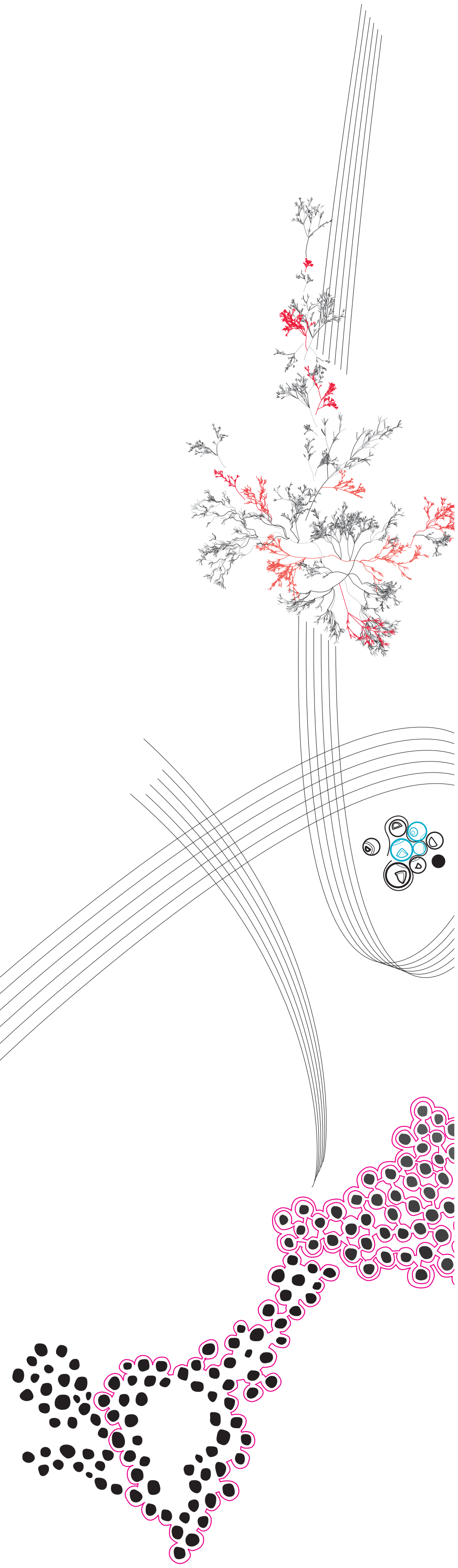
MSc Computer Science
Final Project



Evaluating Generalisability, Limitations, and Adaptations in Social Network Analytics-Based Insurance Fraud Detection

Gilian Schrijver

Supervisors:
Dr. D.K. Sarmah & Dr.Ing. M. El-hajj

Chair of assessment committee:
Dr.Ir. R. Langerak

13th February 2024

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Abstract

Despite insurance companies detecting a large worth of fraudulent insurance claims, detected insurance fraud is assumed to constitute only a small fraction of all insurance fraud. Meanwhile, a large number of claims flagged as 'potentially fraudulent' by fraud detection systems are considered benign after manual review, indicating a high error rate. These combined observations reveal that, at least in theory, there is room for improving current fraud detection systems. In the current research, we extend upon a recently proposed social network analytics-based approach to automobile insurance claims fraud detection. This approach leverages the BiRank algorithm to calculate fraud scores in a graph of claims and stakeholders, which are then combined with other features to train a supervised machine learning classifier. We first establish that our real insurance data also suggests empirical evidence for the homophily assumption proposed by the original work's authors. We then reconstruct their proposed model and corroborate their finding that the inclusion of network-related features enhances fraud classification performance. As an extension, we assess the impact of incorporating time-weighted fraud influence and extending the graph with relations based on shared resources and reveal that our current approach yields limited additional value over the baseline model. Meanwhile, we identify limitations in the original work's methodology and experimental setup. The results of this study provide a deeper understanding of the value of using graph-based insurance fraud detection techniques in practice. These insights shall ultimately aid in saving insurers and their customers from the financial consequences of fraudulent claims.

Keywords: automobile insurance, insurance fraud, fraud detection, BiRank, social networks, supervised learning, machine learning, data mining

Acknowledgements

The past year has unfolded as an interesting and fruitful journey. It began with an introduction to the concept of ‘systematic literature reviews’, which eventually led me to the quest for a suitable journal to publish my own systematic literature review in. And following initial marvel at the impressive existing studies on automobile insurance fraud detection, I later uncovered significant unmentioned limitations within one of those very studies. This experience not only expanded my knowledge and skills in both academia and insurance fraud but also underscored the complexities inherent in the field. To have been granted this opportunity is something I will forever foster.

Yet, it should be noted that this journey has been far from smooth sailing. The discipline required to continuously motivate myself to resume working on the same project that had already shaped my life for months was demanding. Meanwhile, various setbacks further challenged my motivation to proceed. Thankfully, I was not alone in this endeavour and found support from a number of great individuals, whom I would like to thank.

First of all, I would like to thank my supervisors Dipti Sarmah and Mohammed El-Hajj for their invaluable guidance and motivating words during our weekly meetings. Their support has helped shape the output of my graduation period and the quality of my work would not have been the same without their valuable contributions. Additionally, I am thankful for Rom Langerak, who enthusiastically agreed to be committee chair for my final project despite previously limited knowledge on the topic. Furthermore, I express gratitude to the insurer for providing access to the essential data and insights crucial for conducting this research. The use of real insurance data was indispensable, rendering this study unattainable without it.

Last but not least, I would like to thank my family and friends for their great support throughout my full academic journey. Special thanks go to my parents, Frans and Sietske, for enabling and motivating me to pursue my academic and non-academic interests, and to Eline, whose unwavering support has been a constant factor for the past two years. As this journey concludes, a new one awaits, and I am excited to see what the future holds.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Insurance plays a crucial role in today's society. Against payment of a premium, an insurer agrees to make a monetary provision on behalf of an insured party to cover the loss of an insurable interest due to one or more future, well-defined, but uncertain events [1], relieving the insured party of risks related to insurmountable financial damage. However, the insurance sector is plagued by fraudulent behaviour, and the automated detection of insurance fraud is hampered by various complex challenges. In light of these issues, this report presents a study on the automated detection of insurance fraud by employing social network analytics-based fraud detection models.

In this chapter, Section 1.1 first presents statistics that substantiate the aforementioned claims, along with a summary of issues that complicate the detection of insurance fraud. Together, these form the motivation for conducting this research. Then, Section 1.2 introduces the goal, scope and methodology of this research, followed by a description of our hypotheses in Section 1.3. This research's significance is elucidated in Section 1.4. Following that, Section 1.5 concludes this chapter with an overview of the structure of this report, providing readers with an outline of forthcoming chapters and their respective contributions.

1.1 Context and Motivation

In recent years, insurance fraud has remained a noteworthy issue. In 2021, Dutch insurers reported almost 13,000 cases of verifiable insurance fraud to the *Centrum Bestrijding Verzekeringscriminaliteit*[1] [2], a constituent body of the *Verbond van Verzekeraars*[2] representing over 95% of all non-life and life insurers in The Netherlands [3]. The reported number of fraudulent cases for 2021[3] is consistent with previous years [5, 6, 7], with the exception of a surge in 2019, when 23,376 cases were reported [8]. The cases account for an estimated yearly total of €80 million in insurance fraud in The Netherlands [2, 5, 6, 7, 8], contributing to a reported €2.5 billion worth of detected fraudulent claims across Europe in 2017 alone [9].

The numbers seem to suggest that current approaches to fraud detection are effective, but the European insurance and reinsurance federation *Insurance Europe* [10] has estimated that only 20% of fraudulent claims are detected as such [9]. Their most recent estimate of €13 billion worth of detected and undetected fraudulent claims in 2017, compared to €2.5 billion worth of detected fraudulent claims alone, suggests many instances of insurance

[1] The Dutch Centre for Combating Insurance Crime

[2] The Dutch Association of Insurers

[3] As of this writing, more recent numbers are yet to be published [4].

fraud go undetected.[4] This indicates low performance on the 'recall' metric. In other words: a low proportion of real positive (i.e. real fraudulent) cases are correctly predicted positive (i.e. predicted fraudulent) [12].

Meanwhile, numbers only reported for the years 2013 until 2018 highlight that the annual number of fraud investigations performed by members of the *Verbond van Verzekeraars* in response to suspected fraud indicators was 2.5–3.5 times greater than the number of verified fraudulent cases [13]. Assuming that this trend has continued and disregarding true but unverifiable fraud, this suggests low performance on the 'precision' metric. In other words: a low proportion of predicted positive (i.e. predicted fraudulent) cases are correctly real positives (i.e. real fraudulent) [12].

Thus, while the detection of insurance fraud is crucial to retaining fair insurance premiums and sometimes critical to abide by laws and regulations [14, 15], the aforementioned statistics highlight that, at least in theory, there is room for improving upon current fraud detection practices. Relatedly, the automated detection of insurance fraud has been an active area of research (Section 3). Complexities in automatically detecting (insurance) fraud are highlighted by various authors [1, 16]. Van Vlasselaer et al. [16] highlight the *uncommon*, *time-evolving* and *imperceptibly concealed* nature of fraud, among others. Its uncommon nature is found in the relatively limited availability of confirmed fraudulent cases when compared to unknown or legitimate cases, providing data mining techniques with highly skewed class distributions. This challenge is also explicitly denoted in various studies [17, 18, 19]. Its property of evolving with time highlights the need to consider adaptive fraud detection systems; while its imperceptible concealment is found in the fact that fraudulent parties tend to share many of the same characteristics as legitimate parties. Viaene and Dedene [1] identified further challenges in detecting insurance fraud through automated means. Among others, they mentioned that fraud is not self-revealing: it goes unnoticed when it is not actively looked for.

A range of automated fraud detection models underlying current fraud detection practices exist. These can range from simple pre-defined business rules to more advanced data mining models [17, 20, 21]. An illustrative example of a simple pre-defined business rule for fraudulent claims detection is: "flag any claim made within one week from the beginning of the insurance contract." Examples of more advanced data mining models include artificial neural networks (ANNs) [22] and support vector machines (SVMs) [23].

A specific type of model whose relevance for fraud detection has been explored in various publications [16, 18, 19, 24] are graph-based models. Graphs are a natural way to visualise networks with many nodes and complex relations between them, while related graph theory, "the natural framework for the exact mathematical treatment of complex networks" [25], provides the foundation for complex detection algorithms. The choice for graph-based models is justified by the assumption that individuals collaborate to commit fraud [16, 18, 19, 24], thereby forming a network of fraudsters. Indeed, the *Association of Certified Fraud Examiners* distinguishes various types of collaborative automobile insurance fraud, including staged accidents, inflated damages and misrepresented vehicle repairs [26]. Óskarsdóttir et al. [19] and Van Vlasselaer et al. [16] further substantiate their choice for a graph-based approach by referring to the concept of *homophily*: the idea that closely related instances are likely to behave in the same way [27]. They report that in their data, fraudulent entities are indeed more connected to other fraudulent entities than non-fraudulent entities are, and vice versa.

[4] The same estimate of €13 billion worth of detected and undetected fraudulent claims is also mentioned in a later report [11], but with no mention of the worth of *detected* fraudulent claims.

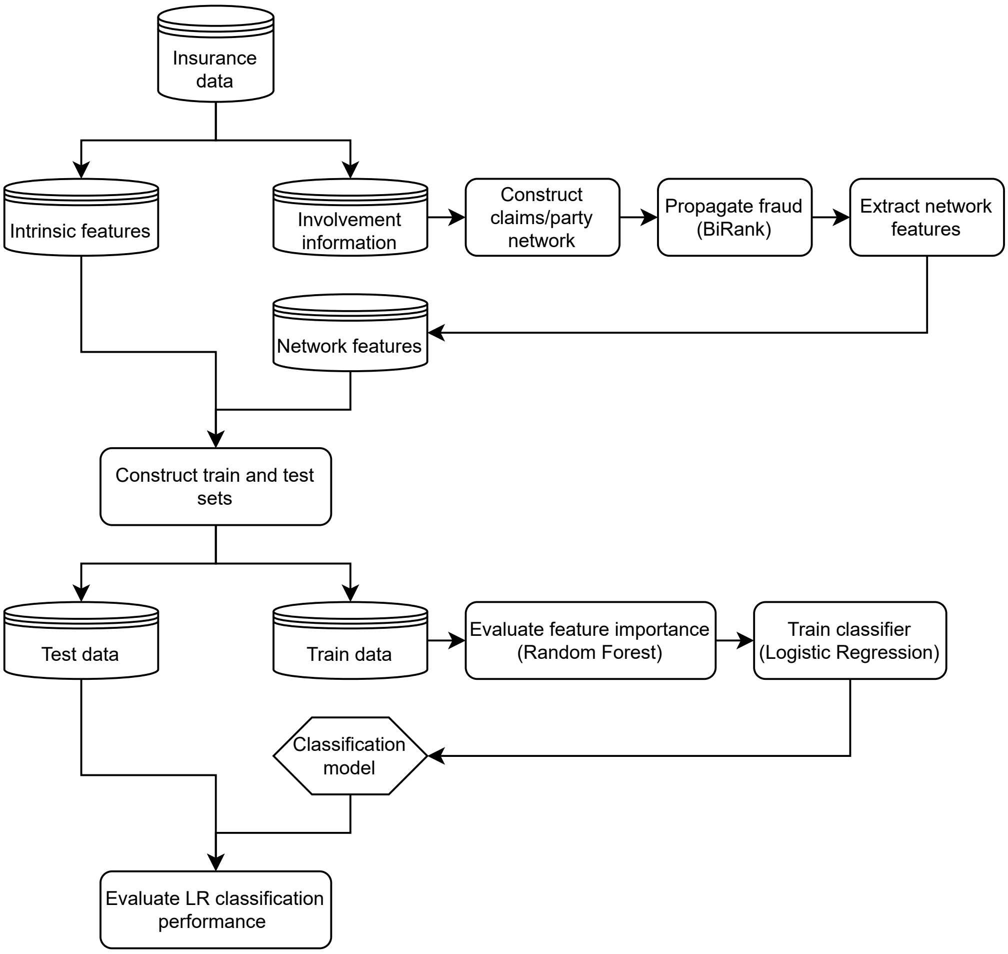

Insurance data
Intrinsic features
Involvement information
Construct claims/party network
Propagate fraud (BiRank)
Extract network features
Network features
Construct train and test sets
Test data
Train data
Evaluate feature importance (Random Forest)
Train classifier (Logistic Regression)
Classification model
Evaluate LR classification performance

Figure 1.1: Simplified illustration of the fraud detection model proposed in [19]

1.2 Goal, Scope, and Methodology

In the proposed study, we extend upon the work conducted by Óskarsdóttir et al. [19]. They establish fraud as a social phenomenon and devise a social network-based approach for fraud detection. A simplified visual representation of the approach is displayed in Figure 1.1.

The presented approach involves first constructing a bipartite network [28] of claims and associated actors. Then, the BiRank algorithm [29] is utilised to propagate fraud through the network and calculate a fraud score for each claim, representing a claim's exposure to known fraudulent claims. Next, features related to fraud scores and the neighbourhood structure of claims are extracted from the network. Finally, the fraud score- and neighbourhood-related features are combined with intrinsic claim features and used to construct a logistic regression model with fraud as the target variable.

The authors show that, when applied to a vast data set of millions of claims, the models incorporating network-derived features demonstrate superior performance compared to models relying solely on intrinsic claim features, measured in terms of area under the receiver operating characteristic curve (Section 4.6.1), area under the precision–recall curve (Section 4.6.2), and top decile lift (Section 4.6.3). Combining network and claim-specific features further enhances this performance.

In the proposed study, we take inspiration from this original work and their suggestions for future research and answer the following research questions:

RQ1 To what extent can empirical evidence for the homophily assumption presented in Óskarsdóttir et al. [19] be found in a different real insurance data set?

RQ2 How do the results presented in Óskarsdóttir et al. [19] generalise to a different real insurance data set?

RQ3 How can the social network analytics-based insurance fraud detection approach presented in Óskarsdóttir et al. [19] be adapted to enhance its classification performance?

  a What is the impact of extending the approach with time-weighted fraud influence and edges?

  b What is the impact of extending the bipartite network with party–party relations based on shared resources?

RQ4 How do the baseline and adapted models along with different combinations of feature sets compare in terms of highlighting interesting and/or suspicious claims that had not been investigated previously?

Answers to these research questions provide a deeper understanding of graph-based insurance fraud detection techniques and provide practical insights that could ultimately save insurers and their customers from the consequences of fraud.

To conduct this research, we first collect unprocessed insurance data and transform it into the types of data specified in the original study. Subsequently, we analyse this data for evidence related to the homophily assumption. Then, we reconstruct the model proposed by the original authors based on their reported methodology and experimental setup and evaluate its performance on our own data, enabling a comparison of results between the two studies and therewith facilitating an evaluation of the generalisability of their results. Finally, two adapted models are constructed based on the existing baseline model and evaluated on the same data. Their performance is then compared to our baseline model results, yielding insight into the value of our proposed adaptations. Subsequently, each models' claims with the highest predicted probability of fraud are analysed by fraud experts for suspicious characteristics, yielding the required information for assessing models' capabilities in recalling previously uninvestigated claims.

1.3 Hypotheses

Our current hypothesis regarding RQ1 is that empirical evidence supporting the homophily assumption is also present in our data set. In part, this is because the homophily assumption might already be either implicitly or explicitly embedded in existing business rules and/or fraud detection models. Further motivation results from the existence of various types of collaborative automobile insurance fraud [26], as well as the explicit distinction between *opportunistic* and *organised* or *gang* fraud reported in various studies, as presented in Section 2.1.

For RQ2, we hypothesise that the main findings in Óskarsdóttir et al. [19] generalise to the data set used in this study. More specifically, we expect that the inclusion of neighbourhood and score features enhances the performance of the supervised machine learning classifier, but assume that the importance ranking of individual features will differ at least slightly. Our expectation regarding the inclusion of score and neighbourhood is directly related to our hypothesis concerning the homophily assumption. Meanwhile, we expect difference in more granular results due to potential differences in implementation and due to the idea that insurers' data will not match one-on-one, in part due to differences in existing fraud detection practices that affect the results.

For RQ3, we hypothesise that both adaptations yield a positive impact on the classification performance of the model. Considering the inclusion of time-weighting (i.e., RQ3a), this hypothesis is consistent with the time-evolving nature of fraud suggested in existing studies [16, 30]. Additionally, it aligns with the positive reported impact of the inclusion of time weighting in a study on fraud detection in the social security domain [16]. Regarding the hypothesis for RQ3b instead, we propose that shared resources, such as shared bank account numbers and shared email addresses, suggest a close relation between the two parties sharing the resource. Following the notion of homophily, these parties are thus more likely to share behavioural characteristics, including the potential perpetration of fraud.

For RQ4, we adopt a similar hypothesis to the one for RQ3 and expect that either one of the two adaptations yields enhanced results in comparison to the baseline model. This hypothesis is based on the same arguments as for RQ4, whose duplication we omit.

1.4 Significance

Section 1.1 provided numbers regarding insurance fraud in The Netherlands and across Europe. It showed that insurance fraud remains a huge problem and suggested that there is room for improving fraud detection in terms of both precision and recall. Doing so through optimisation of fraud detection models underlying current fraud detection mechanisms could prove beneficial, both for insurance companies and their customers, as well as citizens in general. This claim is substantiated as follows.

First, note that fraud detection models are employed in the *fraud detection* phase of the fraud detection and investigation process (see Section 2.2). They decide whether a claim is forwarded to the *fraud investigation* and *fraud confirmation* phases for human investigation. Improvements to the precision of fraud detection models means fewer benign claims unnecessarily enter these subsequent labour-intensive phases, which could lead to a decrease in labour costs and required manpower. The result is a positive impact on the financial position of the insurance company, which could ultimately decrease insurance premiums for their customers.

Second, consider that any undetected fraudulent claims result in unjustified payouts from insurer to fraudster. Improving recall of fraud detection models and therewith the proportion of fraudulent claims that are detected reduces these payouts, providing two main benefits. For one, it would decrease insurance expenditure, again positively impacting the financial position of the insurance company. Simultaneously, it prevents money from flowing to criminals, which emphasises the notion that "crime doesn't pay" and is beneficial to society at large. Especially the explicit consideration of networks of claims and stakeholders, as is done in this study, shall enhance the capabilities in detecting the organised or gang type of fraud (Section 2.1). This might yield more substantial savings than detection of opportunistic types of fraud would.

The proposed study contributes to the optimisation of fraud detection models by exploring the efficacy of various adaptations to the social network analytics-based fraud detection approach presented in [19] and the generalisability of their reported results. It shall highlight opportunities for future research, and provides practitioners with inspiration in terms of graph-based fraud detection models to deploy in practice.

1.5 Report Structure

The remainder of this report is structured as follows. In Chapter 2, background information is provided to enhance comprehension of the research topic. Chapter 3 offers a review of recent literature on automobile insurance fraud detection, derived from a systematic literature review conducted prior to this study. Chapter 4 outlines the methodology established to address the research questions. Chapter 5 presents details regarding the implementation of this methodology and the utilised experimental setup. Chapter 6 provides an objective presentation of the results obtained from our analysis. Chapter 7 presents an interpretation of these results in relation to the research questions, along with this study's limitations and suggestions for future research. Finally, Chapter 8 summarises the main elements and key findings, concluding this report.

