., Zhang, J. and Ré, C. (2020). Contextual embeddings: When are they worth it? arXiv: 2005.09117. Available at: `https://arxiv.org/abs/2005.09117`.

Australian Bureau of Statistics and New Zealand (2006). *Australian and New Zealand Standard Industrial Classification, (ANZSIC) 2006*. Australian Bureau of Statistics : Statistics New Zealand.

Avanzi, B., Taylor, G., Wang, M. and Wong, B. (2021). Synthetic: An individual insurance claim simulator with feature control. *Insurance: Mathematics and Economics* **100**(2), 296–308.

Avanzi, B., Taylor, G., Wang, M. and Wong, B. (2023). Machine learning with high-cardinality categorical features in actuarial applications. arXiv: 2301.12710. Available at: https://arxiv.org/abs/2301.12710

Baesens, B. (2023). Fraud analytics: a research agenda. *Journal of Chinese Economic and Business Studies* **21**(1), 137–141. https://doi.org/10.1080/14765284.2022.2162246

Baesens, B., Van Vlasselaer, V. and Verbeke, W. (2015). *Fraud analytics using descriptive, predictive, and social network techniques : a guide to data science for fraud detection*. 1st edition edn. Hoboken, New Jersey: Wiley.

Barman, S., Pal, U., Sarfaraj, M. A., Biswas, B., Mahata, A. and Mandal, P. (2016). A complete literature review on financial fraud detection applying data mining techniques. *International Journal of Trust Management in Computing and Communications* **3**(4), 336–359.

Bates, D., Mächler, M., Bolker, B. and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software* **67**(1), 1–48.

Beale, E. M. L., Kendall, M. G. and Mann, D. W. (1967). The discarding of variables in multivariate analysis. *Biometrika* **54**(3), 357–366.

Beirlant, J., Goegebeur, Y., Segers, J. and Teugels, J. L. (2005). *Statistics of extremes: theory and applications*. Chichester: Wiley.

Bennett, D. A. (2001). How can I deal with missing data in my study. *Australian and New Zealand Journal of Public Health* **25**(5), 464–469.

Blier-Wong, C., Cossette, H., Lamontagne, L. and Marceau, E. (2021). Machine learning in P&C insurance: A review for pricing and reserving. *Risks* **9**(4), 4.

Bolker, B. M., Brooks, M. E., Clark, C. J., Geange, S. W., Poulsen, J. R., Stevens, M. H. H. and White, J.-S. S. (2009). Generalized linear mixed models: a practical guide for ecology and evolution. *Trends in Ecology & Evolution* **24**(3), 127–135. https://www.sciencedirect.com/science/article/pii/S0169534709000196

Bolker, B. et al. (2022). GLMM FAQ. Viewed 07 November 2022. `https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html#convergence-warnings`.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* **88**(421), 9–25.

Brown, H. and Prescott, R. (2006). *Applied Mixed Models in Medicine*. Wiley.

Bühlmann, H. and Gisler, A. (2006). *A course in credibility theory and its applications*. Berlin/Heidelberg: Springer.

Bureau Van Dijk (2020). Bel-first: financial reports and statistics on Belgian and Luxembourg companies. Viewed 12 March 2021. https://belfirst.bvdinfo.com.

Burnham, K. and Anderson, D. (2002). *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. 2nd ed. edn. New York: Springer.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics* **3**(1), 1–27.

Campo, B. D. C. and Antonio, K. (2023). Insurance pricing with hierarchically structured data: an illustration with a workers' compensation insurance portfolio. *Scandinavian Actuarial Journal* **2023**(9), 853–884. https://doi.org/10.1080/03461238.2022.2161413

Carrizosa, E., Galvis Restrepo, M. and Romero Morales, D. (2021). On clustering categories of categorical predictors in generalized linear models. *Expert Systems with Applications* **182**, 115245.

Carrizosa, E., Mortensen, L. H., Romero Morales, D. and Sillero-Denamiel, M. R. (2022). The tree based linear regression model for hierarchical categorical variables. *Expert Systems with Applications* **203**, 117423.

Cer, D., Yang, Y., yi Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B. and Kurzweil, R. (2018). Universal Sentence Encoder. arXiv: 1803.11175. Available at: `https://arxiv.org/abs/1803.11175`.

Cheung, Y.-m. and Jia, H. (2013). Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number. *Pattern Recognition* **46**(8), 2228–2238.