Chapter 2

Background

To enhance the comprehension of the concept of 'insurance fraud' and provide an understanding of both a typical multi-step approach that is employed for its detection, as well as methods used in this work, this chapter presents background information covering these topics. The concept of 'insurance fraud' is further explained in Section 2.1, whereas Section 2.2 provides information about a typical fraud detection process. Section 2.3 then provides background information about the Random Forests algorithm, one of the main machine learning algorithms employed in this study, while Section 2.4 sheds light on the concept of 'resampling'.

2.1 Insurance Fraud

According to Benedek, Ciumas and Nagy [31], there is no generally accepted definition to automobile insurance fraud. This statement is in line with the statement that "there is disagreement within the industry as to the best working definition of insurance fraud" [32, p. 5]. Morley, Ball and Ormerod [32] do report one possible definition of insurance fraud, originally reported in a different publication [33]: "knowingly making a fictitious claim, inflating a claim or adding extra items to a claim, or being in any way dishonest with the intention of gaining more than legitimate entitlement", but this definition covers insurance *claims* fraud only. It does not cover the second category of fraud distinguished by Viaene and Dedene [1]: insurance *underwriting* fraud.

In this study, insurance *claims* fraud is defined according to the definition of insurance fraud reported in the previous paragraph. This comprises the main topic of this study and is therefore simply denoted as 'insurance fraud' in subsequent sections. Meanwhile, insurance *underwriting* fraud is defined as "the dissimulation of information during application (application fraud) to obtain coverage or a lower premium (premium fraud), the deliberate concealment of existing insurance contracts covering the same property and casualty (P&C) risk, and underwriting coverage for fictitious risks." [1, p. 315].

In several studies, insurance fraud is further categorised based on two extremes: *opportunistic* fraud, and *organised* or *gang* fraud [26, 32, 34, 35]. The former involves parties simply seizing an opportunity to gain unlawful benefits by, for example, inflating the damages of otherwise genuine claim, whereas the latter concerns carefully planned scams involving multiple parties to recurrently deceive insurers for the parties' own benefit. A similar type of distinction between these two types of fraud is made by Viaene and Dedene [1], albeit using the term *soft* fraud for opportunistic fraud, and *hard* fraud for organised fraud.

2.2 Insurance Fraud Detection and Investigation Process

The process of fraud detection typically involves a multi-step approach. For instance, Van Vlasselaer et al. [16] distinguish the stages of *fraud detection*, *fraud investigation* and *fraud confirmation* in their study on social security fraud detection. Meanwhile, Viaene and Dedene [1] combine the first two stages and merely distinguish the *screening* and *investigation* phase. On the other hand, Bolton and Hand [36] do not explicitly label separate stages, but they do emphasise that alerts generated by statistical fraud detection methods should be followed by manual investigation.

In this study, we consider a process similar to the one presented by Van Vlasselaer et al. [16]. The initial phase, *fraud detection*, focuses on identifying and flagging high-risk claims. This task may be performed manually by claim handlers, but there is an increasing reliance on automated fraud detection models to support this process—one of the main topics of this research. Claims flagged in the fraud detection phase are forwarded to the *fraud investigation* stage. In this phase, experts with domain knowledge and insights conduct an initial, manual claim assessment. Claims that are indeed deemed suspicious based on this assessment are forwarded to the fraud confirmation phase for further assessment, while others are returned to the regular claim handling process. In the *fraud confirmation* phase, fraud investigators investigate the claim more thoroughly to confirm whether the suspicions of fraud are substantiated. If there is sufficient evidence to support the fraud allegations, the claim is denied and the fraud is registered. The claim can then be made available to internal fraud detection models as an example of a fraudulent case. Conversely, if the investigation does not yield enough evidence of fraud, the claim is returned to the regular claim handling process.

Figure 2.1 presents a visual illustration of this fraud detection process, constructed using the Business Process Model and Notation (BPMN) visual language [37].

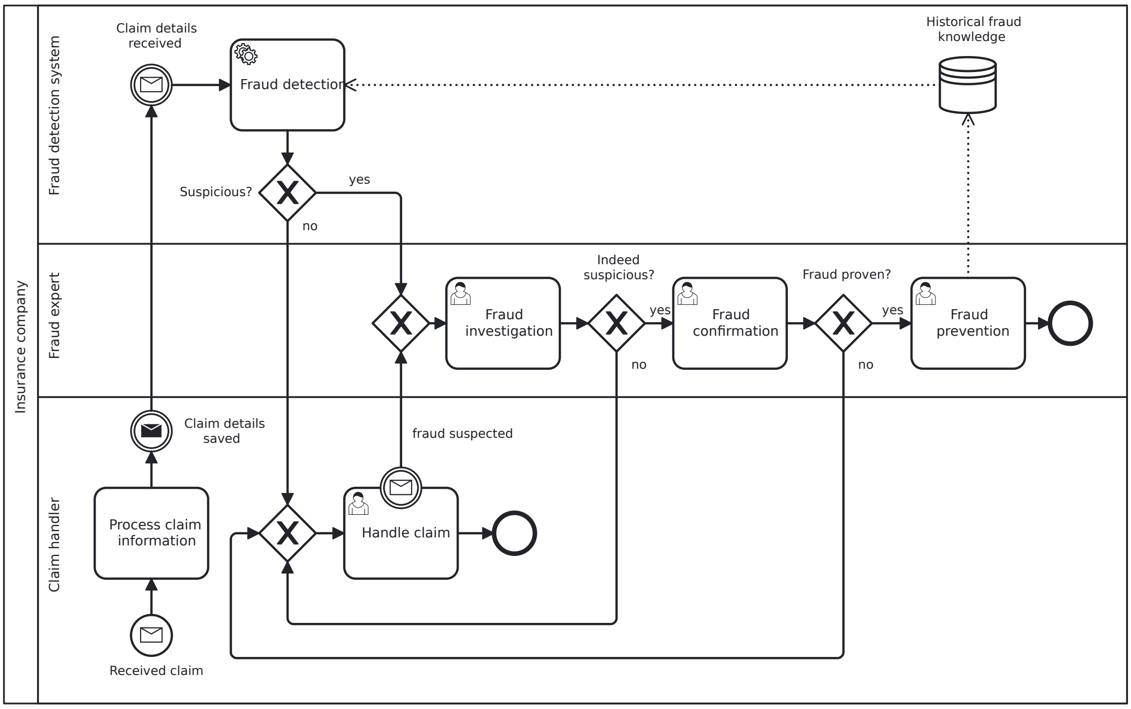


Figure 2.1: Fraud detection and investigation process

2.3 Random Forests

Random forests is an ensemble learning method that was originally proposed by Breiman [38]. It operates by constructing a multitude of decision trees and its output is an aggregate derived from the results of the individual trees in the forest. Random forest models can be constructed for both regression and classification tasks. In this study, only the classification aspect is emphasised.

2.3.1 Approach

A key technique employed in random forest models is the bootstrap aggregating technique, commonly referred to as ‘bagging’. During training, random forests implement this technique by constructing each individual decision tree using only a subset of the complete training data set. This subset, termed the bootstrap set, is generated by selecting a random sample with replacement from the full training set T, composed of $X_{\text{train}} = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}\}$ with corresponding labels $Y_{\text{train}} = \{y_1, \ldots, y_n\}$. For a given tree $k \in \{1, \ldots, n\}$ in a random forest with n trees, the corresponding bootstrap set is denoted as T_k.

What distinguishes random forests from a regular bag of trees is that random forests also employ the random subspace method [39], sometimes referred to as attribute bagging [40]. This method involves randomly selecting only a small group of input variables to split on for each split in each individual tree. The size of this group, denoted as F, is fixed. The primary objective of employing the random subspace method is to reduce the correlation between generated trees, thereby enhancing the generalisation error of the forest.

Breiman [38] presents the classification in random forest models as a majority voting task: input is run through each tree in the forest and the class that is selected by most trees is provided as output. However, alternative approaches are also observed. An implementation of the algorithm in the commonly used Python library scikit-learn, for example, “combines classifiers by averaging their probabilistic prediction, instead of letting each classifier vote for a single class” [41].

2.3.2 Feature Importance Ranking

To use random forest models for evaluating the importance of individual features, multiple approaches can be taken. Here, we distinguish the permutation importance approach [38] and the mean decrease in impurity feature importance approach [42].

The mean decrease in impurity (MDI) feature importance approach focuses on the impurity during splits, which indicate the homogeneity of the labels in the split’s leaf nodes. It is based on the idea that variables which decrease the impurity during splits are important, such that

$$\text{Importance}(x_m) = \frac{1}{n_T} \sum_{i=1}^{n_T} \sum_{t \in T_i | v(t) = x_m} p_{T_i}(t) \Delta i_{T_i}(t), \tag{2.1}$$

where x_m indicates the variable under consideration, n_T denotes the number of trees in the forest, $t \in T_i$ represents node t in optimal subtree T_i, $v(t)$ is the variable used for the split in node t, $p_{T_i}(t) = \frac{n_t}{N}$ is the fraction of samples reaching node t, and $\Delta i_{T_i}(t)$ is the change in impurity at node j in tree T_i. This equation was adapted from Louppe et al. [43].

To rank the importance of features using the permutation importance approach, we first need to calculate the out-of-bag (OOB) error for each data point. This is calculated as follows. Each independent tree k in the random forest is trained using only a subset of all training data, denoted as the bootstrap set T_k. Now, let all data that *is* in T but *is not* in T_k, i.e., $T \setminus T_k$, represent the out-of-bag data for tree k. Then, for each combination $(\boldsymbol{x}, y)$ in the training data, we collect votes from a classifier k only if $(\boldsymbol{x}, y) \notin T_k$. The out-of-bag estimate for the generalisation error is then the error rate of the out-of-bag classifier on the training set.

Then, suppose we have M input variables. After constructing a tree in the forest, we randomly permute the values for variable m in the out-of-bag training data, run the permuted data down the newly constructed tree, and save the classification result. This is repeated for all variables $m \in \{1, \ldots, M\}$ and for all trees. Then, once all trees have been constructed, for all variables m, the predictions of the classifier are compared with the true class labels and the average in out-of-bag errors before and after the permutation is computed. This misclassification rate can then be compared to the performance of the classifier with all variables intact. The larger the decrease in performance resulting from the permutation of variable m, the larger the importance of the variable.

2.4 Resampling

As presented in Section 1.1, one of the complexities in automatically detecting automobile insurance fraud is the highly skewed class distributions that the corresponding machine learning algorithms need to deal with. One approach to addressing this problem is by employing resampling methods [44], which can be further categorised as either undersampling or oversampling methods. Undersampling involves reducing the number of samples of the majority class, whereas oversampling involves increasing the number of samples of the minority class.

Straightforward approaches to undersampling and oversampling are the random undersampling and random oversampling methods [45]. Random undersampling involves randomly removing samples from the majority class. In the context of insurance claims fraud detection, this would entail removing legitimate insurance claims or insurance claims that are not verified legitimate or fraudulent, until the desired class distribution is achieved. In contrast, random oversampling involves randomly duplicating samples of the minority class. In insurance claims fraud detection, this would constitute duplicating confirmed fraudulent claims, resulting in a more balanced data set that contains multiple replicas of the same fraudulent sample.

In addition to these basic approaches, more advanced approaches have been proposed for both undersampling and oversampling. An example of the former is the utilisation of Tomek links [46], which aims to remove noisy or borderline instances from the majority class. On the other hand, an example of an advanced oversampling method is the Synthetic Minority Oversampling Technique (SMOTE) [47]. SMOTE generates synthetic samples for the minority class by interpolating between existing instances, thereby expanding the representation of the minority class.

Resampling should generally be applied after splitting the data into train and test sets, and only on the train set, particularly when oversampling is employed. Applying resampling before the division can distort the real class distribution in the test data, limiting the generalisability of the results obtained on the test set [48]. Moreover, when employing oversampling, applying it before the data split can introduce data leakage compromising the validity of the results [48, 49].

2.5 Conclusion

Having established foundational information that might prove valuable in understanding the remainder of this report, the next chapter presents a review of recent literature that has also explored the subject of automobile insurance fraud detection. This review sheds light on recent advancements, providing insights into the current study's position within a broader context.

Chapter 3

Literature Review

In previous work, we conducted a systematic literature review (SLR) focusing on recent studies employing data mining-based approaches for detecting automobile insurance fraud. This investigation centred on examining data sets, detection methods, and resampling techniques considered in these studies, addressing specific research questions, namely:

- What automobile insurance data sets have been used in recent research on the topic of automobile insurance fraud detection?
- What is the current state-of-the-art in data mining methods for the automated detection of automobile insurance fraud?
- What is the current state-of-the-art in resampling methods for training automobile insurance fraud detection algorithms?

To address these questions, we systematically explored studies published between January 1st, 2019, and April 2nd, 2023, across four reputable scientific databases. The search yielded 50 relevant primary studies, which were subsequently analysed, categorised, and described. In this chapter, Sections 3.1 to 3.3 first present the key findings in relation to the research questions that formed the basis of the SLR, followed by a discussion on merely comparative studies in Section 3.4. Then, Section 3.5 delineates the primary contributions of the present study within the context of existing research in the same automobile insurance fraud detection field.

3.1 Data Sets

The exploration of automobile insurance fraud data sets used in recent research was based on the seemingly limited availability of publicly available automobile insurance fraud data to conduct research on. This idea was previously highlighted in an existing SLR on the same topic [31] and has since been corroborated in our more recent SLR. Our review uncovered six distinct (reportedly) publicly available data sets, of which only three have remained available online [50, 51, 52, 53]. Among these, the most commonly used was ‘carclaims.txt’, originally distributed as part of the Angoss KnowledgeSeeker software and nowadays available in various online repositories [52, 53].

Equivalent to the other two publicly available data sets that were uncovered, the ‘carclaims.txt’ data set provides no properties that facilitate an evaluation of fraud detection approaches that employ unstructured textual data or relational information, as in this study. This suggests further challenges that might hamper the development of novel techniques in the automobile insurance fraud detection research domain.

3.2 Detection Methods

In terms of reportedly novel detection methods, supervised detection methods were distinguished from their unsupervised counterparts. Supervised methods rely on labelled data to ‘learn’ how to perform a classification or prediction task. In the context of automobile insurance fraud detection, these labels are derived from historical knowledge of fraud. Unsupervised methods require no such labels. Correspondingly, their use was often motivated by the limited availability of this type of labelled data [35, 54, 55, 56].

The SLR uncovered that the supervised random forest, logistic regression, and k-nearest neighbours algorithms were among the most commonly considered classification algorithms in recent automobile insurance fraud detection research. Belonging to the group of ‘conventional machine learning methods’ [57], part of their prevalence was attributed to their frequent use in studies focused on a mere evaluation of existing methods, along with their use as part of larger detection models. More generally, it was observed that the use of supervised techniques was more prevalent than the use of unsupervised ones.

3.2.1 Novel Supervised Detection Methods

In terms of studies proposing reportedly novel supervised detection methods, our analysis revealed a predominant focus on enhancing supervised classifiers that rely on intrinsic features of claims and whose optimisation is based on metrics that do not explicitly incorporate the savings or costs that one would incur by (in)correctly classifying the respective claim. An analysis of these studies is provided in the main report covering the SLR, but we omit their description in this chapter. Instead, this section elucidate studies exploring alternative directions only.

For example, Dimri et al. [58, 59] studied the value of incorporating unstructured, textual data for three classification tasks, including insurance fraud detection. Their approach involved further pre-training existing large language models BERT [60] and ULMFiT [61] to construct an insurance-based large language model and yielded improved insurance fraud classification performance compared to the use of conventional text-based approaches. Additional findings included an observed trade-off between the timeliness and correctness of the classification task. The reported value of utilising textual data for fraud detection purposes was also established by Yankol-Schalck [62]. These authors established value in incorporating not only structured data available at the opening of a claim, but also unstructured textual data from the same moment and structured information later generated by the first adjusters’ report, thereby yielding a fraud score that evolves over the life of a claim.

A different approach was taken by Óskarsdóttir et al. [19] and Zhang et al. [63], who studied graph-based methods for insurance fraud detection and motivated their choice by referring to their potential for detecting, for example, fraud networks. Óskarsdóttir et al. [19] proposed using the BiRank [29] algorithm to compute fraud scores for claims in a bipartite network of claims and involved parties. Their findings revealed that the addition of fraud scores and related network features indeed yielded enhanced performance in subsequent supervised classification compared to using intrinsic claim- and policyholder-related features alone, presenting opportunities for future research. Meanwhile, Zhang et al. [64] proposed the use of knowledge graph [65] techniques for automobile insurance fraud detection. They demonstrated the value of knowledge graph embedding techniques for predicting missing accomplice relations and identifying fraud networks, while also reporting enhanced performance in classifying individual claims by incorporating features derived from these predicted accomplice relations.

In a different study, Zelenkov [66] proposed a technique that incorporated the example-dependent cost of misclassification into the optimisation process of the AdaBoost [67] machine learning algorithm. This facilitates optimisation based on cost savings rather than based on common metrics related to the number of correct predictions, which might sometimes be more representative of some insurers' goal in automobile insurance fraud detection.

Meanwhile, different papers presented approaches to insurance fraud detection that are at least partially founded in rules derived from expert systems that were traditionally used. Baumann [68] proposed the use of association rule mining to discover meaningful new rules that reflect dependencies between existing rules, yielding benefits especially in terms of ease of implementation. However, their study revealed existing limitations in terms of performance and methodology. Conversely, Liu et al. [69] proposed an approach based on evidential reasoning, which combines evidence from expert knowledge (i.e., fraud indicators) and probabilities of fraud obtained from historical data. They reported enhanced performance compared to other conventional machine learning algorithms and highlighted their method's retained usability and interpretability.

A different approach altogether was presented by Qazi et al. [70], where the use of a scaleable T-pattern algorithm [71] was proposed to detect patterns in temporal customer data derived from customer interactions and events. Their combined use of hand-crafted features and binary indicators derived from these patterns yielded slightly elevated performance compared to using one of the feature types alone, though statistical significance may be lacking.

A number of authors also established approaches that integrate detection and undersampling by utilising fuzzy C-means clusters (FCM) [72] for both classification and outlier elimination [73, 74, 75, 76]. These approaches first train supervised machine learning classifiers using a training set that is undersampled by removing outliers in the majority class based on FCM clusters. Then, new samples are classified using a two-stage approach. In the first stage, claims are classified as genuine, suspicious or fraudulent based on their Euclidean distance to the cluster centres previously defined during the training phase. In the second stage, claims formerly labelled as suspicious are fed into a trained supervised classifier, which performs a binary 'fraudulent/non-fraudulent' classification. A key difference between the methodologies adopted in the separate studies was the algorithm used to optimise the FCM cluster centres, which ranged from the salp swarm algorithm [73] and the modified whale optimisation algorithm [74] to a regular genetic algorithm [75, 76]. However, our evaluation of the methodologies and experimental configurations adopted in each of these studies reveals potential shortcomings in their robustness, suggesting caution in drawing conclusions from their findings.

3.2.2 Novel Unsupervised Detection Methods

In addition to supervised methods, a number of unsupervised detection methods was proposed. For instance, one approach suggests employing autoencoders (AEs) or variational autoencoders (VAEs) [77] to detect anomalies in claims and reveal properties that are most likely drivers of fraud [54]. The approach was based on the idea that (V)AEs learn a representation of the majority class (i.e., legitimate claims) such that anomalous samples with an aggregate reconstruction error exceeding a defined threshold are likely part of the minority class (i.e., fraudulent claims). In the absence of labels, the method exhibited commendable detection performance in the absence of labels. However, its performance in comparison to supervised counterparts remained comparatively limited.

In a different study, Golden et al. [78] pursued a similar goal using a technique called ‘assymetric PRIDIT’, based on the PRIDIT-framework [79]. Their focus was on identifying individuals engaged in problematic hidden social behaviour through ‘suspicion of target group membership’ derived from ordinal categorical features, attaining additional value through insights into the importance of each variable response in classifying a claim as fraudulent or non-fraudulent. The authors reported promising performance in the presence of a set of predictor variables. These findings were supported by classification performance results similar to those achieved by a supervised classifier.

Alternative approaches were also proposed. Krishna and Ravi [80] proposed an anomaly detection approach based on a combination of modified differential evolution [81] and sparse subspace detection [82]. Disregarding potential limitations in the robustness of their reported experimental setup, their approach yielded mixed results when compared to other methods. Meanwhile, Shaeiri and Kazemitabar [55] introduced a computationally efficient approach to the spectral ranking of anomalies (SRA) technique [83] that shall enhance the scalability and therewith practical feasibility of its application, relying on a combined use with supervised machine learning classifiers. Their approach yielded similar performance to the original SRA algorithm with considerably reduced execution times, but the authors proposed that nowadays’ access to computing clusters yields opportunities for the use of the regular SRA algorithm instead.

Similar to how supervised graph-based approaches were presented, unsupervised graph-based methods were also considered. Tumminello et al. [35] introduced a statistically validated network approach to automobile insurance fraud detection that employs bipartite networks of subjects and accidents or vehicles. They presented methods for excluding weak ties in the network, evaluated various methods to identify communities, and proposed alert metrics to detect suspicious structures. Empirical case studies using an Italian industry-wide data warehouse of insurance claims showed that the proposed approach effectively assigned high or medium levels of statistical anomaly to the majority of externally validated fraudulent cases. Moreover, the authors revealed that network-derived features could be used to discriminate fraudulent from random events. Wang et al. [84] adopted a different kind of graph-based approach and presented how pre-defined suspicious entity groups can be automatically extracted from knowledge graphs. Their article presented no empirical validation of the proposed approach, so the effectiveness of the approach in the insurance fraud detection process could not be established.