Colbrook, M. J., Antun, V. and Hansen, A. C. (2022). The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and smale's 18th problem. *Proceedings of the National Academy of Sciences* **119**(12), e2107151119. https://www.pnas.org/doi/abs/10.1073/pnas.2107151119

Costa, I. G., de Carvalho, F. d. A. and de Souto, M. C. (2004). Comparative analysis of clustering methods for gene expression time course data. *Genetics and Molecular Biology* **27**, 623–631.

Dannenburg, D. R., Kaas, R. and Goovaerts, M. J. (1996). *Practical actuarial credibility models*. Amsterdam: IAE (Institute of Actuarial Science and Econometrics of the University of Amsterdam).

Dastile, X., Celik, T. and Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing* **91**, 106263. https://www.sciencedirect.com/science/article/pii/S1568494620302039

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-1**(2), 224–227.

de Jong, P. and Heller, G. (2008). *Generalized linear models for insurance data*. Cambridge: Cambridge University Press.

de Souto, M. C., Costa, I. G., de Araujo, D. S., Ludermir, T. B. and Schliep, A. (2008). Clustering cancer gene expression data: a comparative study. *BMC Bioinformatics* **9**(1), 1–14.

Delong, Ł., Lindholm, M. and Wüthrich, M. V. (2021). Making Tweedie's compound Poisson model more accessible. *European Actuarial Journal* **11**(1), 185–226.

Denuit, M., Dhaene, J., Goovaerts, M. and Kaas, R. (2005). *Actuarial theory for dependent risks: measures, orders and models.*. West Sussex: Wiley.

Denuit, M., Hainaut, D. and Trufin, J. (2019). *Effective statistical learning methods for actuaries I: GLMs and extensions*. Cham: Springer International Publishing AG.

Denuit, M., Maréchal, X., Pitrebois, S. and Walhin, J.-F. (2007). *Actuarial modelling of claim counts: risk classification, credibility and bonus-malus systems*. Chichester: Wiley.

Denuit, M., Sznajder, D. and Trufin, J. (2019). Model selection based on Lorenz and concentration curves, Gini indices and convex order. *Insurance, Mathematics & Economics* **89**, 128–139.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv: 1810.04805. Available at: `https://arxiv.org/abs/1810.04805`.

Drikvandi, R., Verbeke, G. and Molenberghs, G. (2017). Diagnosing misspecification of the random-effects distribution in mixed models. *Biometrics* **73**(1), 63–71.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics* **4**(1), 95–104.

Ebnesajjad, S. (2011). 8 - characteristics of adhesive materials. *in* S. Ebnesajjad, ed., 'Handbook of Adhesives and Surface Preparation'. Plastics Design Library. Oxford: William Andrew Publishing. pp. 137–183.

European Central Bank (2021). Loans from euro area monetary financial institutions to non-financial corporations by economic activity: Explanatory notes. Accessed: 2023-02-03. `https://www.ecb.europa.eu/stats/pdf/money/explanatory_notes_nace-en_sdw_dissemination_en.pdf?993f98fe6b628ebc6ff44b0af3d2e362`.

European Commission and Eurostat (2017). *NACE Rev. 2: statistical classification of economic activities in the European Community*. Publications Office.

European Insurance and Occupational Pensions Authority (2019). *Big data analytics in motor and health insurance: a thematic review*. Luxembourg: Publications Office of the European Union.

European Insurance and Occupational Pensions Authority (2020). Solvency II single rulebook. Viewed 2021-11-26. `https://www.eiopa.europa.eu/rulebook/solvency-ii/article-6339_en?source=search`.

Everitt, B., Landau, S. and Leese, M. (2011). *Cluster Analysis*. fifth edn. Wiley.

Fang, J. (2019). A critical review of five machine learning-based algorithms for predicting protein stability changes upon mutation. *Briefings in Bioinformatics* **21**(4), 1285–1292. https://doi.org/10.1093/bib/bbz071

FBI (2022). Investigating insurance fraud. Accessed 2022-07-08. `https://www.fbi.gov/stats-services/publications/insurance-fraud`. https://www.fbi.gov/stats-services/publications/insurance-fraud

Ferrario, A. and Naegelin, M. (2020). The art of natural language processing: Classical, modern and contemporary approaches to text document classification. Available at SSRN: `https://ssrn.com/abstract=3547887`.

Fitzmaurice, G., Davidian, M., Verbeke, G. and Molenberghs, G. (2008). *Longitudinal data analysis: a handbook of modern statistical methods*. CRC Press.

FOD Economie (2004). *NACE-Bel: Activiteitennomenclatuur*. Algemene Directie Statistiek en Economische Informatie.

Foss, A. H., Markatou, M. and Ray, B. (2019). Distance metrics and clustering methods for mixed-type data. *International Statistical Review* **87**(1), 80–109.

Fränti, P. and Sieranoja, S. (2019). How much can k-means be improved by using better initialization and repeats?. *Pattern Recognition* **93**, 95–112.

Frees, E. (2015). Analytics of insurance markets. *Annual Review of Financial Economics* **7**, 253–277.

Frees, E., Derrig, R. and Meyers, G. (2014). *Predictive modeling applications in actuarial science: volume 1, predictive modeling techniques*. New York: Cambridge University Press.

Frees, E. W. (2010). *Regression modeling with actuarial and financial applications*. New York: Cambridge University Press.

Frees, E. W., Gao, J. and Rosenberg, M. A. (2011). Predicting the frequency and amount of health care expenditures. *North American Actuarial Journal* **15**(3), 377–392.

Frees, E. W. and Valdez, E. A. (2008). Hierarchical insurance claims modeling. *Journal of the American Statistical Association* **103**(484), 1457–1469.

Frees, E. W., Young, V. R. and Luo, Y. (1999). A longitudinal data analysis interpretation of credibility models. *Insurance Mathematics and Economics* **24**(3), 229–247.

Gabrielli, A. and Wüthrich, M. (2018). An individual claims history simulation machine. *Risks* **6**(2).

Gao, G. and Wüthrich, M. V. (2018). Feature extraction from telematics car driving heatmaps. *European Actuarial Journal* **8**(2), 383–406.

Garrido, J., Genest, C. and Schulz, J. (2016). Generalized linear models for dependent frequency and severity of insurance claims. *Insurance, Mathematics & Economics* **70**, 205–215.

Gelman, A. and Hill, J. (2017). *Data analysis using regression and multilevel/hierarchical models*. 17th pr. edn. Cambridge: Cambridge University Press.

Gertheiss, J. and Tutz, G. (2010). Sparse modeling of categorial explanatory variables. *Ann. Appl. Stat.* **4**(4), 2150–2180.

Ghobadi, F. and Rohani, M. (2016). Cost sensitive modeling of credit card fraud using neural network strategy. *in* '2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS)'. IEEE. pp. 1–5.

Gini, C. (1921). Measurement of inequality of incomes. *The Economic Journal* **31**(121), 124–126.

Goldburd, M., Khare, A., Tevet, D. and Guller, D. (2016). Generalized linear models for insurance rating. *Casualty Actuarial Society, CAS Monographs Series* **5**.

Gomes, C., Jin, Z. and Yang, H. (2021). Insurance fraud detection with unsupervised deep learning. *The Journal of Risk and Insurance* **88**(3), 591–624.

Govender, P. and Sivakumar, V. (2020). Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980-2019). *Atmospheric Pollution Research* **11**(1), 40–56.

Guo, C. and Berkhahn, F. (2016). Entity embeddings of categorical variables. arXiv: 1604.06737. Available at https://arxiv.org/abs/1604.06737

Haberman, S. and Renshaw, A. E. (1996). Generalized linear models and actuarial science. *Journal of the Royal Statistical Society: Series D (The Statistician)* **45**(4), 407–436.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems* **17**(2), 107–145.

Hanley, J. and McNeil, B. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology* **143**(1), 29–36.

Harrison, X. A., Donaldson, L., Correa-Cano, M. E., Evans, J., Fisher, D. N., Goodwin, C. E., Robinson, B. S., Hodgson, D. J. and Inger, R. (2018). A brief introduction to mixed effects modelling and multi-model inference in ecology. *PeerJ (San Francisco, CA)* **2018**(5), e4794–e4794.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.

He, X., Gao, M., Kan, M.-Y. and Wang, D. (2017). Birank: Towards ranking on bipartite graphs. *IEEE Transactions on Knowledge and Data Engineering* **29**(1), 57–71.

Henckaerts, R., Antonio, K., Clijsters, M. and Verbelen, R. (2018). A data driven binning strategy for the construction of insurance tariff classes. *Scandinavian Actuarial Journal* **2018**(8), 681–705.

Henckaerts, R., Antonio, K. and Côté, M.-P. (2022). When stakes are high: Balancing accuracy and transparency with model-agnostic interpretable data-driven surrogates. *Expert Systems with Applications* **202**, 117230. https://www.sciencedirect.com/science/article/pii/S0957417422006042

Henckaerts, R., Côté, M.-P., Antonio, K. and Verbelen, R. (2021). Boosting insights in insurance tariff plans with tree-based machine learning methods. *North American Actuarial Journal* **25**(2), 255–285.

Hennig, C. (2015). What are the true clusters?. *Pattern Recognition Letters* **64**, 53–62.