In contrast to the aforementioned unsupervised approaches for insurance *claims* fraud detection, Vandervorst, Verbeke and Verdonck [56] proposed a method for automatically detecting insurance *underwriting* fraud. Their method involved using a combination of validated data and conditional density estimates derived from historical data to evaluate data misrepresentation risk. In the presence of informative relations between validated and uncertain data, their approach yielded promising results, with proposed contributions pertaining to its adaptability to pricing policy changes and its support for multivariate self-reported data.

3.3 Resampling Methods

Regarding resampling techniques, an examination of studies included in the SLR revealed that oversampling was employed in seventeen studies, while undersampling methods were utilised in ten studies. Delving into specific methods, random undersampling emerged as the most frequently employed undersampling technique, while the most prevalent oversampling technique was SMOTE, followed by the random oversampling method. In addition to these techniques, it was noted that some studies also proposed reportedly novel resampling methods—at least in the automobile insurance fraud detection domain. These are reported hereafter.

In terms of reportedly novel resampling methods in the context of automobile insurance fraud detection, Itri et al. [85] acknowledged that higher volumes of SMOTE oversampling might introduce over-generalisation and subsequently worsen classification performance. To address this issue, they proposed to optimise the classification threshold by choosing the threshold that yields the highest G-mean ($\sqrt{\text{recall} \times \text{specificity}}$). The downside of their approach is the computational complexity associated with a repeated evaluation of resource-intensive classifiers for different oversampling thresholds. This is especially relevant when large data sets are considered.

In a different study [86], authors suggested undersampling with the adaptive synthetic sampling method (ADASYN) [87], which differs from SMOTE by generating more synthetic data for minority class samples that are harder to learn in comparison to minority class samples that are easier to learn. The authors report enhanced performance using ADASYN when compared to using SMOTE. However, they present insufficient details to verify the correctness of their adopted experimental setup.

Adopting a distinctly different strategy, Kate, Ravi and Gangwar [88] introduced a generative adversarial network-based (GAN) [89] oversampling method called chaoticGAN, alongside a recommendation for undersampling using a one-class support vector machine [90]. Unlike conventional approaches, GANs can utilise learned distributional properties to generate synthetic samples. The authors demonstrate the superior performance of their setup compared to previous studies and all other evaluated combinations, except for performance statistically similar to that of a more computationally intensive GAN-baed approach. However, a limitation of GAN-based approaches in general lies in its deployability in production, constrained by the necessity for hyperparameter tweaking and distributed training.

3.4 Comparative Studies

In contrast to studies proposing reportedly novel methods for automobile insurance fraud detection, a large number of studies merely evaluated or compared the performance of existing detection and/or resampling methods. However, a substantial part of these studies show major limitations. In Sections 3.4.1 to 3.4.3, we distinguish the comparative studies based on their evaluation of either detection methods, resampling methods and report on their limitations and main findings.

3.4.1 Comparisons of Detection Methods

Considering studies that merely evaluated existing fraud detection methods, our analysis has suggested many are significantly constrained in their robustness. For example, while Reddy et al. [91] evaluated the classification performance of various supervised machine learning algorithms, their assessment was based on a mere 200 claims, and their reporting presented major constraints for conducting a thorough review of the validity of their results. Similar issues of unclear and even inconsistent reporting were also found in a study that merely evaluated the performance of a random forest classifier [92]. The clarity of reporting was also limited in an alternative comparison of classifiers [93], but their work presented an additional limitation in that it suggested overfitting on the test set, raising concerns about the reliability and generalisability of the reported findings. These types of indicators for overfitting were also found in the methodology of a study that compared a multi-layer perceptron, decision tree, and random forest classifier [94]. In a separate comparison of various supervised machine learning algorithms, a clear specification of the experimental setup was omitted altogether [95]. Meanwhile, reporting in the only study that considered insurance *sales* fraud suggested evaluation was conducted on the same data that was used for training [96], yielding results that are of little use.

Our literature review also uncovered comparative studies with seemingly fewer limitations in the robustness of their methodology. In their study on the use of robotic process automation for insurance fraud detection, S.Patil et al. [97] compared the performance of five supervised machine learning classifiers and reported the best performance metrics for random forest classifiers, followed by classification based on linear discriminant analysis. Meanwhile, in a separate study on an insurance-related blockchain platform [98], authors compared the performance of k-nearest neighbours [99], random cut forest [100], logistic regression and XGBoost [101] classifiers and reported the best performance for XGBoost across all considered performance metrics. Examining the logistic regression, support vector machine, and naive Bayes algorithms instead, Aslam et al. [102] reported that no single classifier outperformed the others across all metrics. Additionally, Itri et al. [103] compared ten supervised classifiers and suggest that random forests yielded the most desirable performance characteristics. Lastly, Piesio, Ganzha and Paprzycki [104] compared clustering algorithms and the supervised XGBoost classifier, highlighting XGBoost's superior performance based on the area under the precision–recall curve and a visual inspection of the results generated using dimensionality reduction. However, they provide limited details regarding the exact validation method used.

3.4.2 Comparisons of Resampling Methods

A single study focused on an evaluation of resampling methods only, assessing the impact of adopting SMOTE on insurance fraud classification performance and revealing enhanced importance on the balanced data set [86]. However, an extensive description of their applied experimental setup is missing. This diminishes the impact of their work, especially since the reportedly perfect performance across all their considered performance metrics seem implausible if a valid experimental setup was applied.

3.4.3 Comparisons of Detection and Resampling Methods

More studies were found that compared both resampling and detection methods instead. For example, Hanafy and Ming [105] evaluated a wide range of classifiers and four distinct resampling methods, namely random undersampling, random oversampling, SMOTE, and SMOTE+EditedNearestNeighbours [106]. They concluded that, while no one resampling method consistently produced the best classification results, the use of resampling methods did enhance the performance of the considered classifiers which, based on the reported performance metrics, requires some nuance.

A similar type of research was conducted by Salmi and Atif [107], who considered the logistic regression and random forest classifiers, the SMOTE and ROSE [108] oversampling techniques, and two feature sets of different sizes. In terms of classifiers and resampling methods, they reported that RF classifiers consistently outperformed LR classifiers, while no discernible difference was found between the use of SMOTE and ROSE.

A more comprehensive evaluation of a wide number of resampling and classification methods was conducted by Soufiane et al. [109]. However, their pipeline visualisation suggests feature selection and oversampling was applied before splitting the data into train and test sets, which has been previously described to induce data leakage from the train to the test set and to present overoptimistic results (Section 2.4). These represent critical limitations that pose a major threat to the validity of their findings and thus the contribution of their results.

3.5 Contributions

Having explored recent developments in the field of data mining-based automobile insurance fraud detection, this study presents the following contributions to the current state-of-the-art:

- We further explore the use of graph-based automobile insurance fraud detection, a type of approach that has been considered promising in recent studies.
- We consider a recent work on graph-based automobile insurance fraud detection and:
  - evaluate the generalisability of their findings, facilitating an improved evaluation of their reported results;
  - reveal limitations in their methodology and experimental setup and present alternative approaches to address these limitations;
  - strengthen their suggestion that homophily is present in networks of claims and involved parties by conducting a more robust analysis of homophily in our data set;
  - evaluate the impact of time-weighted fraud influence, which was suggested for future research;
  - evaluate the impact of extending their model by considering shared resources.

In the following chapter, we present the methodology that is paramount to providing this contribution.

Chapter 4

Methodology

Following the exploration of recent studies on the topic of automobile insurance fraud detection in Chapter 3, this chapter presents the methodology for the current study. A visual outline of this methodology is presented in Figure 4.1.

In line with our goal of extending the work in Óskarsdóttir et al. [19], we first construct a data set that closely mirrors the one used in their study in terms of features and scope, as delineated in Section 4.1. Next, we evaluate whether this data set suggests any evidence for the homophily assumption following the methodology in Section 4.2. Then, we adopt the methodology in Section 4.3 to re-implement the social network analysis-based fraud detection model, and compare the performance of our implementation of the model on our data set with the results obtained in the existing study. This comparison enables us to assert whether the findings of the original authors generalise to our implementation and data set.

The results obtained by *our* implementation on *our* data serve as the baseline for future comparisons. This approach ensures that any observed differences in performance between the baseline model and adapted models are a direct result of the adaptations themselves, rather than stemming from differences in implementation or data characteristics. The first adaptation involves incorporating time-weighted fraud influence in accordance with the methodology outlined in Section 4.4. The second adaptation involves introducing the concept of shared resources based on the methodology presented in 4.5. The metrics and approaches used for model evaluation are presented in Section 4.6.

4.1 Data Set

This research is conducted using a confidential data set sourced from an insurer that is specialised in property and casualty (P&C) insurance types, including automobile insurance, homeowners insurance and liability insurance. More specifically, we consider intrinsic features of claims and policyholders, as well as relations between claims and involved parties. The data set was constructed to closely align with the data set used in the original study by Óskarsdóttir et al. [19], but minor deviations were inevitable. These deviations stemmed from technical challenges, as well as disparities in the data model that is utilised.

The largest difference pertains to the number of years covered by our data set. In contrast to the data set used in the original work, which encompasses automobile insurance claims filed by policyholders over a period of six years, we adopt data on claims filed over a four-year time frame, since incorporating data from a fifth and sixth year proved infeasible due to past system migrations. An additional difference is found in our consideration of closed claims only, whereas the data set in the original work comprised both closed and

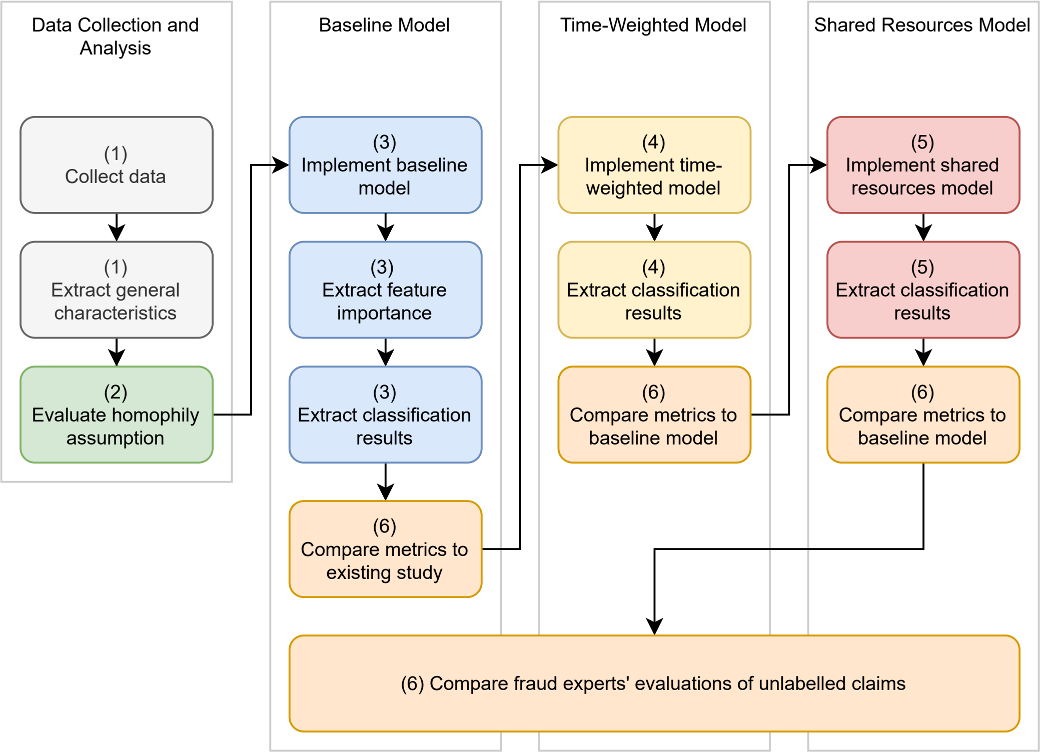

Data Collection and Analysis
Baseline Model
Time-Weighted Model
Shared Resources Model
(1) Collect data
(1) Extract general characteristics
(2) Evaluate homophily assumption
(3) Implement baseline model
(3) Extract feature importance
(3) Extract classification results
(6) Compare metrics to existing study
(4) Implement time-weighted model
(4) Extract classification results
(6) Compare metrics to baseline model
(5) Implement shared resources model
(5) Extract classification results
(6) Compare metrics to baseline model
(6) Compare fraud experts' evaluations of unlabelled claims

FIGURE 4.1: Simplified visualisation of the methodology, with numbers denoting relations to sections

open claims. Smaller differences are found in the intrinsic features that are used and the types of relationships that are considered.

Section 4.1.1 describes the intrinsic features available in our data set and presents a comparison to the intrinsic features in [19]. Section 4.1.2 presents a comparison between the types of relationships considered in our data set and the ones considered in [19]. Section 5.2 offers insight into various characteristics of the data set.

4.1.1 Intrinsic Features

Claims in the data set are characterised by 20 distinct intrinsic properties, presented in Table 4.1. These are subdivided into the categories ‘target’, ‘policyholder characteristics’, and ‘claim characteristics’. The ‘target’ category includes the *fraud* property, i.e., the main property of interest which indicates whether the claim has been confirmed fraudulent, confirmed legitimate, or not investigated. The ‘claim characteristics’ include properties that are specific to an individual claim. The ‘policyholder characteristics’ include properties pertaining to age, responsibility, and historical claims behaviour of the policyholder whose insurance policy is (or would be) claimed on. Note that, although we *do* include the `age` property to achieve high resemblance to the model presented in Óskarsdóttir et al. [19], we posit that the age of the policyholder should be excluded from any operational version of the model. Instead, our recommendation is to concentrate solely on properties that can be influenced by the respective parties, aligning with the stance of the insurance company granting data access. This approach, focusing solely on influenceable properties, minimises the potential for the model to exhibit unfair discrimination based on immutable characteristics and mitigates the risk of reinforcing undesired biases that might be present in the existing data.

Table 4.1: Intrinsic features in the constructed data set

Type	Feature	Description
Target	`fraud`	Is the claim fraudulent: yes, no, or unknown.
Policyholder characteristics	`age`	Age of policyholder when incident occurred.
	`responsibilityCode`	Policyholder's responsibility in the incident: at fault, shared responsibility, full right, or unknown.
	`numContracts`	Number of contracts the policyholder has or has had with the insurer.
	`claimAge`	Number of months from beginning of contract to the date the incident occurred.
	`nClaims1`	Number of claims across all lines of business in last year before the incident occurred.
	`nClaims5`	Number of claims across all lines of business in last five years before the incident occurred.
	`lastClaim`	Number of months since last claim occurrence.
	`amount1`	Claimed amount across all lines of business in last year before current claim occurrence.
	`amount5`	Claimed amount across all lines of business in last five years before current claim occurrence.
	`refused1`	Number of times compensation was refused in the last year before current claim occurrence.
	`refused5`	Number of times compensation was refused in last five years before current claim occurrence.
	`atFault1`	Number of times the policyholder had responsibility code 'at fault' in last year before current claim occurrence.
	`atFault5`	Number of times the policyholder had responsibility code 'at fault' in last five years before current claim occurrence.
	`sameSits1`	Number of times the policyholder has had the same responsibility code as the current claim in last year before current claim occurrence.
	`sameSits5`	Number of times the policyholder has had the same responsibility code as the current claim in last five years before current claim occurrence.
Claim characteristics	`people`	Number of people involved in the claim
	`organisation`	Number of organisations involved in the claim.
	`daysReport`	Number of days from the occurrence of the incident to the filing of the claim.
	`amount`	The claimed amount for closed claims or the expected claimed amount if the claim is still open.

When compared to the intrinsic features presented by Óskarsdóttir et al. [19], the features shown in Table 4.1 show three differences, excluding the removal of one instance of the duplicate `claimAge` feature in the original work. These follow from differences in properties and corresponding categorisations available in the source data. Firstly, the `responsibilityCode` feature is provided a fourth possible value: 'unknown'. Secondly, the distinction between 'people' and 'companies' is replaced with a distinction between 'people' and 'organisations', with a corresponding adjustment of the claim characteristics features to `people` and `organisations`. Lastly, the `police` feature is omitted altogether, since this data was not readily available.

Not all claims have complete intrinsic property information. For instance, when a submitted claim falls outside the coverage defined in a policyholder's contracts with the insurance company, it cannot always be allocated to any of the insurance contracts because it is not covered by any of them. For such claims, the `claimAge` property might be unavailable. Similarly, claims might not always proceed to a stage where details surrounding the claimed amount become relevant, which affects the availability of the `amount` feature. We describe our approach to dealing with this in Section 5.4.4.

4.1.2 Claim-to-Party Relations

Along with the intrinsic features, the data set also incorporates references to parties involved in a claim. However, in contrast with the intrinsic features, involvement information is included for all P&C claims, as opposed to automobile insurance claims only. This enables us to construct a more complete bipartite network, as described in Section 4.3.

Different from Óskarsdóttir et al. [19], we consider relations to policyholders, claimants, witnesses, injured, garages, legal counsel, agents, representatives, and experts, wherein agents serve as intermediaries facilitating transactions between policyholders and insurance companies, representatives act on behalf of involved parties, and experts contribute specialised knowledge related to, for example, vehicle damages. Thus, we step away from adopting the very broad definition of the term 'policyholder' used in the original study and: make an explicit distinction between policyholders, claimants, witnesses, and injured; add relations to agents, representatives, and legal counsel; and omit relations to brokers due to an unavailability of such data. The considered types of people and organisations are together referred to as 'parties'.

In addition, while authors of the original study consider distinct types of *parties* via node attributes, we consider distinct types of *involvements* via edge attributes. Accordingly, whereas parties in the original work assume a fixed role across claims, a single party in our data set can, for example, be involved as a witness in one claim and as a claimant in another. The party can even assume multiple distinct roles within a single claim, as frequently seen via the combination of a 'claimant' and 'policyholder' relation. The change should have no impact on the performance of the detection models, since neither party types of this kind nor involvement types are explicitly considered in the classification models. It does, however, disallow us from reporting on the exact same data characteristics, as described in Section 5.2.

Note that while the raw data used to construct the data set for this study includes relations to entities representing various departments within the insurance company, these relationships are intentionally excluded from the final data set. This decision is consistent with our emphasis on identifying fraud perpetrated by external parties specifically.

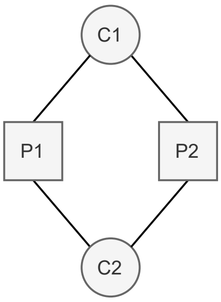


FIGURE 4.2: Visual representation of a four-cycle

4.1.3 Data Set Characteristics

After constructing the data set, we gather summary statistics to gain insights into its characteristics. These statistics are instrumental in facilitating comparisons between our study's data set and the one utilised by Óskarsdóttir et al. [19]. Following differences in the structure of the available data, we cannot always report on the exact same characteristics described by the original authors, but the reported characteristics may nevertheless shed light on potential factors contributing to divergent results obtained by the detection models.

To enable for a comparison of general network-related characteristics of our data and the data employed in the original work, we first focus on comparing statistics that describe the degree of claims in the two data sets. This concerns involvement information and therefore includes claims across all P&C lines of insurance at the insurance company, in line with Section 4.1.2. The effect of including non-automobile insurance claims is that the reported characteristics will reveal to be lower than for the automobile insurance claims alone. This follows from our focus on only general relation types (agent, claimant, legal counsel, policyholder, representative, and witness) and automobile-specific relation types (garage), thereby excluding relation types specific to e.g. travel insurance.

Next, we report on the existence of cycles in our data set, similar to the original authors. These structures, which are also referred to as 'simple cycles', are composed of distinct nodes connected by edges, creating a closed loop where the last node is linked to the first node [110]. More formally, consider a graph $G = (V, E)$ of nodes (vertices) V and edges E. Let $e_1, \ldots, e_n$ be a trail with node sequence $v_1, \ldots, v_n, v_1$. Then, if the only repeated nodes on the trail are v_1 (indicating the start and end), the subgraph G' induced by the set of edges $\{e_1, \ldots, e_2\}$ is referred to as a cycle of G.

In the context of insurance fraud, cycles can signify the existence of collaborations among parties. For example, consider cycles comprising four nodes, commonly known as four-cycles. Stemming from our bipartite network structure, these consist of two claims and two parties and reveal that two parties were involved in two of the same claims. If both claims are subsequently revealed to be fraudulent, this may serve as an indication of organised fraud efforts, as described in Section 2.1.

Figure 4.2 provides an illustration of a four-cycle. In this figure, claims are presented as circles and prefixed with 'C', whereas parties are presented as squares and prefixed with 'P'. Thus, the figure shows that parties 'P1' and 'P2' are both involved in claims 'C1' and 'C2'.

4.2 Evaluation of Homophily Assumption

Following the analysis of explicit structures in our data, we examine whether there is empirical support for the homophily assumption. Hereto, we compute the relative frequency of fraudulent and non-fraudulent claims in the second- and fourth-order neighbourhoods of claims with a known label, in line with the methodology outlined by Óskarsdóttir et al. [19]. In contrast to their approach, we then employ T-tests to compare the mean relative frequencies. This allows us to ascertain the statistical significance of any observed differences.

Diverging from the original work, our reporting not only encompasses the relative frequency of fraudulent and non-fraudulent claims among *all* claims in the second- and fourth-order neighbourhoods of labelled claims, but also considers their relative frequency among all *investigated* claims. By conducting this evaluation, we aim to discern whether the observed differences can be exclusively attributed to an increased number of investigations into claims surrounding known fraudulent claims, or if they indeed actually establish some empirical evidence for homophily.

4.3 Implementing the Baseline Model

Following the collection, processing, and analysis of data in Section 4.1, we re-implement the model proposed in Óskarsdóttir et al. [19]. This model lays the groundwork for later adaptations and serves as the baseline to which adapted versions of the model will be compared. The implementation involves the construction and analysis of a bipartite network, the computation of fraud scores using BiRank, the extraction of network features, and the implementation of a supervised learning model.

4.3.1 Constructing the Bipartite Graph

First, we construct a bipartite network consisting of two types of nodes: claims and parties. These types of nodes are connected via edges (i.e., relations), which represent the involvement of a party in a claim. Figure 4.3 presents a visualisation of this idea. Red, green, and white circles represent claims that are labelled fraudulent (C2 and C3), non-fraudulent (C1 and C5), and unknown (C4), respectively. Meanwhile, squares denote parties, whereas edges show a party's involvement in a claim. Edge labels denotes the role of a party in a claim, i.e., the type of involvement. Since we allow at most one edge between two distinct nodes, edge labels may be composed of multiple relationship types.

Contrary to the full fraud detection model, the bipartite network is not restricted to automobile insurance claims alone, but includes claims from all P&C lines of business of the insurance company. This is in line with the methodology adopted in the original study and shall provide a holistic view on the riskiness of clients. To eliminate data inconsistency issues, we merge distinct parties in the graph whenever both their names and postal codes match. This approach ensures that parties with duplicate profiles are properly depicted as a single party in the network, notwithstanding the small likelihood that it introduces false positive merges in exceptional cases.

We use the bipartite network to analyse whether our data also establishes evidence for the homophily assumption described in Section 1.1. Accordingly, we compute the relative frequency of fraudulent and non-fraudulent claims occurring in the neighbourhoods of other fraudulent and non-fraudulent claims. The availability or lack of evidence supporting the homophily assumption shall provide some indication as to the validity of this assumption

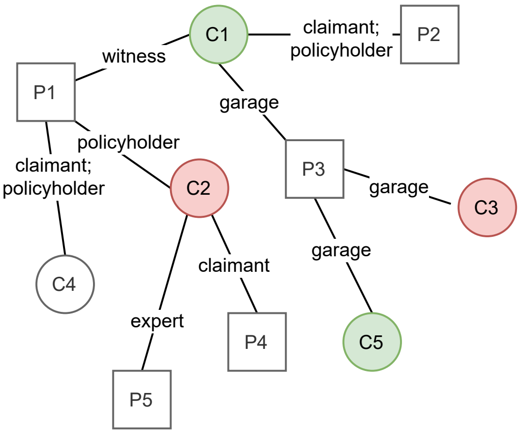


FIGURE 4.3: Example network illustrating claims and involved parties

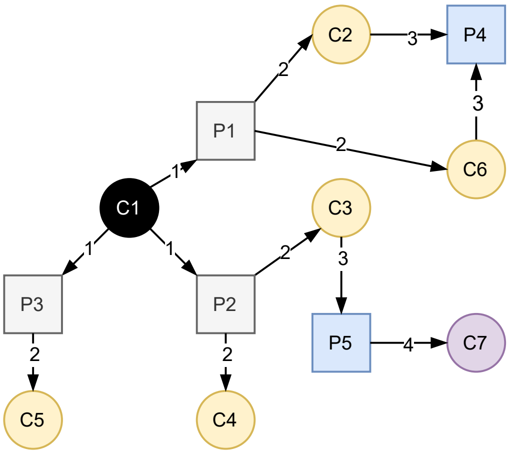


FIGURE 4.4: Example network illustrating kth-order neighbourhoods of claim 'C1'

across different insurance fraud data sets. In turn, this might prompt suggestions for future approaches to insurance fraud detection.

Formally, we let $G = (C \cup P, E)$ denote the bipartite graph G consisting of nodes C and P and edges E. Each node in G belongs to exactly one of the vertex sets C and P, and edges in E connect one node in C to one node in P. The number of nodes in C and P are represented as $|C|$ and $|P|$, respectively. We let C correspond to insurance claims and P represent the various parties involved in the claims. Then, c_i and c_j denote individual claim and party nodes, where $i \in \{1, \dots, |C|\}$ and $j \in \{1, \dots, |P|\}$.

Edges in E carry non-negative weights w_{ij} modelling the strength of the relationship between c_i and p_j. The lack of a connection between node c_i and p_j is represented by a weight of zero. In unweighted networks, w_{ij} is a binary indicator, where

$$w_{ij} = \begin{cases} 1 & \text{if } c_i \text{ and } p_j \text{ are connected;} \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

The edge weights altogether are represented as a $|C| \times |P|$ weight matrix $W = (w_{ij})$ with $i \in \{1, \dots, |C|\}$ and $j \in \{1, \dots, |P|\}$. As we are working with undirected edges, it holds that $w_{ij} = w_{ji}$ for all $i \in |C|, j \in |P|$.

The kth-order neighbourhood of a node c_i, denoted as $\mathcal{N}_{c_i}^k$, represents the set of all nodes that are connected to node c_i via a path of exactly k edges. Figure 4.4 provides a visual illustration of this concept for example claim ‘C1’. In the figure, arrowheads depict the direction ‘away from’ C1 and edge labels indicate that the target node is k steps away from C1. P1–3 resemble the first-order neighbourhood; C2–5 the second-order neighbourhood; P4–5 the third-order neighbourhood; and C7 the fourth-order neighbourhood.

The first-order neighbourhood of a claim c_i is the set of all parties involved in a claim, formally defined as

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

The second-order neighbourhood consists of all other claims in which the parties in $\mathcal{N}_{c_i}^k$ are involved, formally defined as

$$\mathcal{N}_{c_i}^2 = \left\{c_k \mid w_{kj} \neq 0, p_j \in \mathcal{N}_{c_i}^1\right\} \setminus \{c_i\}. \tag{4.3}$$

The third- and fourth-order neighbourhoods are defined in a similar way:

$$\mathcal{N}_{c_i}^3 = \left\{p_l \mid w_{kl} \neq 0, c_k \in \mathcal{N}_{c_i}^2\right\} \setminus \mathcal{N}_{c_i}^1; \tag{4.4}$$

$$\mathcal{N}_{c_i}^4 = \left\{c_m \mid w_{ml} \neq 0, p_l \in \mathcal{N}_{c_i}^3\right\} \setminus \left(\{c_i\} \cup \mathcal{N}_{c_i}^2\right) . \tag{4.5}$$

The equations reveal a pattern that can be extended to kth-order neighbourhoods for $k > 4$. Since this study only considers neighbourhoods up to $k = 4$, this generalisation is omitted here.

The degree d_i of a node c_i delineates the sum of weights on the edges between node c_i and the nodes in its first-order neighbourhood. An equivalent characteristic of node p_j is represented by d_j. In unweighted networks, d_i is equivalent to $|\mathcal{N}_{c_i}^1|$, i.e., the number of nodes in the node’s first-order neighbourhood. However, more general formulae that are also applicable to weighted networks are given by

$$d_i = \sum_{j=1}^{|P|} w_{ij}; \qquad d_j = \sum_{i=1}^{|C|} w_{ij}. \tag{4.6}$$

The diagonal $|C| \times |C|$ matrix D_c denotes the weighted degrees of all vertices in C such that $(D_c)_{ii} = d_i$. Similarly, the diagonal $|P| \times |P|$ matrix D_p denotes the weighted degrees of all vertices in P such that $(D_p)_{jj} = d_j$ represents the weighted degree of node p_j.

4.3.2 Implementing the BiRank Algorithm

Following the construction of a bipartite network, we compute fraud scores of claims in the graph. To do so, we implement the BiRank algorithm [29]. The BiRank algorithm scores nodes in a bipartite network based on the number of connecting nodes as well as the scores of these connecting nodes. By initialising the algorithm following the approach in Óskarsdóttir et al. [19], these scores resemble the exposure of a claim to known fraudulent claims.

BiRank takes as input a bipartite graph $G = (C \cup P, E)$ and its corresponding weight matrix W (Section 4.3.1), along with query vectors $\boldsymbol{c^0}$ and $\boldsymbol{p^0}$. These query vectors encode the prior belief concerning the vertices in C and P, respectively, enabling us to include information on known fraud.

The algorithm outputs a function $f : P \cup U \to \mathbb{R}$, which maps each vertex in G to a real number such that $f(c_i)$ represents the ranking score of node c_i and $f(p_j)$ represents the ranking scores of node p_j. Hereafter, we simplify the notation by using c_i to denote

$f(c_i)$ and p_j to denote $f(p_j)$. The final ranking scores of all nodes are collected in the ranking vectors $\boldsymbol{c} = [c_i, \dots, c_n]$ and $\boldsymbol{p} = [p_i, \dots, p_m]$ where $n = |C|$ and $m = |P|$.

The score of a node is iteratively computed by taking the sum of the contribution of its connected nodes. To ensure convergence and stability, edge weights are normalised by the degree of its two connected nodes, such that

$$c_i = \sum_{j=1}^{|P|} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} p_j; \tag{4.7}$$

$$p_j = \sum_{i=1}^{|C|} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} c_i. \tag{4.8}$$

This use of symmetric normalisation, a 'key characteristic of BiRank' [29], lessens the contribution of high-degree nodes and should have a positive effect on the results whenever high-degree nodes are present, like in our data set (Section 5.2.1).

The prior information available in query vectors $\boldsymbol{c}^{\boldsymbol{0}}$ and $\boldsymbol{p}^{\boldsymbol{0}}$ is factored directly into the ranking process. Accordingly, Equations 4.7 and 4.8 are adapted to

$$c_i = \alpha \sum_{j=1}^{|P|} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} p_j + (1-\alpha) c_i^0; \tag{4.9}$$

$$p_j = \beta \sum_{i=1}^{|C|} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} c_i + (1-\beta) p_j^0, \tag{4.10}$$

where α and β serve as hyper-parameters that enable adjusting the influence of the graph structure and the prior query vector, to be set between $[0,1]$[1]. In matrix form, Equations 4.9 and 4.10 are expressed as

$$\boldsymbol{c} = \alpha S \boldsymbol{p} + (1-\alpha)\boldsymbol{c}^{\boldsymbol{0}}; \tag{4.11}$$

$$\boldsymbol{p} = \beta S^{\top} \boldsymbol{c} + (1-\beta)\boldsymbol{p}^{\boldsymbol{0}}, \tag{4.12}$$

where $S = D_c^{-\frac{1}{2}} W D_p^{-\frac{1}{2}}$ is the symmetrically normalised weight matrix. Together, Equations 4.11 and 4.12 form the basis for the iterative BiRank algorithm, presented in Algorithm 1. Note that the algorithm makes reference to stopping criteria. He et al. [29] propose that these can be formulated based on either attaining a sufficiently small change in ranking vectors $\boldsymbol{c}, \boldsymbol{p}$ or based on a comparison with validation data to prevent overfitting.

4.3.3 Employing BiRank for Computing Fraud Scores

To employ the BiRank algorithm for the purpose of insurance fraud detection, we construct query vector $\boldsymbol{c}^{\boldsymbol{0}}$ such that

$$c_i^0 = \begin{cases} 1 & \text{if } l_i \in \{\text{fraudulent}\}; \\ 0 & \text{if } l_i \in \{\text{non-fraudulent}, \text{unknown}\}, \end{cases} \tag{4.13}$$

where l_i denotes the label of a claim. Claims labelled 'fraudulent' have undergone a fraud investigation and failed to dispel the suspicion of fraud. Claims labelled as 'non-fraudulent'

[1] Note that α in the formulas reported in Óskarsdóttir et al. [19] is equivalent to β in He et al. [29], and vice versa.

Algorithm 1 The Iterative BiRank Algorithm

```
Input: Weight matrix $W$, query vector $\boldsymbol{c^0}$, $\boldsymbol{p^0}$, and hyper-parameters $\alpha$, $\beta$
Output: Ranking vectors $\boldsymbol{c}$ and $\boldsymbol{p}$
Symmetrically normalise $W$: $S = D_c^{\frac{1}{2}} W D_p^{\frac{1}{2}}$
Randomly initialise $\boldsymbol{c}$ and $\boldsymbol{p}$
while Stopping criteria are not met do
    $\boldsymbol{c} \leftarrow \alpha S \boldsymbol{p} + (1 - \alpha)\boldsymbol{c^0}$
    $\boldsymbol{p} \leftarrow \beta S^T \boldsymbol{u} + (1 - \beta)\boldsymbol{p^0}$
end while
return $\boldsymbol{c}$ and $\boldsymbol{p}$
```

have also been investigated but were confirmed as non-fraudulent. Claims labelled 'unknown' have never been subject to a fraud investigation.

We define $\boldsymbol{p^0} \equiv \boldsymbol{0}$, following the idea that whereas parties can submit or be involved in fraudulent claims, they are not fraudulent themselves. This enables us to omit query vector $\boldsymbol{p^0}$ altogether, such that Equation 4.12 can be simplified to

$$\boldsymbol{p} = S^{\mathsf{T}} \boldsymbol{c} \tag{4.14}$$

4.3.4 Extracting Network Features

Execution of the BiRank algorithm is succeeded by the extraction of features from the resulting network. These features capture the network characteristics pertaining to the respective claim, which enables the use of these characteristics in supervised learning models. We extract features equivalent to those in the original work [19] and adopt the same categorisation, distinguishing between 'neighbourhood' features and 'score' features. The key distinction lies in score features relying on fraud scores computed using the BiRank algorithm introduced in Sections 4.3.2 and 4.3.3, whereas these are not utilised in neighbourhood features. This distinction shall provide insight into the value of using BiRank over considering relations between claims and parties in general. The neighbourhood features are presented in Table 4.2, whereas the score features are shown in Table 4.3.

In terms of score features, the different metrics each correspond to alternative characteristics, as denoted in the original work. For example, high maximum fraud scores (`n1.max` and `n2.max`) reveal that there is at least one node with a high score in the corresponding neighbourhood, whereas high first quartiles (`n1.q1` and `n2.q1`) suggest there are several of such nodes. Focusing on neighbourhood features instead, the original authors described utilising size and ratio features to represent the homophilic nature of fraud, whereas they included `n2.binFraud` because of its utilisation in a business rule employed by the company that provided their data.

4.3.5 Constructing the Analytical Model Data Sets

Next, we construct supervised learning models that utilise the intrinsic features defined in Section 4.1.1 and the score and neighbourhood features outlined in Section 4.3.4. These feature matrices are hereforth denoted as X^{intr}, X^{score}, and X^{nbh}, while the feature vectors of individual nodes c_i are represented as $\boldsymbol{x_i}^{\text{intr}}$, $\boldsymbol{x_i}^{\text{score}}$, and $\boldsymbol{x_i}^{\text{nbh}}$, respectively. In line with Óskarsdóttir et al. [19], we construct machine learning models for two distinct tasks.

Table 4.2: Neighbourhood features extracted from the network

Feature	Description
`n1.size`	The number of nodes in the node's first-order neighbourhood.
`n2.size`	The number of nodes in the node's second-order neighbourhood.
`n2.ratioFraud`	The number of known fraudulent claims in the node's second-order neighbourhood divided by `n2.size`.
`n2.ratioNonFraud`	The number of known non-fraudulent claims in the node's second-order neighbourhood divided by `n2.size`.
`n2.binFraud`	A binary value indicating whether there is a known fraudulent claim in the node's second-order neighborhood

Table 4.3: Score features

Feature	Description
`scores0`	The node's fraud score.
`n1.q1`	The first quartile of the empirical distribution of the fraud scores in the node's first-order neighbourhood.
`n1.med`	The median of the empirical distribution of the fraud scores in the first-order neighbourhood.
`n1.max`	The maximum of the empirical distribution of the fraud scores in the first-order neighbourhood.
`n2.q1`	The first quartile of the empirical distribution of the fraud scores in the node's second-order neighbourhood.
`n2.med`	The median of the empirical distribution of the fraud scores in the second-order neighbourhood.
`n2.max`	The maximum of the empirical distribution of the fraud scores in the second-order neighbourhood.

The first task involves distinguishing claims with a known label from those with an unknown label, with the motivation that this analysis provides insight into how claims are selected for further investigation. Thus, we define

$$y_i^{\text{known}} = \begin{cases} 1 & \text{if } l_i \in \{\text{fraudulent}, \text{non-fraudulent}\} \\ 0 & \text{if } l_i \in \{\text{unknown}\}, \end{cases} \tag{4.15}$$

where l_i represents the label assigned to a claim (see Section 4.3.3). The corresponding data set is denoted as $\mathcal{D}^{\text{known}}$ and formulated as $\mathcal{D}^{\text{known}} = \left\{\left(\boldsymbol{x_i}^{\text{intr}} \oplus \boldsymbol{x_i}^{\text{score}} \oplus \boldsymbol{x_i}^{\text{nbh}}\right), \boldsymbol{y_i}^{\text{known}}\right\}_{i=1}^{n}$, where $\oplus$ represents concatenation and n signifies the total number of claims.

The second task revolves around distinguishing claims with a known fraud label from the rest, i.e., known legitimate claims and unknown claims. To achieve this, we define

$$y_i^{\text{fraud}} = \begin{cases} 1 & \text{if } l_i \in \{\text{fraudulent}\} \\ 0 & \text{if } l_i \in \{\text{non-fraudulent}, \text{unknown}\}, \end{cases} \tag{4.16}$$

resulting in the data set $D^{\text{fraud}} = \left\{\left(\boldsymbol{x_i}^{\text{intr}} \oplus \boldsymbol{x_i}^{\text{score}} \oplus \boldsymbol{x_i}^{\text{nbh}}\right), \boldsymbol{y_i}^{\text{fraud}}\right\}_{i=1}^{n}$.

4.3.6 Employing Random Forests for Feature Importance Ranking

For each of the aforementioned two tasks, we evaluate the importance of all available features using random forests. This approach is consistent with the approach adopted in the existing paper [19], which facilitates a comparison between findings. Since the original authors did not specify the type of feature importance considered in their study, we assume they adopted one of the two 'default' RF feature importance measures proposed by the original author of the RF algorithm and consider the Gini importance and the permutation accuracy importance (herafter denoted as 'permutation importance'). Details on both approaches are found in Section 2.3. In what follows, we motivate our choice for evaluating the Gini importance only, but first argue that both approaches might provide—and might have provided—biased results.

Our reason for casting doubt on the choice for either of the two importance measures is based on the fact that the data utilised in this study exhibit variations in terms of measurement scale (e.g. continuous versus nominal) and the number of categories corresponding to the predictor variable. Consider, for example, the nominal `responsibilityCode` feature versus the continuous `amount` feature; and the comparatively large range of categories available for the `responsibilityCode` feature when compared to the binary `n2.binFraud`. In their work [111], authors assessed the impact of this type of variation on the reliability of random forest feature importance measures and found that, in situations with large variation, both permutation importance and Gini importance are unreliable, with the strongest bias observed in the Gini importance measure.

Consistent with this observation, it seems most reasonable to select the least unreliable importance measure: the permutation importance. However, the computation of permutation importance relies on either an explicit validation set or the use of out-of-bag (OOB) samples. The construction and use of an explicit validation set is not described by Óskarsdóttir et al. [19], Meanwhile, the OOB sample-based approach would provide unreliable results, since its use in conjunction with the employed oversampling approach introduces data leakage between in- and out-of-bag samples, similar to the data leakage introduced by oversampling before a train-test split [112]. Apart from these concerns, the permutation importance approach may also reveal bias resulting from correlated features in the data set and unrealistic data instances caused by permutations [113, 114].

Following the aforementioned considerations, in this work, we only consider the Gini importance. We acknowledge its limitations and recognise the constraints in our reporting on the importance of individual features.

4.3.7 Evaluating Logistic Regression Classifier

Following random forest feature ranking, we build logistic regression classifiers [115]. Logistic regression classifiers for binary classification define the probability of the target (i.e., dependent variable) y being equal to one as the value of the logistic function $\sigma = \frac{1}{1+e^{-t}}$ of the linear regression expression $t = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$. Correspondingly,

$$p(y = 1 \mid x_1, \ldots, x_n) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{n} \beta_i x_i)}}, \tag{4.17}$$

where $x_1, \ldots, x_n$ denote the explanatory (i.e., independent) variables, $\beta_1, \ldots, \beta_n$ denote the corresponding coefficients, and β_0 specifies the intercept—the value of the linear regression expression t when all explanatory variables equal zero. The parameters $\boldsymbol{\beta}$ are optimised numerically using one of many available optimisation algorithms [116], which 'fit' the classifier to a sample of training data X and corresponding target labels $\boldsymbol{y}$.

In the construction of our logistic regression classifiers, we first add one feature at a time, starting with the most important feature. We evaluate the performance of the model at each iteration, thereby yielding insight into the performance of the model at various levels of complexity. Then, we build a logistic regression classifier using all features. This classifier is used for the actual classification of claims and, contrary to the random forest models, would be one of the models deployed in practice. The results of the final logistic models provide some insight into the generalisability of findings reported by authors of the original model. In addition, they offer a baseline to compare the results of adapted models against.