Hilal, W., Gadsden, S. A. and Yawney, J. (2022). Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Systems with Applications* **193**, 116429. https://www.sciencedirect.com/science/article/pii/S0957417421017164

Hocking, R. R. and Leslie, R. N. (1967). Selection of the best subset in regression analysis. *Technometrics* **9**(4), 531–540.

Höfling, H., Binder, H. and Schumacher, M. (2010). A coordinate-wise optimization algorithm for the fused lasso. arXiv: 1011.6409. Available at: `https://arxiv.org/abs/1011.6409`.

Holizki, T., McDonald, R., Foster, V. and Guzmicky, M. (2008). Causes of work-related injuries among young workers in British Columbia. *American Journal of Industrial Medicine* **51**(5), 357–363.

Hsu, C.-C. (2006). Generalizing self-organizing map for categorical data. *IEEE Transactions on Neural Networks* **17**(2), 294–304.

Jensen, D. (1997). Prospective assessment of ai technologies for fraud detection: A case study. *in* 'AAAI Workshop on AI Approaches to Fraud Detection and Risk Management'. Citeseer. pp. 34–38.

Jewell, W. S. (1975). The use of collateral data in credibility theory : a hierarchical model. *Giornale dell'Istituto Italiano degli Attuari* **38**, 1–16.

Jørgensen, B. and Souza, M. C. P. D. (1994). Fitting tweedie's compound poisson model to insurance claims data. *Scandinavian Actuarial Journal* **1994**(1), 69–93.

Jung, Y. G., Kang, M. S. and Heo, J. (2014). Clustering performance comparison using k-means and expectation maximization algorithms. *Biotechnology & Biotechnological Equipment* **28**(sup1), S44–S48.

Kaufman, L. and Rousseeuw, P. (1990*a*). *Finding groups in data: an introduction to cluster analysis.*. New York (N.Y.): Wiley.

Kaufman, L. and Rousseeuw, P. (1990*b*). *Partitioning Around Medoids (Program PAM)*. John Wiley & Sons, Ltd. chapter 2, pp. 68–125.

KBC Brussels (n.d.). Accident with no dispute, or claim on comprehensive insurance. Accessed: 2023-01-27. `https://www.kbcbrussels.be/retail/en/insurance/vehicle/damage-or-theft/accident-with-no-dispute-or-claim-on-comprehensive-insurance.html`.

Kho, J. R. D. and Vea, L. A. (2017). Credit card fraud detection based on transaction behavior. *in* 'TENCON 2017-2017 IEEE Region 10 Conference'. IEEE. pp. 1880–884.

Khondoker, M., Dobson, R., Skirrow, C., Simmons, A. and Stahl, D. (2016). A comparison of machine learning methods for classification using simulation with multiple real data examples from mental health studies. *Statistical Methods in Medical Research* **25**(5), 1804–1823. PMID: 24047600. https://doi.org/10.1177/0962280213502437

Kinnunen, T., Sidoroff, I., Tuononen, M. and Fränti, P. (2011). Comparison of clustering methods: A case study of text-independent speaker modeling. *Pattern Recognition Letters* **32**(13), 1604–1617.

Kläs, M. and Vollmer, A. M. (2018). Uncertainty in machine learning applications: A practice-driven classification of uncertainty. *in* B. Gallina, A. Skavhaug, E. Schoitsch and F. Bitsch, eds, 'Computer Safety, Reliability, and Security'. Springer International Publishing. Cham. pp. 431–438.

Kogan, J., Nicholas, C. and Teboulle, M. (2005). *Grouping Multidimensional Data: Recent Advances in Clustering*. Berlin, Heidelberg: Springer Berlin / Heidelberg.

Kohonen, T. (1995). *Self-organizing maps*. Berlin: Springer.

Kou, G., Peng, Y. and Wang, G. (2014). Evaluation of clustering algorithms for financial risk analysis using MCDM methods. *Information Sciences* **275**, 1–12.

Kumar, P. (2010). Probability distributions and estimation of ali-mikhail-haq copula. *Applied Mathematical Sciences* **4**(14), 657–666.

Lakshminarayanan, B., Pritzel, A. and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* **30**.

Lee, G. Y., Manski, S. and Maiti, T. (2020). Actuarial applications of word embedding models. *ASTIN Bulletin: The Journal of the IAA* **50**(1), 1–24.

Lemmens, A. and Croux, C. (2006). Bagging and boosting classification trees to predict churn. *Journal of Marketing Research* **43**(2), 276–286.

Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J. and Wu, S. (2013). Understanding and enhancement of internal clustering validation measures. *IEEE Transactions on Cybernetics* **43**(3), 982–994.

Lopez-Rojas, E. A., Gorton, D. and Axelsson, S. (2015). Using the RetSim simulator for fraud detection research. *International Journal of Simulation and Process Modelling* **10**(2), 144–155.

Lorenz, M. O. (1905). Methods of measuring the concentration of wealth. *Publications of the American Statistical Association* **9**(70), 209–219.

Luong, T., Socher, R. and Manning, C. (2013). Better word representations with recursive neural networks for morphology. *in* 'Proceedings of the Seventeenth Conference on Computational Natural Language Learning'. Association for Computational Linguistics. Sofia, Bulgaria. pp. 104–113. https://aclanthology.org/W13-3512

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. *in* 'Proceedings of the fifth Berkeley symposium on mathematical statistics and probability'. Vol. 1. Oakland, CA, USA. pp. 281–297.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M. and Hornik, K. (2022). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.4. https://CRAN.R-project.org/package=cluster

Mangiameli, P., Chen, S. K. and West, D. (1996). A comparison of SOM neural network and hierarchical clustering methods. *European Journal of Operational Research* **93**(2), 402–417.

Marques, A. I., Garcia, V. and Sanchez, J. S. (2013). On the suitability of resampling techniques for the class imbalance problem in credit scoring. *The Journal of the Operational Research Society* **64**(7), 1060–1070.

McCullagh, P. and Nelder, J. A. (1999). *Generalized linear models*. London: Chapman and Hall.

McCulloch, C. E. and Neuhaus, J. M. (2011). Prediction of random effects in linear and generalized linear models under model misspecification. *Biometrics* **67**(1), 270–279.

McGowan, M. A., Andrews, D. and Millot, V. (2018). The walking dead? Zombie firms and productivity performance in OECD countries. *Economic Policy* **33**(96), 685–736.

McNicholas, P. (2016*a*). *Mixture Model-Based Classification*. CRC Press.

McNicholas, P. D. (2016*b*). Model-based clustering. *Journal of Classification* **33**(3), 331–373.

Micci-Barreca, D. (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *SIGKDD explorations* **3**(1), 27–32.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv: 1301.3781. Available at: `https://arxiv.org/abs/1301.3781`.

Mohammad, S. M. and Hirst, G. (2012). Distributional measures of semantic distance: A survey. arXiv: 1203.1858. Available at: `https://arxiv.org/abs/1203.1858`.

Molenberghs, G. and Verbeke, G. (2005). *Models for discrete longitudinal data*. New York: Springer New York.

Molenberghs, G. and Verbeke, G. (2011). A note on a hierarchical interpretation for negative variance components. *Statistical Modelling* **11**(5), 389–408.

Morris, T. P., White, I. R. and Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine* **38**(11), 2074–2102.

Murugesan, N., Cho, I. and Tortora, C. (2021). Benchmarking in cluster analysis: A study on spectral clustering, dbscan, and k-means. *in* 'Data Analysis and Rationality in a Complex World'. Vol. 5 of *Studies in Classification, Data Analysis, and Knowledge Organization*. Cham: Springer International Publishing. pp. 175–185.

Newman, M. (2010). *Networks : an introduction*. Oxford: Oxford University Press.

Ng, A., Jordan, M. and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems* **14**.

Ngai, E., Hu, Y., Wong, Y., Chen, Y. and Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems* **50**(3), 559–569.

Nur Prasasti, I. M., Dhini, A. and Laoh, E. (2020). Automobile insurance fraud detection using supervised classifiers. *in* '2020 International Workshop on Big Data and Information Security (IWBIS)'. IEEE. pp. 47–52.

Oelker, M.-R., Gertheiss, J. and Tutz, G. (2014). Regularization and model selection with categorical predictors and effect modifiers in generalized linear models. *Statistical Modelling* **14**(2), 157–177.

Ohlsson, E. (2005). Simplified estimation of structure parameters in hierarchical credibility. Presented at the Zurich ASTIN Colloquium. `http://www.actuaries.org/ASTIN/Colloquia/Zurich/Ohlsson.pdf`

Ohlsson, E. (2008). Combining generalized linear models and credibility models in practice. *Scandinavian Actuarial Journal* **2008**(4), 301–314.

Ohlsson, E. and Johansson, B. (2010). *Non-life insurance pricing with generalized linear models*. Berlin, Heidelberg: Springer.

Oliveira, I., Molenberghs, G., Verbeke, G., Demetrio, C. and Dias, C. (2017). Negative variance components for non-negative hierarchical data with correlation, over-, and/or underdispersion. *Journal of Applied Statistics* **44**(6), 1047–1063.