Note that, while Óskarsdóttir et al. [19] employ stepwise regression using a combination of both forward and backward feature selection to construct this final classifier, we construct the logistic regression model using all features instead to ensure consistency in our reporting. Without this approach, the results of fraud experts' evaluation (Section 4.6.4) would rely on different models compared to those employed for generating other results. This discrepancy arises from time and resource constraints preventing the implementation of stepwise forward and backward regression before the evaluation takes place.

4.4 Incorporating Time-Weighting

The first adaptation involves extending the baseline model to include time-weighted fraud influence, which shall emphasise recent fraud over historical fraud. In that regard, Óskarsdóttir et al. [19] make reference to the methodology introduced by Van Vlasselaer et al. [16]. However, this approach was construed using the Personalised PageRank algorithm [117], instead of BiRank. Additionally, while the temporal nature of the relations between companies and resources in their graphs allow for a sensible distinction between 'time-weighted fraud influence' (related to the time since fraud detection) and 'time-weighted edges' (related to the time since the relation appeared), the permanent nature of relations between claims and parties allow no such distinction to be sensibly made. Following these observation, our strategy for incorporating time-weighting will involve first adapting the proposed approach for compatibility with bipartite graphs, and then incorporating this approach in the new fraud detection model.

We implement time-weighted fraud influence by changing query vector $\boldsymbol{c^0}$ in Equation 4.13 to

$$c_i^0 = \begin{cases} e^{-\lambda h} & \text{if } l_i \in \{\text{fraudulent}\}; \\ 0 & \text{if } l_i \in \{\text{non-fraudulent}, \text{unknown}\}, \end{cases} \tag{4.18}$$

where h denotes the time passed since the fraudulent claim was submitted and λ is a hyper-parameter representing the decay constant.

We aim to integrate the full impact of the adaptations into the final model solely through changes in fraud scores. Consequently, the time-weighted model adapts the exact same features employed in the baseline model. This approach minimises the alterations to independent variables and enhances our confidence in attributing shifts in model performance to the influence of time-weighted fraud influence.

4.5 Incorporating Shared Resources

The second modification to the baseline model entails incorporating relations between parties based on resources they share, such as email addresses, bank account numbers, and telephone numbers. These indicators suggest close relationships among the parties involved and, in line with the homophily assumption, are noteworthy for fraud detection models. Our focus is exclusively on resources that are deliberately shared among individuals, minimising the likelihood of coincidental sharing. Consequently, we acknowledge a shared full home address as a shared resource but refrain from establishing relations based on less granular information, such as the street name alone. Additionally, we assign no intrinsic value to the resource itself; it only acts as a linking pin. As a result, the final model shall not discriminate based on factors like geographical location, reducing the chance of unfair bias.

To include shared resources into the model, we introduce a third type of entity, the 'shared resource', and transform the bipartite graph into a tripartite graph of claims, parties, and shared resources. By choosing this approach over an approach that represents these relations as edges between party nodes directly (i.e., edges between two nodes of the same type), we continue adhering to the characteristics of n-partite graphs. This enables us to adopt the tripartite variant of BiRank.

First, we establish the tripartite graph by defining that $G = (C \cup P \cup R, E_{cp} \cup E_{pr} \cup E_{cr})$ with nodes C, P, and R and edges E_{cp}, E_{pr}, E_{cr}. We let C correspond to insurance claims, P represent the various parties involved in the claims, and R denote shared resources and designate the different node types using the symbols c, p, and r. Each node in G exclusively belongs to one of the vertex sets C, P and R. The number of nodes in each of the vertex sets C, P, and R are represented as $|C|$, $|P|$, and $|R|$, respectively. We use c_i, p_j, and r_k to denote individual claim, party, and resource nodes, where $i \in \{1, \ldots, |C|\}$, $j \in \{1, \ldots, |P|\}$, and $k \in \{1, \ldots, |R|\}$. For any pair of different node types $x, y \in \{c, p, r\}$ where $x \neq y$, edges in E_{xy} establish a connection between one node of type x and one node of type y. Since G is undirected, it holds that $E_{xy} \equiv E_{yx}$.

Having established the tripartite graph, we define the corresponding tripartite graph ranking algorithm. For that purpose, we replace Equations 4.11 and 4.12 with Equations 4.19 and 4.20, and introduce Equation 4.21 for the shared resource ranking vector $\boldsymbol{r}$,

Algorithm 2 The Iterative Tripartite Ranking Algorithm

```
Input: Weight matrix $W^\intercal_{cp}(= W^\intercal_{pc})$, $W_{pr}(= W^\intercal_{rp})$, $W_{rc}(= W^\intercal_{cr})$; query vectors $\boldsymbol{c}^{\mathbf{0}}$, $\boldsymbol{p}^{\mathbf{0}}$, $\boldsymbol{r}^{\mathbf{0}}$;
    and hyper-parameters $\alpha_{XY}$ for all node types $x, y \in \{c, p, r\}$ where $x \neq y$.
Output: Ranking vectors $\boldsymbol{c}$, $\boldsymbol{p}$, and $\boldsymbol{r}$
For all $x, y \in \{c, p, r\}$ where $x \neq y$, symmetrically normalise $W : S = D_x^{\frac{1}{2}} WxyD_y^{\frac{1}{2}}$
Randomly initialise $\boldsymbol{c}$, $\boldsymbol{p}$, and $\boldsymbol{r}$
while Stopping criteria are not met do
    $\boldsymbol{c} \leftarrow \alpha_{cp} S_{cp} \boldsymbol{p} + \alpha_{cr} S_{cr} \boldsymbol{r} + (1 - \alpha_{cp} - \alpha_{cr}) \boldsymbol{c}^{\mathbf{0}}$
    $\boldsymbol{p} \leftarrow \alpha_{pc} S_{pc} \boldsymbol{c} + \alpha_{pr} S_{pr} \boldsymbol{r} + (1 - \alpha_{pc} - \alpha_{pr}) \boldsymbol{p}^{\mathbf{0}}$
    $\boldsymbol{r} \leftarrow \alpha_{rc} S_{rc} \boldsymbol{c} + \alpha_{rp} S_{rp} \boldsymbol{p} + (1 - \alpha_{rc} - \alpha_{rp}) \boldsymbol{r}^{\mathbf{0}}$
end while
return $\boldsymbol{c}$, $\boldsymbol{p}$, and $\boldsymbol{r}$
```

such that

$$\boldsymbol{c} = \alpha_{cp} S_{cp} \boldsymbol{p} + \alpha_{cr} S_{cr} \boldsymbol{r} + (1 - \alpha_{cp} - \alpha_{cr}) \boldsymbol{c}^{\mathbf{0}}; \tag{4.19}$$

$$\boldsymbol{p} = \alpha_{pc} S_{pc} \boldsymbol{c} + \alpha_{pr} S_{pr} \boldsymbol{r} + (1 - \alpha_{pc} - \alpha_{pr}) \boldsymbol{p}^{\mathbf{0}}; \tag{4.20}$$

$$\boldsymbol{r} = \alpha_{rc} S_{rc} \boldsymbol{c} + \alpha_{rp} S_{rp} \boldsymbol{p} + (1 - \alpha_{rc} - \alpha_{rp}) \boldsymbol{r}^{\mathbf{0}}, \tag{4.21}$$

For $x, y \in \{c, p, r\}$ with $x \neq y$, $S_{xy} = S^\intercal_{yx} = D_x^{-\frac{1}{2}} W_{xy} D_y^{-\frac{1}{2}}$ is the symmetrically normalised weight matrix and α_{xy} represents a numerical hyperparameter for adjusting the influence of the graph structure and the prior query vector while considering the propagation from nodes of type x to nodes of type y. Equations 4.19 to 4.21 are employed in the iterative tripartite ranking algorithm, which is displayed in Algorithm 2.

In line with the statement in Section 4.3.3 that we consider only claims to be fraudulent, we can once again omit the query vectors in Equations 4.20 and 4.21 and change the equations to

$$\boldsymbol{p} = \alpha_{pc} S_{pc} \boldsymbol{c} + \alpha_{pr} S_{pr} \boldsymbol{r}; \tag{4.22}$$

$$\boldsymbol{r} = \alpha_{rc} S_{rc} \boldsymbol{c} + \alpha_{rp} S_{rp} \boldsymbol{p}, \tag{4.23}$$

simplifying future calculations. In addition, we note that shared resources are exclusively connected to parties and not to claims, implying that $E_{cr} = \varnothing$. Consequently, $W_{cr} = W^\intercal_{rc}$ and $S_{cr} = S^\intercal_{rc}$ are zero matrices, enabling a further simplification of the calculations to

$$\boldsymbol{c} = \alpha_{cp} S_{cp} \boldsymbol{p} + (1 - \alpha_{cp}) \boldsymbol{c}^{\mathbf{0}}; \tag{4.24}$$

$$\boldsymbol{p} = \alpha_{pc} S_{pc} \boldsymbol{c} + \alpha_{pr} S_{pr} \boldsymbol{r}; \tag{4.25}$$

$$\boldsymbol{r} = S_{rp} \boldsymbol{p}, \tag{4.26}$$

This results in the simplified ranking algorithm presented in Algorithm 3.

Consistent with the approach discussed in Section 4.4, this adapted model uses the exact same features used in the baseline model. Accordingly, after the ranking algorithm has been executed, we eliminate shared resources from the network entirely, enabling the reuse of feature extraction processes previously generated for the baseline model.

4.6 Evaluation

We compare and evaluate our models using the same metrics reported by Óskarsdóttir et al. [19], facilitating a comparison to their existing work. Accordingly, we report on the

Algorithm 3 Simplified Iterative Tripartite Ranking Algorithm

```
Input: Weight matrices $W_{cp}(= W_{pc}^{\intercal})$, $W_{pr}(= W_{rp}^{\intercal})$; query vector $\boldsymbol{c}^{\mathbf{0}}$; and hyper-parameters
    $\alpha_{cp}, \alpha_{pc}, \alpha_{pr}$.
Output: Ranking vector $\boldsymbol{c}$
For all $(x, y) \in \{(cp), (pc), (pr), (rp)\}$, symmetrically normalise $W_{xy}$ : $S_{xy} = D_x^{\frac{1}{2}} W xy D_y^{\frac{1}{2}}$
Randomly initialise $\boldsymbol{c}$, $\boldsymbol{p}$, and $\boldsymbol{r}$
while Stopping criteria are not met do
    $\boldsymbol{c} \leftarrow \alpha_{cp} S_{cp} \boldsymbol{p} + (1 - \alpha_{cp}) \boldsymbol{c}^{\mathbf{0}}$
    $\boldsymbol{p} \leftarrow \alpha_{pc} S_{pc} \boldsymbol{c} + \alpha_{pr} S_{pr} \boldsymbol{r}$
    $\boldsymbol{r} \leftarrow S_{rp} \boldsymbol{p}$
end while
return $\boldsymbol{c}$, $\boldsymbol{p}$, and $\boldsymbol{r}$
```

area under the receiver operating characteristic curve, the area under the precision recall curve, and the top-decile lift, elucidated in Sections 4.6.1 to 4.6.3.

The three aforementioned metrics are computed based on known claim labels only and thus provide no insight into the performance of the model in detecting previously unknown fraudulent claims. We address this limitation, and thereby extend the original work, by collecting a sample of the top-k predicted fraudulent claims from each model and having these investigated by fraud experts. This is elaborated upon in Section 4.6.4.

4.6.1 Area Under the Receiver Operating Characteristic Curve

The area under the receiver operating characteristic curve (AUC-ROC) is a performance metric used to quantify the effectiveness of a binary classification model by summarising the receiver operating characteristic curve (ROC) into a single number. The ROC curve is depicted in a two-dimensional graph and visually illustrates the trade-off between the number of correctly classified positive samples and the number of incorrectly classified negative samples at various classification thresholds [118].

In the ROC curve, the true positive rate (TPR)—also known as ‘sensitivity’ or ‘recall’—and false positive rate (FPR) are depicted on the horizontal and vertical axes, respectively, for various classification thresholds γ. This threshold determines the predicted class $\hat{y}$ of a record x according to the conditional assignment

$$\hat{y}_x = \begin{cases} 0 & \text{if } P(x) < \gamma \\ 1 & \text{if } P(x) \geq \gamma \end{cases}, \tag{4.27}$$

where $P(x)$ denotes the probability that a record x belongs to the positive class, as computed by the model. Equations 4.28 and 4.29 reveal the formulas used to calculate the TPR and FPR, respectively. These equations incorporate the counts of true negatives (TN), true positives (TP), false negatives (FN), and false positives (FP), whose meanings are visualised in the confusion matrix in Table 4.4.

$$\text{true positive rate} = \text{recall} = \frac{TP}{TP + FN} \tag{4.28}$$

$$\text{false positive rate} = \frac{FP}{FP + TN} \tag{4.29}$$

Table 4.4: Confusion matrix

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

The AUC-ROC ranges from 0.0 to 1.0, with higher values reflecting superior model performance. Values below 0.5 represent worse performance than random choice, a value of 0.5 indicates that the model performs no better than a random model, while a value of 1.0 signifies a (theoretically) perfect model.

4.6.2 Area Under the Precision–Recall Curve

An alternative to utilising an ROC curve is to consider the precision–recall curve (PR). Davis and Goadrich [119] refer to six distinct papers in citing that PR curves are used as an alternative to ROC curves for tasks with heavily imbalanced data sets. PR curves are similar to ROC curves in that they denote the recall (TPR) on one axis. However, whereas ROC curves combine the recall with the FPR, PR curves replace the FPR with the precision, defined as

$$\text{precision} = \frac{TP}{TP + FP}. \tag{4.30}$$

In addition, the axes are swapped. The PR curve depicts the recall on the horizontal axis and the precision on the vertical axis.

In the context of data sets with heavily skewed distributions towards the negative class, the impact of replacing the FPR in ROC curves with the precision in PR curves is that a large change in the number of false positives yields a much larger influence on the PR curve than on the ROC curve. As a result, the PR curve better depicts the impact of the large number of negative samples on the performance [119] such that, for imbalanced data sets, the PR curve is recommended over the ROC curve [120]. This idea extends to the AUC-PR and AUC-ROC.

Similar to AUC-ROC, AUC-PR values range from 0.0 to 1.0 where higher values signify superior model performance. However, unlike AUC-ROC, the baseline in AUC-PR is non-universal; it varies based on the class distribution [121]. The baseline PR curve takes takes the form of a horizontal line with a height (i.e., precision) equivalent to the proportion of positive samples in the data set. Consequently, given a width of 1, the baseline AUC-PR is also equivalent to the proportion of positive samples in the data set.

4.6.3 Top-Decile Lift

Lift indicates how much better a model is at identifying positive cases compared to random selection. It is calculated over a segment of the data, which—in the case of top-decile lift (TDL)—comprises the top-10% of samples that are attributed the highest probability by the model. Lemmens and Croux [122] provide the following equation:

$$\text{top decile lift} = \frac{\hat{\pi}_{10\%}}{\hat{\pi}}. \tag{4.31}$$

Here, $\hat{\pi}_{10\%}$ denotes the fraction of positive cases in the top-10% of samples that were given the highest probability by the model, whereas $\hat{\pi}$ denotes the proportion of positive samples in the whole data set.

Elevated lift values signify improved model performance. As can be derived from Equation 4.31, lift values below one signify that the model performs worse than random selection; a lift value equal to one suggests equivalent performance to random selection; lift values exceeding one indicate superior model performance compared to random selection.

4.6.4 Fraud Expert Evaluation of Unlabelled Claims

The aforementioned metrics rely solely on known labels and offer no insights into the model's ability to correctly classify previously unlabelled claims as potentially fraudulent. To address this limitation, we engage fraud experts within the insurance company to evaluate a small sample of the top-k unlabelled claims from each model. The goal of this evaluation is to assess whether the models can correctly classify previously unlabelled claims as warranting further investigation, which presents an enhancement over the work in Óskarsdóttir et al. [19].

In this specific assessment, we consider the full baseline and adapted models and collect the top-k unlabelled claims based on their predicted probability of fraud, as computed by the respective model. The subsequent evaluation is summarised using the precision-at-k metric [123], which measures the precision (Equation 4.30) achieved on the top-k retrieved records. The advantage of utilising this metric lies in its independence from an estimation of the total number of investigation-worthy claims in the data set.

4.7 Conclusion

In conclusion, this chapter has established the methodological framework that is adopted to address this study's research questions. Next, Chapter 5 details the practical implementation of the methodology and a description of the experimental design, granting insights into the validity of the results and findings discussed in later chapters.

Chapter 5

Experimental Setup

Having established the theoretical foundations of our work in Chapter 4, the current chapter proceeds with a description of the practical implementation and execution the research. In Section 5.1 and 5.2, we present details concerning the construction and characteristics of the data set employed in this study. Following this, Section 5.3 elucidates the approach taken toward establishing empirical evidence for the homophily assumption. Then, Section 5.4 describes details concerning the implementation of the baseline model, followed by additional information about the integration of time-weighting and shared resources in Sections 5.5 and 5.6, respectively. Section 5.7 outlines how models' retrieval of previously unknown (i.e., unlabelled) claims are evaluated.

5.1 Data Set Construction

The process of constructing a data set that aligns closely with the data set employed by Óskarsdóttir et al. [19] involves the merging and processing of data from a wide variety of databases and tables present at the insurance company. This data is available via distributed Apache Hive [124] tables, enabling their processing using the Apache Spark [125] data analytics engine for large-scale data processing. The interaction with Apache Spark is established using PySpark [126], the Python application programming interface (API) for Apache Spark. More specifically, its Spark SQL API is employed, which offers an interface that shows major similarities to the standard SQL database language.

In this work, we omit many details regarding the exact steps taken to process the raw data into the features and data described in Section 4.1 because of confidentiality. Nevertheless, we present further information regarding our interpretation of the `amount` and `claimAge` features to address issues of multi-interpretability. We note that `amount` is calculated as the sum of all successful payments made towards a claim, combined with all remaining reservations. Consequently, `amount` signifies the total damage burden for the insurer. Additionally, we specify that `claimAge` denotes the number of months from the beginning of the *most recently entered* contract to the date the incident occurred. Since Óskarsdóttir et al. [19] did not describe their approach to calculating `claimAge` for claims covering multiple contracts, this approach might be different from the approach adopted in their study. However, our approach shall aid in detecting 'past posting', an insurance fraud scheme in which a person claims damages on insurance they obtained after the damage was incurred [26].

As mentioned in Section 4.1.1, not all claims have complete intrinsic property information. Table 5.1 shows the ratio of all claims for which each intrinsic feature is available. In Section 5.4.4, we compare these ratios for different classification target values and elucidate

Table 5.1: Percentage of claims with missing feature in data set

Feature	Missing
`responsibillityCode`	50.11%
`claimAge`	3.77%
`lastClaim`	29.42%
`amount`	3.80%

Table 5.2: Summary statistics of claims, parties, and associated relations

Number of ...	Study	Mean	Min	Max	Median
Distinct parties per claim	This study	1.81	1	33	2
	[19]	3.79	1	42	3
Distinct relations per claim	This study	2.59	1	33	2
Distinct relation types per claim	This study	2.34	1	7	2
Distinct claims per party	This study	2.38	1	95,380	1
	[19]	3.84	1	125,951	N/A

our approach to dealing with missing information.

The processed data is divided into three individual data sets, each serving a distinct purpose: an *intrinsic features* data set, a *network claims* data set, and an *edge list*. These are individually stored using the Parquet column-oriented file format [127]. The intrinsic features data set encompasses all intrinsic features of the hundreds of thousands of automobile insurance claims explicitly considered in this work. The network claims data set contains the necessary claims data for constructing the bipartite network, thereby incorporating claims from other P&C lines of business as well. The edge list includes a record for each relation between a party and a claim in the network, including information on the involvement type and the partner type (e.g. 'person' or 'organisation').

5.2 Data Set Characteristics

After constructing the data sets, we extract the data characteristics described in Section 4.1.3. The extraction of simple degree-related characteristics involves aggregations on the edge list data set. This is a straightforward process conducted in PySpark and therefore not further elaborated upon, but the aggregated data is reported upon in Section 5.2.1. Extracting four-cycles is more intricate. Our approach and the corresponding results are displayed in Section 5.2.2.

5.2.1 Basic Network Characteristics

Table 5.2 presents statistics summarising the characteristics of claims, parties, and relations in our data set. The table reveals that the mean and maximum degree of claims in our data set are 1.81 and 33, respectively, which contrasts with the mean degree of 3.79 and maximum degree of 42 reported by Óskarsdóttir et al. [19]. Meanwhile, the reported minimum degrees of claims in the two data sets are equivalent at values of 1, while the median degrees of 2 in our data set differs from the median degree of 3 reported by the original work's authors. The observed differences may be partially attributed to differences in the types of involvement considered in the two data sets. However, this attribution cannot be definitively verified.

Other characteristics presented in Table 5.2 show that, on average, 2.34 distinct types

Table 5.3: Relative relation type frequency in data set

Relation type	Relative frequency
claimant	0.3944
policyholder	0.3870
garage	0.2121
representative	0.0030
witness	0.0023
legal counsel	0.0012
expert	0.0005
agent	0.0001

of relations are associated with a claim. Meanwhile, the median number of relation types is 2 and the maximum is 7. Table 5.3 provides insight into the relative occurrence frequency of each considered type of relation. It reveals that the relative frequencies of the claimant, policyholder, and garage relation types far exceed the relative frequency of all others, which indicates that representatives, experts, witnesses, legal counsel, and agents are involved in a comparatively small number of claims.