Onan, A. (2017). A k-medoids based clustering scheme with an application to document clustering. *in* '2017 International Conference on Computer Science and Engineering (UBMK)'. pp. 354–359.

Oommen, T., Baise, L. G. and Vogel, R. M. (2011). Sampling bias and class imbalance in maximum-likelihood logistic regression. *Mathematical Geosciences* **43**(1), 99–120.

Óskarsdóttir, M., Ahmed, W., Antonio, K., Baesens, B., Dendievel, R., Donas, T. and Reynkens, T. (2022). Social network analytics for supervised fraud detection in insurance. *Risk Analysis* **42**(8), 1872–1890.

Ostrovsky, R., Rabani, Y., Schulman, L. and Swamy, C. (2012). The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM* **59**(6), 1–22.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B. and Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. https://arxiv.org/abs/1906.02530

Page, L., Brin, S., Motwani, R. and Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web.. Technical report. Stanford InfoLab.

Pargent, F., Pfisterer, F., Thomas, J. and Bischl, B. (2022). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics* **37**(5), 2671–2692.

Park, J. and Barabási, A.-L. (2007). Distribution of node characteristics in complex networks. *Proceedings of the National Academy of Sciences - PNAS* **104**(46), 17916–17920.

Parodi, P. (2014). *Pricing in general insurance*. New York: CRC Press.

Phillips, J. M. (2021). *Mathematical Foundations for Data Analysis*. Cham: Springer International Publishing.

Pinheiro, J., Pinheiro, J. and Bates, D. (2009). *Mixed-Effects Models in S and S-PLUS*. Springer.

Poon, L. K. M., Liu, A. H., Liu, T. and Zhang, N. L. (2012). A model-based approach to rounding in spectral clustering. arXiv: 1210.4883. Available at: `https://arxiv.org/abs/1210.4883`.

Pourhabibi, T., Ong, K.-L., Kam, B. H. and Boo, Y. L. (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems* **133**, 113303. https://www.sciencedirect.com/science/article/pii/S0167923620300580

Pryseley, A., Tchonlafi, C., Verbeke, G. and Molenberghs, G. (2011). Estimating negative variance components from gaussian and non-gaussian data: A mixed models approach. *Computational Statistics & Data Analysis* **55**(2), 1071–1085. https://www.sciencedirect.com/science/article/pii/S0167947310003452

Quijano Xacur, O. A. and Garrido, J. (2015). Generalised linear models for aggregate claims: to Tweedie or not?. *European Actuarial Journal* **5**(1), 181–202.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Reiss, A. J. (1988). Co-offending and criminal careers. *Crime and Justice (Chicago, Ill.)* **10**, 117–170.

Rentzmann, S. and Wuthrich, M. V. (2019). Unsupervised learning: What is a sports car?. Available at SSRN: https://ssrn.com/abstract=3439358 or http://dx.doi.org/10.2139/ssrn.3439358.

Reynkens, T., Devriendt, S. and Antonio, K. (2018). *smurf: Sparse Multi-Type Regularized Feature Modeling*. R package version 1.0.0. https://CRAN.R-project.org/package=smurf

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. d. F. and Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PloS One* **14**(1), e0210236.

Rosenberg, M. and Zhong, F. (2022). Using clusters based on social determinants to identify the top 5% utilizers of health care. *North American Actuarial Journal* **26**(3), 456–469.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* **20**, 53–65.

Roy, R. and George, K. T. (2017). Detecting insurance claims fraud using machine learning techniques. *in* '2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT)'. IEEE. pp. 1–6.

Saefken, B., Kneib, T., van Waveren, C.-S. and Greven, S. (2014). A unifying approach to the estimation of the conditional akaike information in generalized linear mixed models. *Electronic Journal of Statistics* **8**(1).

Schomacker, T. and Tropmann-Frick, M. (2021). Language representation models: An overview. *Entropy (Basel, Switzerland)* **23**(11), 1422.

Schubert, E. (2021). A triangle inequality for cosine similarity. *in* 'Similarity Search and Applications'. Lecture Notes in Computer Science. Cham: Springer International Publishing. pp. 32–44.

Schwertman, N. C., Owens, M. A. and Adnan, R. (2004). A simple more general boxplot method for identifying outliers. *Computational Statistics & Data Analysis* **47**(1), 165–174.

Sela, R. J. and Simonoff, J. S. (2012). RE-EM trees: a data mining approach for longitudinal and clustered data. *Machine Learning* **86**(2), 169–207.