Moving on, we focus on statistics that describe the degree of parties, i.e., the number of claims that each party is connected to. Óskarsdóttir et al. [19] delineate these statistics for each involvement type individually, enabled by their inclusion of the involvement type as a node attribute. In contrast, our approach treats involvement types as relationship attribute, allowing a party to assume different roles within a claim and across claims. Following this deviation, we cannot conclusively assign a type to a party and are limited to describing degree statistics for the entire set of parties only.

To nevertheless facilitate a comparison between the party degree statistics reported in the original work and those collected for our study, we translate their per-type statistics into general party statistics. To achieve this, we compute the weighted mean—weighted by the relative frequency of a specific party within their full set of parties—and collect the minimum and maximum degrees. The collected results are presented aside the same statistics for our data set in Table 5.2.

The median degree of parties reported by Óskarsdóttir et al. [19] cannot be conclusively derived from the reported data. Meanwhile, the observed mean degree of parties in our data set deviates from theirs by a factor of $-2/5$, while the maximum degree observed in our data is similarly large at approximately $3/4$ of their reported maximum degree.

5.2.2 Four-Cycles

To extract four-cycles, in line with the methodology presented in Section 4.1.3, we first load the edge list into a Pandas [128, 129] DataFrame. Then, we employ the NetworkX Python library [130] to generate a network from this edge list. We then generate a subgraph of the aforementioned network, filtered on investigated claims and their involved parties, to enable extracting only four-cycles in which both claims are labelled (i.e., both claims are investigated)—matching the work in Óskarsdóttir et al. [19]. We employ NetworkX's implementation of the bounded-length simple cycle algorithm [131] to extract the actual four-cycles, after which we assess the type of each included party and the fraud status of each included claim to obtain the aggregate statistics.

Table 5.4 presents statistics concerning 96 four-cycles with known claim labels in our bipartite network, facilitating a direct comparison to the findings in Óskarsdóttir et al. [19]. Comparing these statistics with those presented by the original authors reveals a notable

Table 5.4: Relative frequency of four cycles with zero, one, and two fraudulent claims among four-cycles with known claim labels

		Composition of party nodes		
Nr. of fraudulent claim nodes	Study	Two people	One person; one organisation	Total
Zero	This study	11.11 %	14.10 %	13.54 %
	[19]	16.33 %	36.56 %	35.66 %
One	This study	22.22 %	17.95 %	18.75 %
	[19]	12.24 %	30.29 %	29.47 %
Two	This study	66.67 %	67.95 %	67.71 %
	[19]	71.43 %	33.15 %	34.87 %

difference in the relative frequency of four-cycles wherein both claims are fraudulent. In our data set, this frequency is significantly higher than in the data analysed in Óskarsdóttir et al. [19]. They report an approximately uniform distribution of four cycles with zero, one, and two fraudulent nodes (35.66%, 29.47%, and 34.87%, respectively). In our data set, these percentages are 13.54%, 18.75%, and 67.71%, respectively. The dissimilarity is primarily attributed to a difference in the corresponding distribution of four cycles involving one person and one organisation ('one person & one company' in Óskarsdóttir et al. [19]). These four cycles make up over 81% of all considered four cycles with known claim labels in our data.

5.3 Evaluation of Homophily Assumption

To assert whether our data establishes any empirical evidence for the homophily assumption, we use a highly resource-intensive approach in PySpark. In Section 5.3.1, we outline our approach to extracting the first- to fourth-order neighbourhoods of claims. Section 5.3.2 details the computed statistics for each neighbourhood type. Then, in Section 5.3.3, we elucidate our approach to assessing the significance of observed differences.

5.3.1 Establishing Neighbourhoods

Initially, we process the edge list data set to include at most one relation between each claim and party, thereby excluding information concerning the type of involvement. This processed data set is denoted as $E_{\text{simple}} = \{(c, p)\}$, where (c, p) signifies that the set consists of combinations of claims c and parties p. Subsequently, we duplicate E_{simple} and modify it by discarding all rows with uninvestigated claims, resulting in a table $E_{\text{nbh1}} = \{(c_{\text{center}}, p)\}$. Here, p again denotes parties, whereas c_{center} denotes the claim at the centre of attention, i.e., the claim whose first-order neighbourhood is represented by the associated parties. Effectively, E_{nbh1} includes the relations from all investigated claims to their first-order neighbourhoods.

Next, we obtain relations to claims in the second-order neighbourhood. To achieve this, we initiate a join operation between E_{simple} and a duplicate of E_{nbh1} based on the party column. We then remove the party column to retain a data set in the shape of $\{(c_{\text{center}}, c)\}$, from which we remove both duplicate records and records whose target (i.e., c) already appeared in a lower-order neighbourhood. If n were the neighbourhood's order, we have to consider orders $[n-2, n-4, \ldots, 0]$. In this specific case, this means considering the zeroth-order neighbourhood only, indicative of eliminating rows where $c_{\text{center}} \equiv c$. The resulting data set, $E_{\text{nbh2}} = \{(c_{\text{center}}, c)\}$, includes the relations from all investigated claims to their second-order neighbourhoods.

To obtain relations to parties in the third-order neighbourhood, we replicate the process used to obtain relations to claims in the second-order neighbourhood. However, this time, we join E_{simple} with a duplicate of E_{nbh2} based on the claim column c. Then, after eliminating the claim column, removing duplicate rows, and removing rows whose target already appeared in a lower-order neighbourhood, we obtain $E_{\text{nbh3}} = \{(c_{\text{center}}, p)\}$. The fourth-order neighbourhood E_{nbh4} is then obtained using a process similar to that described for the second-order neighbourhood.

5.3.2 Extracting Neighbourhood Statistics

Following the construction of edge lists representing relations between labelled claims and their first- to fourth-order neighbourhoods, we extract aggregate statistics from E_{nbh2} and E_{nbh4}. First, we accompany each edge list with information on whether each centre claim c_{centre} was fraudulent or non-fraudulent, and whether neighbourhood claims c were investigated and, if known, fraudulent or non-fraudulent. Then, for each claim c_{centre}, we extract the counts of claims, investigated claims, known fraudulent claims, and known non-fraudulent claims in the respective neighbourhood. These counts are utilised to compute the ratio of fraudulent and non-fraudulent claims among all claims and among all investigated claims in the neighbourhood of the respective claim $\mathrm{c}_{\text{centre}}$. The mean and variance of these ratios are then computed, offering insights into these metrics for the neighbourhoods of all known fraudulent and non-fraudulent claims. This analysis forms the basis for evaluating whether the data provides empirical evidence supporting the homophily assumption.

Note that the computation of means and variances ignores undefined ratios related to division by zero. These are particularly prevalent among ratios concerning only investigated claims, since many neighbourhoods lack claims with a known label, leading to division by zero.

5.3.3 Asserting Significance of Observed Differences

The variances are used to conduct F-tests for equality of variances. These enable us to determine whether to use Student's T-test [132] or Welch's T-test [133] to assert the significance of differences in mean ratios.

Let $S_{c,n,r}$ represent the variance of the average relative frequency of fraudulent claims among r claims in the nth order neighbourhood of c claims. Here, r is one of 'investigated' or 'all', n is chosen from the set $\{1, 2\}$, and c is one of 'fraudulent' or 'non-fraudulent'.

For each type of neighbourhood n and ratio r, we first calculate the test-statistic

$$F(n, r) = \frac{S_{c=\text{fraudulent},n,r}}{S_{c=\text{non-fraudulent},n,r}}. \tag{5.1}$$

This test statistic, along with the degrees of freedom corresponding to each variance, is then input into SciPy's [134] cumulative distribution function `scipy.stats.f.cdf` and one minus its output is adopted as a p-value. If the computed p-value is below 0.05, we reject the null-hypothesis that the variances are equal and opt for Welch's T-test. Otherwise, we proceed with Student's T-test.

The T-tests are then performed using SciPy's `scipy.stats.ttest_ind` function, adjusting the `equal_var` parameter in alignment with the results of the related F-test. Other parameters are kept at their default values. Consistent with the calculation of the means and variances in Section 5.3.2, undefined ratios are excluded from the data employed in conducting the T-tests. The p-value returned by SciPy's function is used to conclude whether the observed difference is significant or not.

5.4 Implementing the Baseline Model

Having evaluated whether our data establishes any empirical evidence for the homophily assumption, we move on to building the baseline model. Óskarsdóttir et al. [19] rightfully emphasise the necessity of careful experimental design to capture the model's utility and constraints in real-world applications. For example, the classification of a claim in these experiments should not be influenced by information that would not have been available were the claim classified in a practical setting. Their approach for the computation of fraud scores and extraction of network-related features seems appropriate to sufficiently address these limitations. Consequently, we adopt a similar approach in this work.

Sections 5.4.1 to 5.4.3 present details regarding the aforementioned approach to initialising the query vector for the BiRank algorithm, implementing the BiRank algorithm, and extracting network features from the bipartite graph. In Section 5.4.4, we elucidate our strategy for constructing the data sets to be used in the analytical models. Then, Section 5.4.5 outlines our use of random forest models for feature importance ranking. Finally, Section 5.4.6 describes how the final logistic regression classifier is built.

5.4.1 Initialising the Query Vector

In Section 4.1, we highlighted that our data spans a period of four years. We employ this full data set for the construction of our bipartite graph of claims and parties. For the construction of the query vector $\boldsymbol{c^0}$, however, we only consider fraud in claims registered in the first three years of the covered date range. This is explained as follows.

Any claim included as a source of information in the query vector $\boldsymbol{c^0}$ will, by definition, obtain a high score when BiRank is applied—as can be derived from Equation 4.11. Accordingly, constructing a query vector based on all known data would allow subsequent supervised classification models to derive a correct fraud/no-fraud classification from the claim's fraud score alone. In effect, the model would have access to important data that would not have been available during the classification of a claim in a practical setting, which would yield results that would not generalise to the practical application of the model. By taking the current approach, we retain sufficient sources of information in the network from the historical claims, while enabling the last year of claims to be appropriately used for model building and evaluation in a setting reflecting that of a deployed model.

5.4.2 Implementing BiRank

Having established our approach to constructing a query vector, we describe details concerning our implementation and use of BiRank. First, we modify the *edge list* mentioned in Section 5.1 to retain only a single edge for each combination of claim and party, discarding involvement types entirely. Then, we assign each edge a weight of 1, followed by transforming the edge list into an adjacency matrix. This adjacency matrix contains a row for each claim and a column for each party in the data set. Whenever a relation exists between a claim and a party, the cell at coordinate (<claim row>, <party column>) is populated with the weight of the corresponding edge.

The adjacency matrix is substantial in terms of its number of rows and columns. However, it is also extremely sparse, since a single claim tends to show a relation to only few parties, and vice versa. This enables us to use data structures specifically intended for storing sparse matrices. We adopt the SciPy compressed sparse row (CSR) matrix: a sparse matrix that enables efficient arithmetic operations and is therefore suitable for use in the BiRank algorithm.

Our Python implementation of the BiRank algorithm was adapted from an existing Python implementation [135], modified to more closely align with the algorithm presented in Algorithm 1 and originally proposed by He et al. [29]. The main change in comparison to the implementation in Aronson [135] includes randomisation of the initial vertex scores, while a different change is found in the newfound support for defining a maximum number of iterations as stopping criterion.

Equivalent to Óskarsdóttir et al. [19], we run BiRank with $\alpha = 0.85$, representing the influence of the graph structure relative to prior information on claims. As the stopping criterion, we opt for selecting a maximum number of iterations in the order of thousands that exceeds the maximum number of iterations needed for convergence by a long shot, enabled by the short computation time required for a single iteration. In a practical setting, this would be further optimised to eliminate unnecessary iterations.

5.4.3 Extracting Network Features

After running the BiRank algorithm using the adjacency matrix only, we again build a bipartite network using the NetworkX Python library previously mentioned in Section 5.2.2. This time, the network is extended by assigning each node properties pertaining to their fraud score and, for claims, whether the node has been investigated and confirmed (non-)fraudulent. Subsequently, we iterate over all automobile insurance claims in the network that were submitted in the most recent year and extensively use NetworkX's `neighbors` function to extract score and neighbourhood features.

The `n1.q1`, `n1.med`, `n2.q1`, and `n2.med` features are computed by employing NumPy's [136] `quantile` function for probabilities 0.25 and 0.50, with default values for the other parameters. Accordingly, the quantiles are estimated using the 'linear' method, corresponding to method definition 7 in Hyndman and Fan [137]. Whenever features cannot be properly calculated, which can occur when a claim has no second-order neighbourhood, the corresponding feature is set to zero. The extracted features are again stored in Parquet files, enabling their reuse.

5.4.4 Preparing Analytical Model Data Sets

In line with the methodology in Section 4.3.5, we construct data sets to be used by the supervised classification models. In this section, we first present an analysis of the missing features in each data set, along with our approach to addressing this issue. Then, we describe the train–test split and preprocessing strategy that is employed.

Dealing with missing features

For our analytical models, we construct two independent data sets D^{known} and D^{fraud} with targets y_i^{known} and y_i^{fraud}, respectively. However, before doing so, we decide on how to deal with the incomplete intrinsic information previously revealed in Table 5.1. For that purpose, we first compute what percentage of claims is missing each feature for $y_i^{\text{known}} = 0$, $y_i^{\text{known}} = 1$, $y_i^{\text{fraud}} = 0$, and $y_i^{\text{fraud}} = 1$, respectively. This enables us to assert whether a missing value is in some way indicative of the claim being known or the claim being known fraudulent. The results are presented in Table 5.5.

Table 5.5: Percentage of claims with missing feature for each target type

	D^{known}		D^{fraud}	
Feature	$y_i^{\text{known}} = 0$	$y_i^{\text{known}} = 1$	$y_i^{\text{fraud}} = 0$	$y_i^{\text{fraud}} = 1$
`age`	0.00 %	0.12 %	0.00 %	0.18 %
`responsibillityCode`	50.11 %	50.41 %	50.11 %	49.04 %
`claimAge`	3.78 %	1.39 %	3.78 %	1.58 %
`lastClaim`	29.37 %	42.39 %	29.38 %	44.13 %
`amount`	3.82 %	0.93 %	3.81 %	1.05 %

The table reveals that the feature `responsibilityCode` is absent for approximately half of the claims for both target values in both D^{known} and D^{fraud}. As this pertains to a categorical variable, we address this issue by introducing an additional category 'missing', allowing the feature to assume one of the values *at fault*, *shared responsibility*, *full right*, *unknown*, and *missing*. Note that we purposefully distinguish *missing* from *unknown*. The designation of *missing* could indicate that the value was absent due to the claims handler either forgetting to enter the correct value or the value being irrelevant for the specific claim type. Meanwhile, *unknown* explicitly indicates that the policyholder's responsibility was not or could not be determined.

The `lastClaim` feature is missing for a slightly smaller but still noteworthy percentage of claims. This concerns informative missingness, where the absence of a value signifies that the policyholder had not previously submitted a claim. To address this issue, we adopt the missing indicator method [138]. Consequently, we impute the missing values with the mean of all observed `lastClaim` values and introduce an indicator feature. This indicator feature takes on a value of 1 when the `lastClaim` value was originally missing and 0 otherwise.

While the missing indicator method can enhance the predictive performance of linear models and neural networks in the presence of informative missing values, it is likely to provide marginal additional information for tree-based methods [138], such as random forest models. As a result, this approach is likely to be advantageous for our final logistic regression classifier but unlikely to yield substantial benefits for our random forest models used for feature importance ranking. We accept this trade-off to avoid introducing major complexity in the preprocessing stage by adopting more involved imputation methods.

The `age`, `claimAge`, and `amount` features are missing for very small percentages of claims (0.00% to 3.82% per combination of data set and target value). For that reason, we choose to discard all records with missing `age`, `claimAge`, and/or `amount` features, thereby adopting a complete case analysis approach, also referred to as list-wise deletion [139]. The complete case analysis approach could introduce bias whenever data are not missing completely at random (MCAR) [140]. However, we consider the percentages of claims with missing features to be sufficiently low to accept this risk. This enables us to avoid entering into another time-intensive and error-prone data transformation trajectory.

Generating the data sets

Having eliminated any missing features, we construct data sets D^{known} and D^{fraud} by aggregating all labelled claims from the most recent year and supplementing them with a random sample of 20,000 unlabelled claims from the same period. This approach aligns with that reported in Óskarsdóttir et al. [19]. Then, utilising scikit-learn's [141] `train_test_split`, we split both data sets into training sets (70% of the observations) and test sets (remaining 30%) in a stratified manner, ensuring that the class distribution in both the training and test sets mirrors that of the original data sets D^{known} or D^{fraud}. Notably, positive samples in D^{known} constitute approximately 4.06% of all samples in the data set, whereas those in D^{fraud} constitute around 2.69%.

For oversampling the data sets, we adopt a different strategy than the one adopted by Óskarsdóttir et al. [19] to address potential data leakage and overoptimistic results. More specifically, consider the description of the oversampling approach employed in their work:

> We use the SMOTE sampling technique to over sample the minority class and under sample the majority class (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). As such, the ratio of the minority class in each sampled data set is increased to 15%. We use these sampled data sets to evaluate the features' importance using random forests. We first tune the hyperparameters of the random forests using 10-fold cross-validation on the training sets.

This depiction suggests that oversampling was applied to the training data sets before employing the ten-fold cross validation approach, which provides biased results [112]. The practice is prone to introducing data leakage from the training to the validation set. In addition, it results in the validation set's class distribution deviating from a faithful representation of what one would observe in a practical setting, thereby leading to overly optimistic outcomes.

Our approach to resampling and additional preprocessing involves utilising the Python-library 'imbalanced-learn' [142] to construct a classification pipeline that includes column transformations, oversampling, and classification. By conducting ten-fold cross validation with this full pipeline as the estimator, we ensure that oversampling is applied to only the training data in each fold individually, preventing the aforementioned issues.

The column transformation step in the pipeline is composed of imputing numerical features, standardising numerical features, and dummy encoding categorical features. Imputation is applied to the `lastClaim` feature only, employing the approach outlined in Section 4.3.5. For this task, we utilise scikit-learn's `SimpleImputer` with `strategy = 'mean'` and `add_indicator = True`. Standardisation is applied to all numerical features, i.e., all but `responsibilityCode`, by removing the mean and scaling to unit variance. For this purpose, we leverage scikit-learn's `StandardScaler`. Dummy encoding exclusively targets the `responsibilityCode` feature, using scikit-learn's `OneHotEncoder`. To avoid introducing multicollinearity and falling into the dummy variable trap where one variable can be easily predicted with the rest, the `OneHotEncoder` is configured to 'drop' the first category.

For the oversampling step, we adopt the same SMOTE algorithm [47] as used by Óskarsdóttir et al. [19]. We employ the implementation provided by imbalanced-learn and set the `sampling_strategy` to 15/85, equivalent to the ratio of 15% samples of the minority class reported by the original authors. The other parameters are left at their default value. The final estimator in the pipeline is adjusted depending on the task at hand.

5.4.5 Employing Random Forests for Feature Importance Ranking

In alignment with the approach reported in the original work [19], we utilise a ten-fold cross-validated grid-search to find the random forest classifier configuration that yields the best results. To achieve this, we employ scikit-learn's `GridSearchCV` with a parameter grid identical to that specified by Óskarsdóttir et al. [19]. We explore configurations for the total number of features configured for each split and the total number of trees in the forest. The search space for the total number of features considered for each split is

Table 5.6: Random forest model parameters

Parameter	Value
`n_estimators`	Grid search: $[100, 300, 500, 700, 900]]$
`criterion`	gini
`max_depth`	None
`min_samples_split`	2
`min_samples_leaf`	1
`min_weight_fraction_leaf`	0.0
`max_features`	Grid search: $[1, 3, \ldots, \text{NoF}]$
`max_leaf_nodes`	None
`min_impurity_decrease`	0.0
`bootstrap`	True
`oob_score`	False
`warm_start`	False
`class_weight`	None
`ccp_alpha`	0.0
`max_samples`	None

$[1, 3, \ldots, \text{NoF}]$, where 'NoF' denotes the total number of available features. The search space for the number of trees is $[100, 300, 500, 700, 900]$.

Authors in [19] did not specify the metric they consider for choosing the best configuration. In this work, we choose to optimise the AUC-PR due to its suitability in scenarios involving imbalanced class distributions, as described in Section 4.6.2. To compute this metric, we employ scikit-learn's `average_precision` function.

Our estimator for the grid search is the pipeline described in Section 5.4.4, augmented with a scikit-learn `RandomForestClassifier`. Aside from the parameters explored through the grid search, this classifier is constructed using the default configuration, as illustrated in Table 5.6. The grid search is conducted for both data sets D^{known} and D^{fraud} independently, for four distinct feature sets: $X^{\text{intr}}, X^{\text{score}}, X^{\text{nbh}}$, and $X^{\text{all}}(= X^{\text{intr}} \oplus X^{\text{score}} \oplus X^{\text{nbh}})$.

Following the grid search, the parameters yielding the best AUC-PR are utilised to construct new random forest classifiers, each trained on the full training data set. This process is repeated for each data set and each feature set independently. The resultant classification models automatically store the Gini importance of each observed feature. For each combination of data and feature set, we extract these importances and report on the results.

Then, in line with Óskarsdóttir et al. [19], we construct logistic regression models to assess the performance impact of sequentially adding a new feature to the model in descending order of importance. For that purpose, we augment the preprocessing steps described in Section 5.4.4 with scikit-learn's `SelectFromModel` feature selector and a `LogisticRegression` classifier. A pipeline is created for each value n in $[1, 2, 3, \ldots, \text{NoF}]$, each time configuring the feature selector to select only the n features with the highest Gini importance from the pre-trained random forest models. Only these selected features are then forwarded to the logistic regression models. The logistic regression models are constructed with parameters outlined in Table 5.7. These mirror default settings, except for the removal of the penalty to increase the likelihood of resembling the classifier used in the original work and an increase in the maximum number of iterations to yield convergence.

The entire process is encapsulated in a ten-fold cross validation procedure, implemented using scikit-learn's `StratifiedKFold`. The AUC-PR, AUC-ROC and TDL of the logistic regression models, trained on the train fold and evaluated on the validation fold, are aggregated in terms of mean and standard deviation for each number of n features and

TABLE 5.7: Scikit-learn logistic regression classifier parameters

Parameter	Value
`penalty`	None
`dual`	False
`tol`	0.0001
`C`	1.0
`fit_intercept`	True
`intercept_scaling`	1
`class_weight`	None
`solver`	lbfgs
`max_iter`	500
`multi_class`	ovr
`warm_start`	False

each data set and feature set. These aggregated results are then visualised for each data set and feature set individually.

5.4.6 Constructing the Final Classifier

The final classification model is again built using the aforementioned preprocessing pipeline and a logistic regression classifier. However, in this instance, the classifier is constructed using the `Logit` implementation from statsmodels [143]. This approach facilitates retrieval of the same types of model features as reported in [19], specifically feature coefficients and p-values, enabling a comparison of outcomes between the two studies. To maintain a level of consistency between the two classification model implementations, we optimise statsmodels' classifier using a similar optimisation algorithm to the limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) [144] optimisation algorithm used by default in scikit-learn's logistic regression classifier (Section 5.4.5). This time, the original non-limited-memory version of the algorithm was used, addressing technical difficulties.

For each data set and feature set, the logistic regression classifier is provided the full range of features and training samples. Subsequently, the performance of the classification model is evaluated on the test set.

5.5 Incorporating Time-Weighting

The incorporation of time-weighted fraud influence following the methodology presented in Section 4.4 involves multiplying the baseline query vector by $e^{-\lambda h}$, where λ represents the decay constant and h denotes the time elapsed since the claim's registration date (see Section 4.4). Given that this study employs historical data, it is inappropriate for h to signify the time between the claim registration date and the date of the analysis. Instead, we define h as as the time from the claim registration date to the last claim registration date considered in this study. For claims registered at the beginning of the last year—which is the beginning of the period considered by the supervised analytical models—this results in a one-year discrepancy compared to the calculation in a practical deployment scenario. We acknowledge this limitation to avoid the technical complexities associated with recomputing h, the query vector, and all fraud scores for every evaluated claim.

In determining a suitable value for the decay constant, we examine the duration that an instance of fraud remains recorded in national fraud registers. We observe that an insurance fraud register deployed in the UK reports on a maximum registration period of five years [145], whereas a widely adopted fraud register in the Netherlands reports a maximum period of eight years [146]. In this study, we consider the eight-year period and set the fraud influence of an eight-year-old claim to be 50% of the influence of a recent claim. This reflects our intention to have time-weighting impact the ranking without completely diminishing the influence of historical fraud, as a value close to 0% would, given the already limited knowledge of fraudulent claims. Thus, with h representing a period in days, we define

$$e^{-\lambda h} = e^{-\lambda 365.25 \times 8} = 0.5 \tag{5.2}$$

such that

$$\lambda = -\frac{\ln 0.5}{365.25 \times 8} \approx 0.000237. \tag{5.3}$$

We use this value to construct a new query vector and then use the BiRank algorithm and extract network features utilising the same approach described in Section 5.4.3.

5.6 Incorporating Shared Resources

To incorporate shared resources in line with the methodology presented in Section 4.5, we first collect relevant information for all parties present in the network. Using this information, we generate formatted strings resembling identifiers for each specific resource. The creation of resource identifier strings is contingent upon the availability of sufficient relevant information to mitigate coincidental resource sharing based on insufficiently granular resources. For example, we refrain from creating address resource strings whenever only partial address information is available to prevent establishing relations based on matching street names alone.

Utilising these resource strings, we construct an edge list composed of all edges between parties and their associated resources. Subsequently, we remove any resources connected to a single party only, as these do not contribute relevant information to the model. From this edge list, we construct a party–resource adjacency matrix.

For the construction of a party–resource adjacency matrix, we have to consider that the number of rows in W_{pr} should match the number of rows in D_p, which should also match the number of rows in W_{pc}. This can be derived from the equations for the values of S_{cp} and S_{pc} in Equations 4.24 and 4.25. Moreover, it is crucial that the row indices correspond, meaning that the same parties assume identical index numbers across all index numbers. Both requirements are not explicitly guaranteed by default, in part because not every party in the network is necessarily related to any shared resource and therefore included in the corresponding weight matrix. To address this issue, we augment the party–resource edge list. For all parties included in the claim–party edge list but missing in the party–resource edge list, we extend the party–resource edge list by incorporating a zero-weight relation from the respective party to a resource already present in the party–resource edge list. Then, rows in the sparse W_{pr} matrix are sorted equivalently to the columns in W_{cp}.

The claim–party and party–resource adjacency matrices are provided as input to the simplified iterative tripartite ranking algorithm presented in Algorithm 3, implemented by adapting the BiRank implementation described in Section 5.4.2. We set α_{cp} to 0.85, consistent with the hyper-parameter configuration adopted for the baseline model. This hyper-parameter adjusts the influence of the query vector and associated parties on claim nodes. In addition, we configure $\alpha_{pc} = 1$ and $\alpha_{pr} = 1$ so that a connection from a shared resource to a party holds the same value as a connection from a claim to a party. This was empirically validated to yield intuitive scores in various scenarios, though a more thorough analysis and investigation into how optimisation of the hyper-parameters influences the performance of the model could be an interesting avenue for future research. For this research, this is out-of-scope.

We consider a tripartite graph only for the computation of fraud scores. Consequently, after obtaining the fraud scores from the tripartite network, the subsequent graph construction for network feature extraction again assumes only a bipartite graph of claims and parties. This facilitates reuse of the implementation built for the baseline model (Section 5.4.3), enabled by not including features explicitly reliant on the inclusion of shared resources in the graph.

5.7 Fraud Expert Evaluation of Unlabelled Claims

From each model, we collect the top-k claims from the test set for D^{fraud} that were assigned the highest probability of fraud by the model. Subsequently, these claims are submitted to fraud experts for evaluation. Each claim undergoes assessment by a single fraud expert, who determines whether they would have recommended the claim to be flagged by the model. They explicitly refrain from asserting whether the claim is genuinely fraudulent. As a result, this evaluation yields no definitive information about the model's performance in detecting fraud, but instead offers insights into its ability to identify potentially suspicious claims.

The fraud experts are provided only claim IDs, allowing them to access claim details within the insurance company's information systems. They receive no information on which model the claim was retrieved from, nor are they given access to the specific set of features utilised in this model or the graphs surrounding the respective claims. This eliminates potential bias, ensuring an equivalent evaluation of claims from each model, though it also deprives the experts of contextual information that could have been relevant to the classification.

5.8 Conclusion

Concluding this chapter, we have presented information about the construction and characteristics of the employed data set, along with insights into our analysis of homophily and the implementation and evaluation of the baseline and adapted models. Having established these details, we transition into the next chapter for an exploration of the results of our work.

Chapter 6

Results

Having previously presented our methodology and details concerning the experimental setup that is employed, the current chapter presents insights into the results that were achieved. Whenever relevant, the results are presented in a way that facilitates a comparison to the findings reported by authors of the original work [19]. However, the actual comparison is presented in Chapter 7 instead.

Section 6.1 presents the results of our investigation into empirical evidence for homophily in our data. Then, Section 6.2 sheds light on the results attained in relation to feature importance ranking. Lastly, Section 6.3 provides insight into the performance of both the baseline models and the adapted models on a held-out test set, along with a summary of the baseline models' characteristics.

6.1 Homophily Assumption

Following the methodology and implementation outlined in Sections 4.2 and 5.3, we analysed claims in the second- and fourth-order neighbourhoods of labeled claims to yield the statistics necessary to examine whether our data establishes any evidence for the homophily assumption. The results of this assessment are detailed in Table 6.1 and elucidated hereafter.

The table shows that fraudulent claims exhibit a higher average relative frequency of other fraudulent claims in their second-order neighbourhoods than non-fraudulent claims. This observation holds true when computing the average relative frequency when compared to *all* claims (4.121% versus 1.781%) and when compared to all *investigated* (i.e., known) claims in the second-order neighbourhood. A similar observation is made for fourth-order neighbourhoods, albeit with less pronounced differences of 0.336% versus 0.313% and 69.217% versus 68.663%.

Table 6.1: Average relative frequency of fraudulent and non-fraudulent claims in the neighbourhoods of known claims.

		Avg. rel. freq. (all)		Avg. rel. freq. (investigated)	
Neighbourhood	Label	Non-fraudulent	Fraudulent	Non-fraudulent	Fraudulent
Second-order	Non-fraudulent	1.601 %	1.783 %	35.083 %	64.917 %
	Fraudulent	0.828 %	4.121 %	27.419 %	72.581 %
Fourth-order	Non-fraudulent	0.109 %	0.313 %	31.337 %	68.663 %
	Fraudulent	0.146 %	0.336 %	30.783 %	69.217 %

To assess the significance of these differences, we conducted significance tests, beginning with an F-test on equality of variances to determine whether to use Student's T-test or Welch's T-test. For second-order neighbourhoods, the F-score for the observed average relative frequencies of fraud compared to *all* claims was 3.1818, with variances of 0.025677 and 0.008070 for the neighbourhoods of fraudulent claims and non-fraudulent claims, respectively. This yielded a p-value of approximately 1.11×10^{-16}, resulting in the rejection of the null hypothesis of equal variances. A similar outcome was obtained when considering investigated claims only, with an F-score of 1.25994 and a p-value of approximately 1.57×10^{-6}. Consequently, we adopted Welch's T-test for both significance tests concerning the second-order neighbourhoods.

Subsequent analysis of the significance of observed differences in average relative frequency of fraudulent claims among all claims in the second-order neighbourhoods using Welch's T-test yielded a p-value of $\approx 5.888 \times 10^{-14}$. Considering the same frequency among investigated claims instead, the p-value was 4.613×10^{-13}. The fact that both p-values are near zero suggests a significant difference between fraudulent and non-fraudulent claims in terms of the average relative frequency of fraudulent claims in their second-order neighbourhoods. This significance is suggested both when considering *all* claims in the neighbourhood and when considering *investigated* claims only.

Exploring fourth-order neighbourhoods, we observed unequal variances in average relative frequencies of fraudulent claims among all claims but equal variances among investigated claims. The associated F-scores were 4.0267 and 1.0569 with p-values of approximately 0 and 0.1015, respectively. We therefore adopt Welch's T-test for the ratio among all claims, and Student's T-test for the ratio among investigated claims, revealing p-values of 0.5885 and 0.1702. This suggests insignificant differences between fraudulent and non-fraudulent claims when comparing the average relative frequency of fraudulent claims in their fourth-order neighbourhoods, whether considering all claims or investigated claims.

6.2 Feature Importance

Following the analysis of the network, we performed a ten-fold cross validated grid search on the training data to discover the best configuration for random forest classification models, based on the baseline model. The goal was to uncover the set of hyper-parameters that provides the best AUC-PR, both for each group of features (X^{intr}, X^{score}, X^{nbh}, and X^{all}) and each data set ($\mathcal{D}^{\text{known}}$ and $\mathcal{D}^{\text{fraud}}$) to enable a comparison to the results in the original work [19]. In Appendix A, we present the performance achieved for each explored combination of hyperparameters. Meanwhile, the optimal parameters are displayed in Table 6.2. The table reveals some variation between data sets and feature sets in terms of the parameters that yielded the best result, but a configuration with the maximum considered number of trees proved best for six out of eight cases.

We chose the models that achieved the best results and collected the normalised Gini importances corresponding to the features that were provided to the model as input. These Gini importances are presented in Section 6.2.1. Then, Section 6.2.2 presents ten-fold cross-validated results achieved by logistic regression classifiers that were constructed using the top n most important features in the best random forest models for varying numbers of n.

6.2.1 Gini Feature Importance

The Gini importances are presented in Figures 6.1 to 6.4, employing a visualisation format consistent with the original work [19]. This facilitates a comparison between the results

Table 6.2: Optimal hyper-parameters for our random forest models

Data set	Feature group	No. of trees	Features per split
D^{fraud}	intrinsic	900	5
	score	300	5
	neighbourhood	900	5
	all	900	3
D^{known}	intrinsic	900	7
	score	900	1
	neighbourhood	900	5
	all	700	9

achieved by our baseline model and the results presented in their study. It is worth noting that the original work exhibits inconsistency in reported feature names, including a typographical error in their seventh figure. In this report, we adhere to feature names consistent with those in Tables 4.1 to 4.2. Accordingly, when comparing feature names in our figures to theirs, `responsibilityCode` matches `clResp`, `age` matches `pAge`, and `n1.q1` matches `n1.1q`. Additionally, for reasons unbeknown to us, the feature importance figures in the original work do not present all features considered in their research. In this report, we do report on the importance of all features.

Note that, for both `lastClaim` and `responsibilityCode`, Figures 6.1 and 6.4 reveal multiple suffixed features. Feature `lastClaim_x` represents the 'missing indicator' that was previously detailed in Section 5.4.4. Meanwhile, features `responsibilityCode_1`, `responsibilityCode_2`, `responsibilityCode_3`, and `responsibilityCode_x` match the features generated by dummy-encoding the original `responsibilityCode` feature, as described in Section 5.4.4.

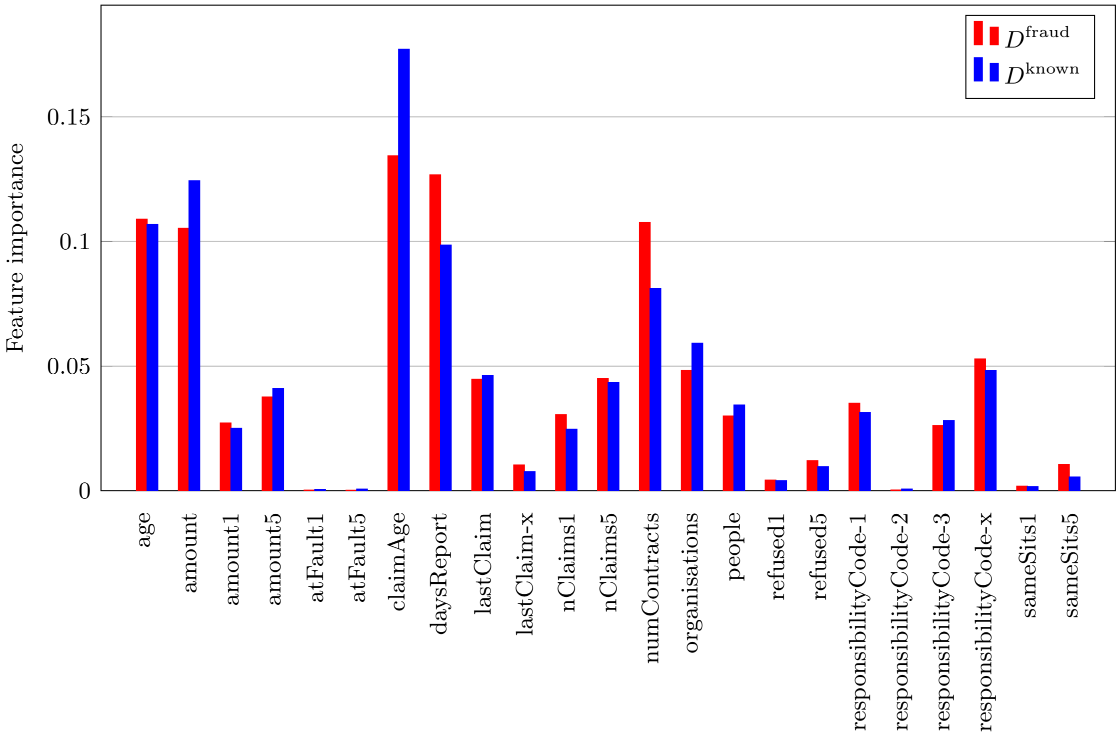


Figure 6.1: Gini feature importance of intrinsic features

Examining Figure 6.1, we observe that the overall ranking of features for D^{known} and D^{fraud} is very similar. We extract that the top five intrinsic features with the highest importance include `claimAge`, `amount`, `age`, `daysReport`, and `numContracts` for both data sets, and notice that features pertaining to the policyholder's responsibility in historical claims exhibit noticeably small feature importance. More specifically, this concerns `atFault1`, `atFault5`, `sameSits1`, and `sameSits5`.

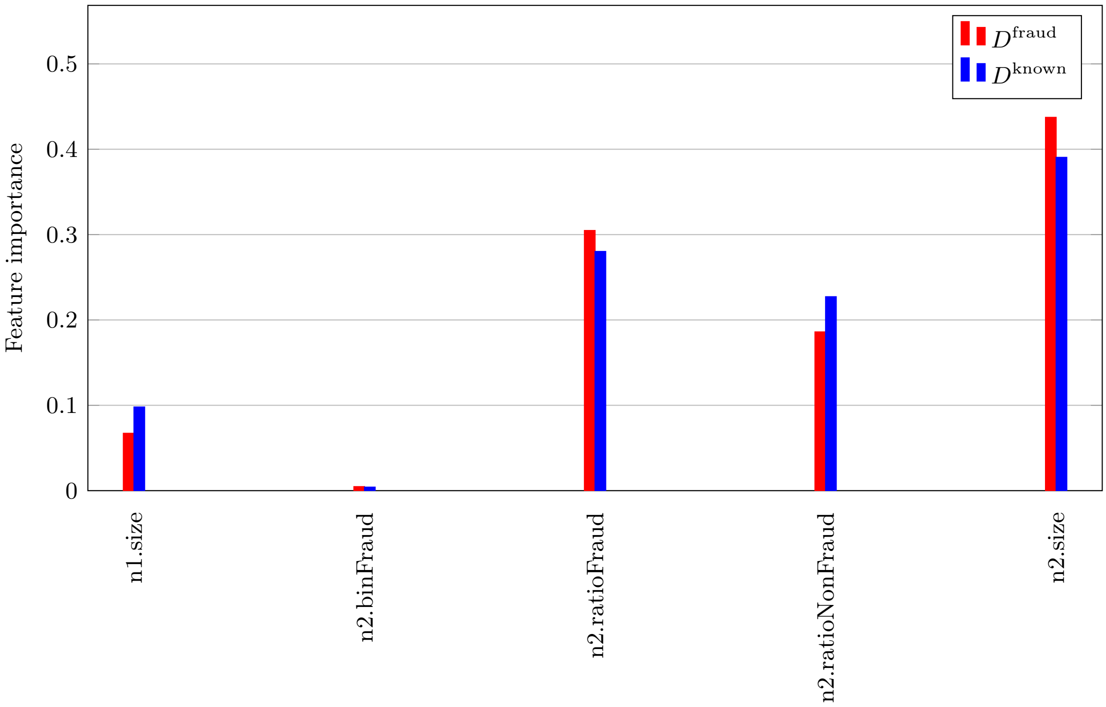


Figure 6.2: Gini feature importance of neighbourhood features

Moving on to the neighbourhood features in Figure 6.2, we again observe that the importance of features for D^{known} and D^{fraud} are similar. For both data sets, the figure reveals the highest Gini importance attributed to `n1.size`, followed by `n2.ratioFraud`, `n2.ratioNonFraud`, and `n1.size`. The `n2.binFraud` feature is attributed a Gini importance nearing zero for both types of classification.

Figure 6.3 depicts the Gini importance of score features. It reveals a mostly uniform distribution of Gini importance over features for D^{known}, with slight variation for D^{fraud}. For the latter data set, the largest Gini importance is achieved by `n1.max`, closely followed by `n2.max` and `n2.med`.

Considering all features in Figure 6.4 instead, we observe that the top five features with the highest Gini importance for D^{known} are `claimAge`, `amount`, `age`, `daysReport`, and `responsibilityCode_x`. For D^{fraud}, this list is composed of the same features, but in a slightly different order: `claimAge`, `age`, `daysReport`, `amount`, and `responsibilityCode_x`. This is consistent with our earlier observation that the importance of features to D^{known} and D^{fraud} are relatively similar. Considering the features with the lowest Gini importance in the figure, we make the same observations as those for the independent groups of features. This means that both features related to the policyholder's responsibility in historical claims (e.g. `atFault1` and `atFault5`) and `n2.binFraud` exhibit very low importance.