Skrondal, A. and Rabe-Hesketh, S. (2009). Prediction in multilevel generalized linear models. *Journal of the Royal Statistical Society. Series A, Statistics in society* **172**(3), 659–687.

Smyth, G. K. and Jørgensen, B. (2002). Fitting Tweedie's compound Poisson model to insurance claims data: Dispersion modelling. *ASTIN Bulletin* **32**(1), 143–157.

So, B., Boucher, J.-P. and Valdez, E. A. (2021). Synthetic dataset generation of driver telematics. *Risks* **9**(4).

Srivastava, A., Yadav, M., Basu, S., Salunkhe, S. and Shabad, M. (2016). Credit card fraud detection at merchant side using neural networks. *in* '2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)'. Bharati Vidyapeeth, New Delhi as the Organizer of INDIACom - 2016. pp. 667–670.

Stassen, B., Denuit, M., Mahy, S., Maréchal, X. and Trufin, J. (2017). A unified approach for the modelling of rating factors in workers compensation insurance. White paper by Reacfin. Available at: https://www.reacfin.com/wp-content/uploads/2016/12/170131-Reacfin-White-Paper-A-Unified-Approach-for-the-Modeling-of-Rating-Factors-in-Work-ers%E2%80%99-Compensation-Insurance.pdf.

Statistical Office of the European Communities (1996). *NACE Rev. 1: Statistical Classification of Economic Activities in the European Community*. Office for Official Publications of the European Communities.

Storchmann, K. (2004). On the depreciation of automobiles: An international comparison. *Transportation (Dordrecht)* **31**(4), 371–408.

Struyf, A., Hubert, M. and Rousseeuw, P. (1997). Clustering in an object-oriented environment. *Journal of Statistical Software* **1**(1), 1–30.

Subudhi, S. and Panigrahi, S. (2020). Use of optimized fuzzy c-means clustering and supervised classifiers for automobile insurance fraud detection. *Journal of King Saud University. Computer and Information Sciences* **32**(5), 568–575.

Sundarkumar, G. G. and Ravi, V. (2015). A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence* **37**, 368–377.

Thabtah, F., Hammoud, S., Kamalov, F. and Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. *Information Sciences* **513**, 429–441. https://www.sciencedirect.com/science/article/pii/S0020025519310497

Timm, N. H. (2002). *Applied multivariate analysis*. Pittsburgh: Springer.

Tohme, T., Vanslette, K. and Youcef-Toumi, K. (2022). Reliable neural networks for regression uncertainty estimation. *Reliability Engineering & System Safety* p. 108811.

Troxler, A. and Schelldorfer, J. (2022). Actuarial applications of natural language processing using transformers: Case studies for using text features in an actuarial context. arXiv: 2206.02014. Available at: `https://arxiv.org/abs/2206.02014`.

Tuerlinckx, F., Rijmen, F., Verbeke, G. and De Boeck, P. (2006). Statistical inference in generalized linear mixed models: A review. *British Journal of Mathematical & Statistical Psychology* **59**(2), 225–255.

Tukey, J. W. (1977). *Exploratory data analysis.*. Reading: Addison-Wesley.

Tumminello, M., Consiglio, A., Vassallo, P., Cesari, R. and Farabullini, F. (2023). Insurance fraud detection: A statistically validated network approach. *The Journal of Risk and Insurance* **90**(2), 381–419.

Tutz, G. and Oelker, M. (2017). Modelling clustered heterogeneity: Fixed effects, random effects and mixtures. *International Statistical Review* **85**(2), 204–227.

van den Goorbergh, R., van Smeden, M., Timmerman, D. and Van Calster, B. (2022). The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression. *Journal of the American Medical Informatics Association* **29**(9), 1525–1534. https://doi.org/10.1093/jamia/ocac093

Van Der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research* **9**, 2579–2625.

van Koppen, M. V., de Poot, C. J., Kleemans, E. R. and Nieuwbeerta, P. (2010). Criminal trajectories in organized crime. *British Journal of Criminology* **50**(1), 102–123.

Van Vlasselaer, V., Eliassi-Rad, T., Akoglu, L., Snoeck, M. and Baesens, B. (2016). GOTCHA! network-based fraud detection for social security fraud. *Management Science* **63**(9), 3090–3110.

Vanacker, T. R. and Manigart, S. (2008). Pecking order and debt capacity considerations for high-growth companies seeking financing. *Small Business Economics* **35**(1), 53–69.

Vendramin, L., Campello, R. J. G. B. and Hruschka, E. R. (2010). Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining: The ASA Data Science Journal* **3**(4), 209–235.

Verma, D. and Meila, M. (2003). A comparison of spectral clustering algorithms. Technical report. University of Washington Tech Rep UWCSE030501.

Verma, V. K., Pandey, M., Jain, T. and Tiwari, P. K. (2021). Dissecting word embeddings and language models in natural language processing. *Journal of Discrete Mathematical Sciences & Cryptography* **24**(5), 1509–1515.

von Luxburg, U. (2007). A tutorial on spectral clustering. arXiv: 0711.0189. Available at: `https://arxiv.org/abs/0711.0189`.

von Luxburg, U., Belkin, M. and Bousquet, O. (2008). Consistency of spectral clustering. *The Annals of Statistics* **36**(2), 555–586.

von Luxburg, U., Bousquet, O. and Belkin, M. (2004). On the convergence of spectral clustering on random samples: The normalized case. *in* 'LEARNING THEORY, PROCEEDINGS'. Vol. 3120 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. Berlin, Heidelberg. pp. 457–471.

Vosseler, A. (2022). Unsupervised insurance fraud prediction based on anomaly detector ensembles. *Risks (Basel)* **10**(7), 132–.

Walters, J. K., A. Christensen, K., K. Green, M., E. Karam, L. and D. Kincl, L. (2010). Occupational injuries to oregon workers 24 years and younger: An analysis of workers' compensation claims, 2000-2007. *American Journal of Industrial Medicine* **53**(10), 984–994.

Wang, H. and Song, M. (2011). Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming. *The R journal* **3**(2), 29–33.

Wang, X. and Keogh, E. (2008). A clustering analysis for target group identification by locality in motor insurance industry. *in* 'Soft Computing Applications in Business'. Vol. 230 of *Studies in Fuzziness and Soft Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 113–127.

Warren, D. E. and Schweitzer, M. E. (2018). When lying does not pay: How experts detect insurance fraud. *Journal of Business Ethics* **150**(3), 711–726.

West, J. and Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & Security* **57**, 47–66.

Wierzchoń, S. and Kłopotek, M. (2019). *Modern Algorithms of Cluster Analysis*. Springer International Publishing.

Wood, S. (2017). *Generalized Additive Models: An Introduction with R*. 2 edn. Chapman and Hall/CRC.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)* **73**(1), 3–36.

Wurzelbacher, S. J., Meyers, A. R., Lampl, M. P., Timothy Bushnell, P., Bertke, S. J., Robins, D. C., Tseng, C.-Y. and Naber, S. J. (2021). Workers' compensation claim counts and rates by injury event/exposure among state-insured private employers in ohio, 2007-2017. *Journal of Safety Research* **79**, 148–167.

Wüthrich, M. V. (2017). Covariate selection from telematics car driving data. *European Actuarial Journal* **7**(1), 89–108.

Wüthrich, M. V. (2020). Bias regularization in neural network models for general insurance pricing. *European Actuarial Journal* **10**(1), 179–202.

Xu, S., Zhang, C. and Hong, D. (2022). BERT-based NLP techniques for classification and severity modeling in basic warranty data study. *Insurance: Mathematics and Economics* **107**, 57–67.

Yang, Y., Qian, W. and Zou, H. (2018). Insurance premium prediction via gradient tree-boosted Tweedie Compound Poisson models. *Journal of Business & Economic Statistics* **36**(3), 456–470.

Yeo, A. C., Smith, K. A., Willis, R. J. and Brooks, M. (2001). Clustering technique for risk classification and prediction of claim costs in the automobile insurance industry. *Intelligent Systems in Accounting, Finance and Management* **10**(1), 39–50.

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series* **1168**(2), 22022–.

Yu, D., Liu, G., Guo, M. and Liu, X. (2018). An improved k-medoids algorithm based on step increasing and optimizing medoids. *Expert Systems with Applications* **92**, 464–473.

Zappa, D., Borrelli, M., Clemente, G. P. and Savelli, N. (2021). Text mining in insurance: From unstructured data to meaning. *Variance* **14**(1).

Zhang, Y. (2013). Likelihood-based and Bayesian methods for Tweedie Compound Poisson linear mixed models. *Statistics and Computing* **23**.

Zhu, R. and Wüthrich, M. V. (2021). Clustering driving styles via image processing. *Annals of Actuarial Science* **15**(2), 276–290.

Zuur, A., Ieno, E., Walker, N., Saveliev, A. and Smith, G. (2009). *Mixed Effects Models and Extensions in Ecology with R*. Springer New York.

Doctoral dissertations from the Faculty of Economics and Business, see:
https://www.kuleuven.be/english/research/doctoraldefences/archive.