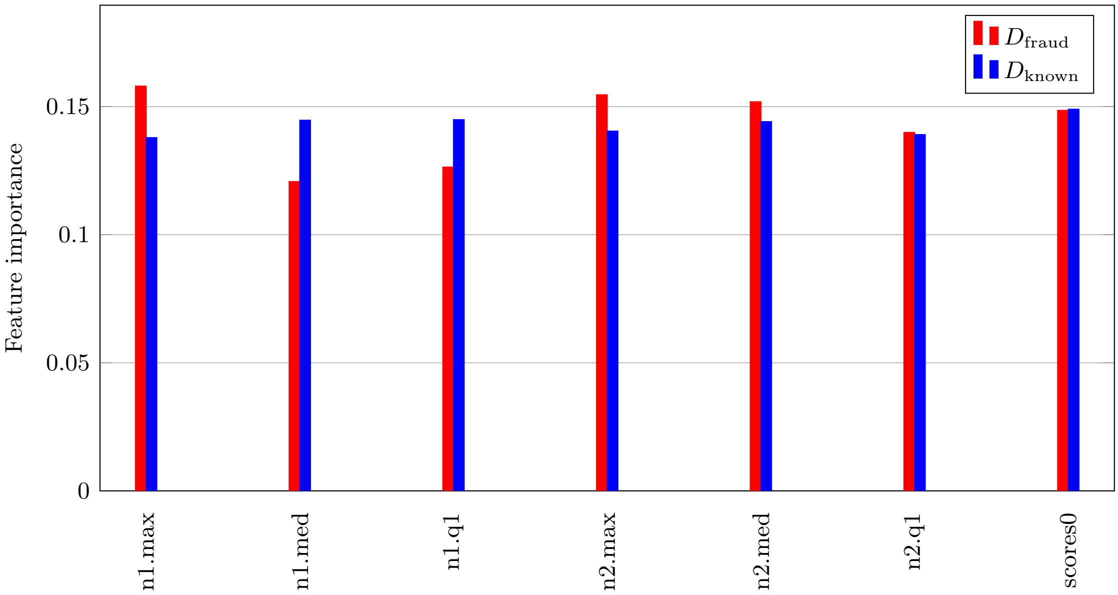


FIGURE 6.3: Gini feature importance of score features

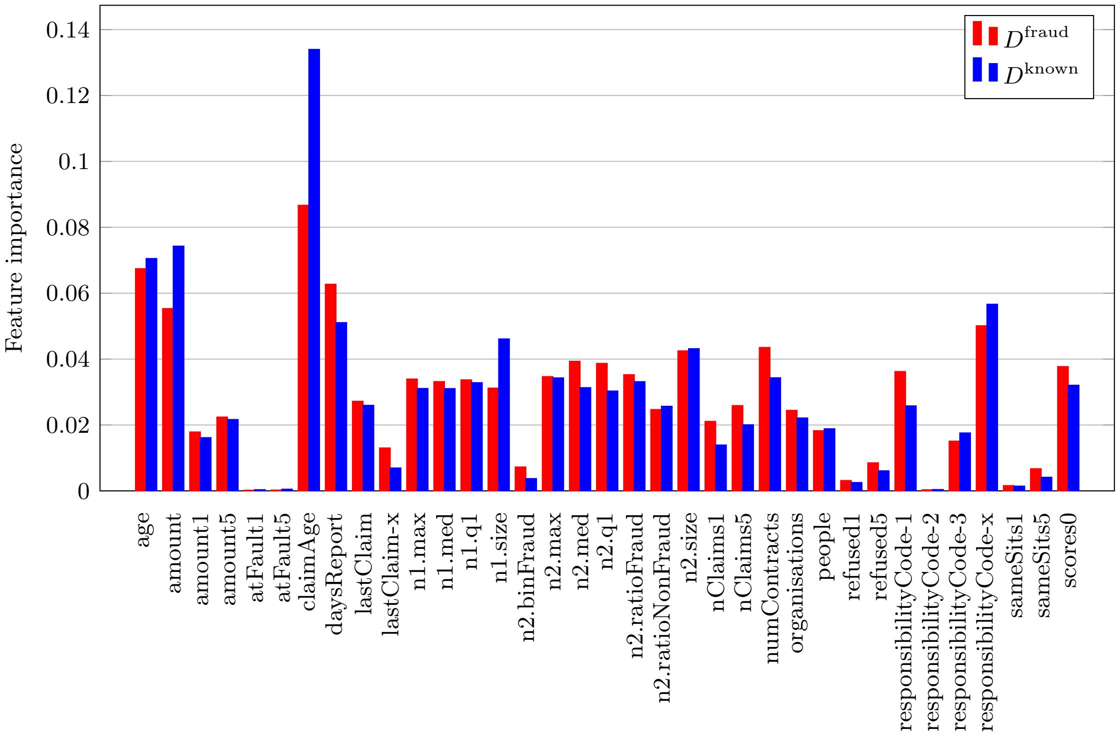


FIGURE 6.4: Gini feature importance of all features

6.2.2 Sequential Feature Addition

Following the evaluation of individual features' Gini importance, we proceed to evaluate the performance of a logistic regression classifier constructed using the top n features with the highest Gini importance. We considered all values of n in the range of $[1, 2, \ldots, \text{NoF}]$, where 'NoF' indicates the total number of features in the feature set. The results were obtained through ten-fold cross validation on the training sets and presented in Figure 6.5. Note that the different categories for the categorical feature `responsibilityCode` and the missing indicator for `lastClaim` are presented as distinct features. Consequently, the maximum number of features in this figure deviates from the total number of features presented in Tables 4.1, 4.3 and 4.2.

Upon examining the plots, we first observe that there was similar performance for D^{known} and D^{fraud} when considering AUC-ROC, whereas the AUC-PR and TDL suggest superior performance was achieved when classifying D^{known}. Furthermore, we observe that the use of only intrinsic features yielded improved performance when compared to using only score or neighbourhood features, regardless of the metric considered (AUC-ROC, AUC-PR, or TDL) and whether considering D^{fraud} or D^{known}. Utilising all features is shown to have further enhanced performance across all three metrics, albeit sometimes marginally. For both data sets, our observed maximum differences in AUC-ROC, AUC-PR, and TDL between using all features and using intrinsic features are in the range of 0.05, 0.05, and 0.5, respectively.

Analysing the absolute values obtained for the three performance metrics, we observe maximum values of 0.79, 0.25, and 4.69 for AUC-ROC, AUC-PR, and TDL, respectively, when considering D^{known}. For D^{fraud}, these values are 0.77, 0.15, and 4.26 instead. Investigating when maximum performance is achieved, we observe that the classification performance of the model remains relatively consistent past $n = 14$ features for both D^{known} and D^{fraud}.

Figure B.1 in Appendix B also presents the cross-validation standard deviations corresponding to the mean results presented in Figure 6.5. The figure reveals that the difference in mean performance between intrinsic features and all features frequently lies within the range of one standard deviation for all three performance metrics. In making a comparison between score and neighbourhood features, a similar observation is made. The difference is more pronounced when comparing the mean performance of intrinsic or all features to the mean performance of score or neighbourhood features. In these comparisons, differences larger than one standard deviation are observed, except for D^{fraud} with AUC-PR.

6.3 Evaluation on Test Set

The previous sections presented Gini importance of features in the baseline model and the impact of including different numbers of features on the model's ten-fold cross-validated classification performance on the training data. In this section, we present the results of evaluating both the baseline model, the time-weighted model, and the shared resources model on test data instead. This facilitates a comparison between our final baseline model and the model construed by Óskarsdóttir et al. [19], as well as an evaluation of the impact of the adaptations on the classification performance of our model.

In Section 6.3.1, we first present each model's performance in correctly classifying claims based on known labels. Then, Section 6.3.2 sheds light on the models' performance in retrieving interesting, previously unlabelled claims.

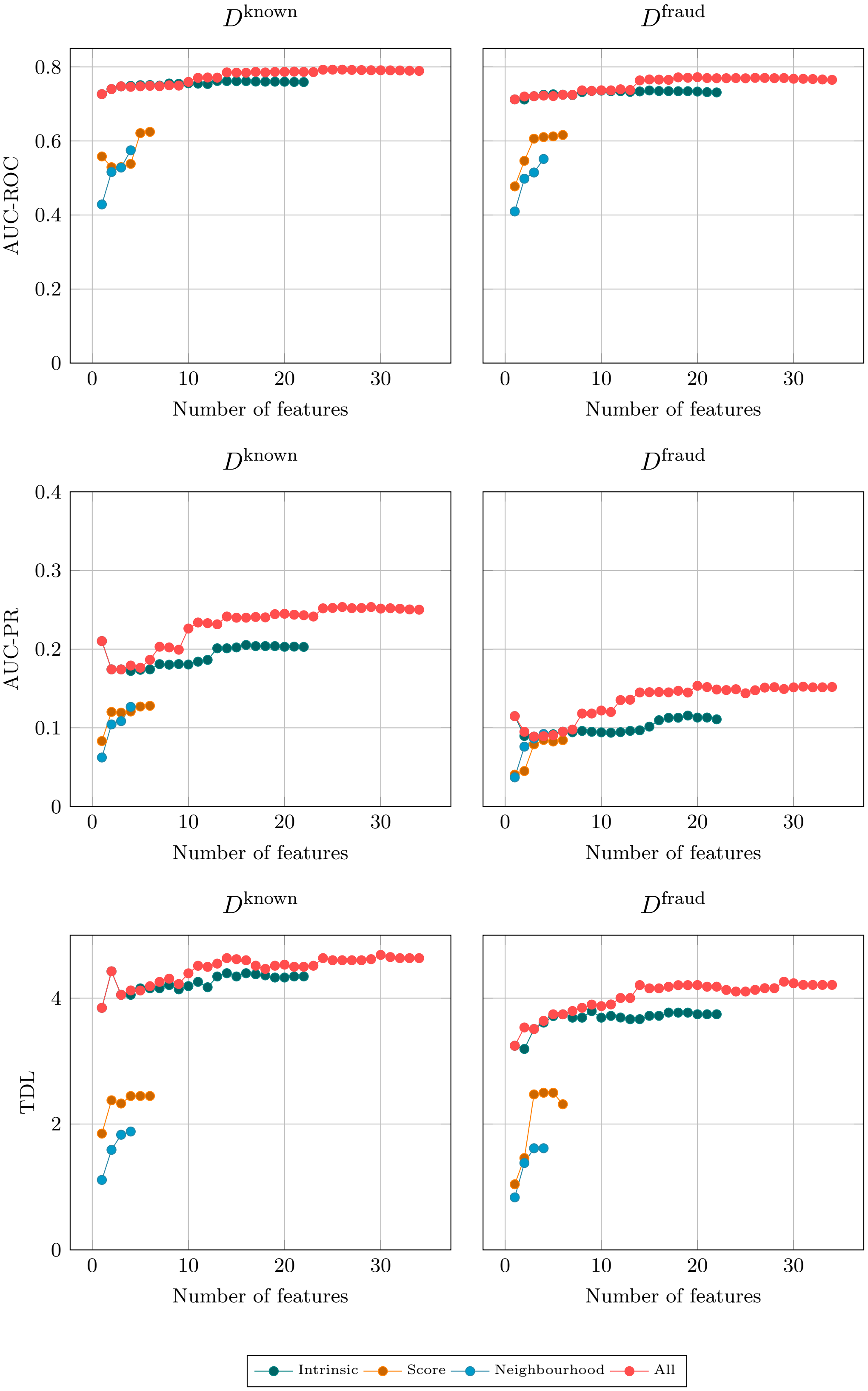


FIGURE 6.5: Ten-fold cross-validated performance of LR models for sequential feature addition

Table 6.3: Model performance for different feature sets on each test data set

		D^{known}			D^{fraud}		
Model	Features	AUC-ROC	AUC-PR	TDL	AUC-ROC	AUC-PR	TDL
Baseline	Intrinsic	0.767	0.207	4.445	0.735	0.105	3.985
	Score	0.617	0.084	1.810	0.618	0.058	1.903
	Neighbourhood	0.684	0.123	3.344	0.668	0.117	2.617
	All	0.803	0.255	4.839	0.777	0.153	4.044
Time-weighted	Intrinsic	0.767	0.207	4.445	0.735	0.105	3.985
	Score	0.622	0.089	2.006	0.603	0.057	1.665
	Neighbourhood	0.684	0.123	3.344	0.668	0.117	2.617
	All	0.804	0.257	4.839	0.778	0.157	4.104
Shared resources	Intrinsic	0.767	0.207	4.445	0.735	0.105	3.985
	Score	0.639	0.105	2.321	0.633	0.065	2.312
	Neighbourhood	0.684	0.123	3.344	0.668	0.117	2.617
	All	0.805	0.260	4.760	0.779	0.156	4.223

6.3.1 Evaluation on Labelled Data

Table 6.3 presents the performance of each model and each feature set in classifying the test set. In Figure 6.6, the same data is depicted graphically. The model adaptations had no influence on the values in X^{intr} and X^{nbh}. Consequently, the results of the baseline, time-weighted and shared resources models were equivalent for these specific feature sets.

Comparing the performance of each model, we observe that the differences are small, both for D^{known} and D^{fraud}. For example, when considering D^{fraud} with the set of all features, we observe differences in AUC-ROC, AUC-PR, and TDL of at most 0.02, 0.004 and 0.179, respectively. These differences are slightly larger for the score features, with values of 0.030, 0.08 and 0.647, respectively. Comparing the classification of D^{known} versus the classification of D^{fraud} instead, we observe that all models perform better at distinguishing known claims from unknown claims than they do at distinguishing fraudulent claims from non-fraudulent or unknown claims, independent of the considered feature set and performance metric.

Moving on to a comparison based on the feature set that was employed, we observe that the set of all features yielded improved performance when compared to considering intrinsic, score, or neighbourhood features only. This observation is consistent across all three models and all three performance metrics. Considering a single feature set, the table reveals that the intrinsic features were most valuable in correctly classifying claims as known/unknown or fraudulent/non-fraudulent.

Having evaluated the AUC-ROC and AUC-PR for the different model, feature set, and data set configurations, we present the AUC and PR curves that were summarised by these metrics in Figures 6.7 and 6.8. These curves share the same story as their corresponding summary metrics while also revealing the best combination of performance metrics that could be achieved for each configuration. For example, Figure 6.8 reveals that, when considering D^{fraud} and the baseline model with all features, achieving a recall of 0.2 on the test set is accompanied by a precision of 0.2. A higher recall can be attained, but at the cost of lower precision, and vice versa.

6.3.2 Fraud Expert Evaluation on Unlabelled Data

Whereas the previous results shed light on models' performance in correctly classifying claims based on known labels, Table 6.4 presents the results of fraud experts' evaluation

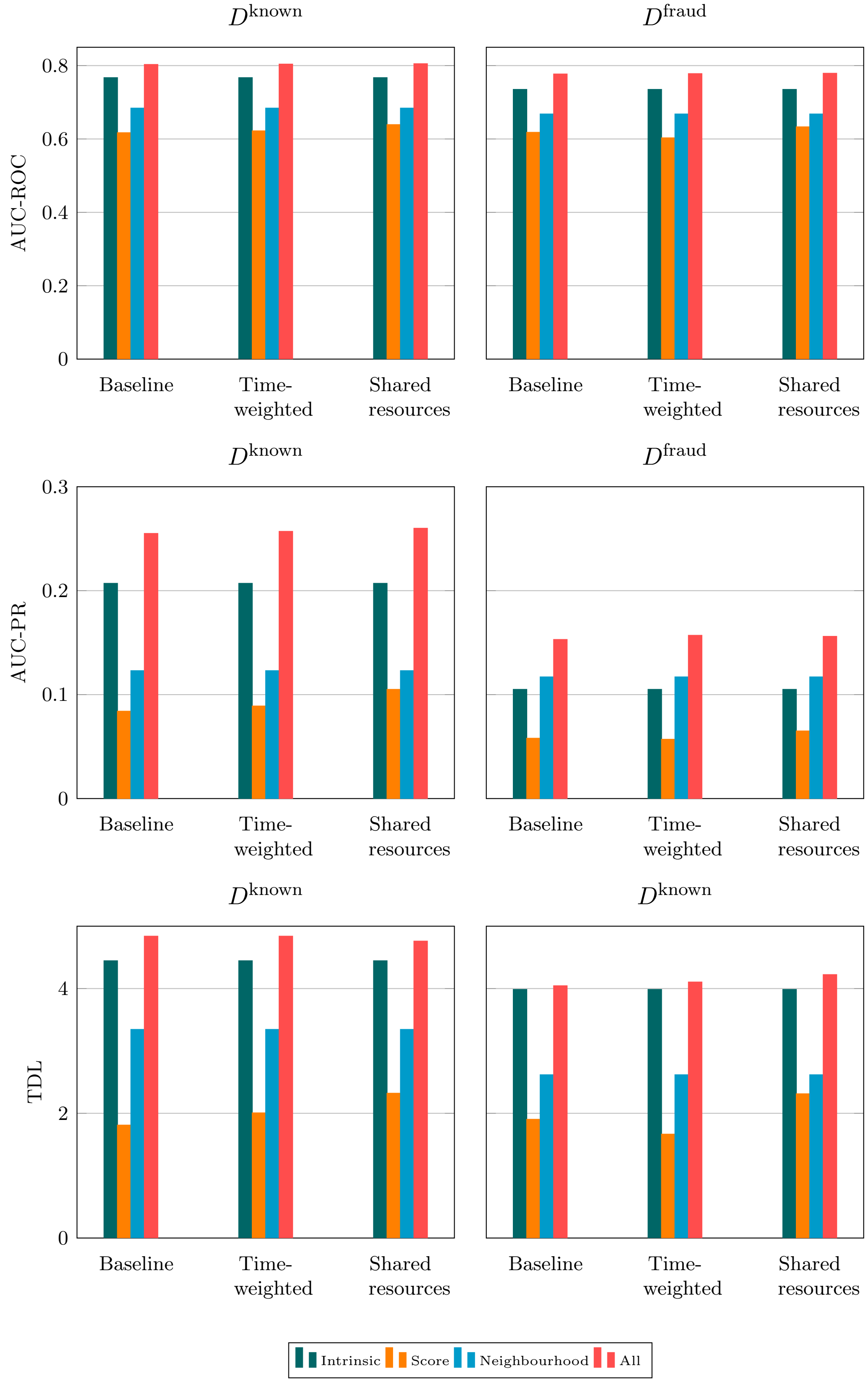


FIGURE 6.6: Test set evaluation results

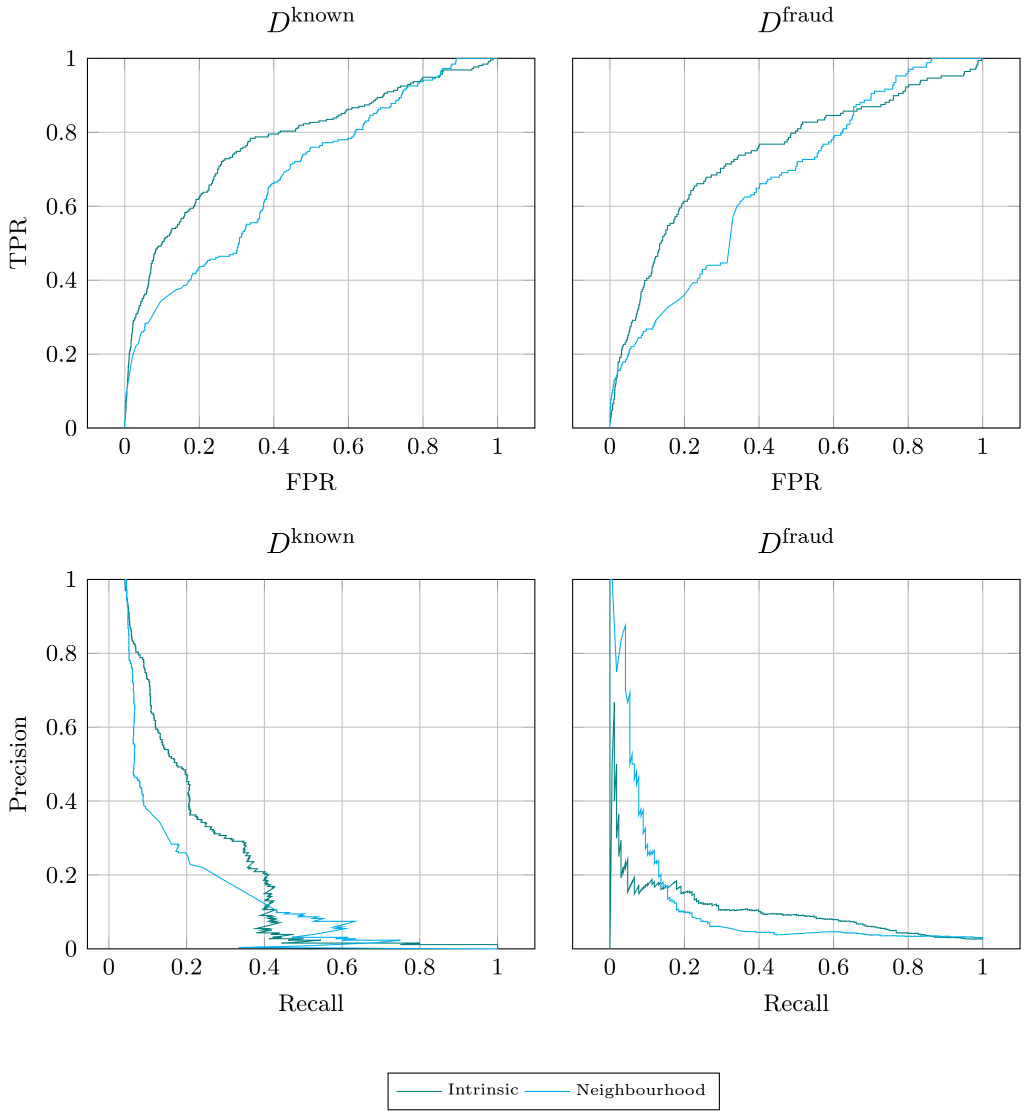


Figure 6.7: AUC and PR curves using intrinsic or neighbourhood features

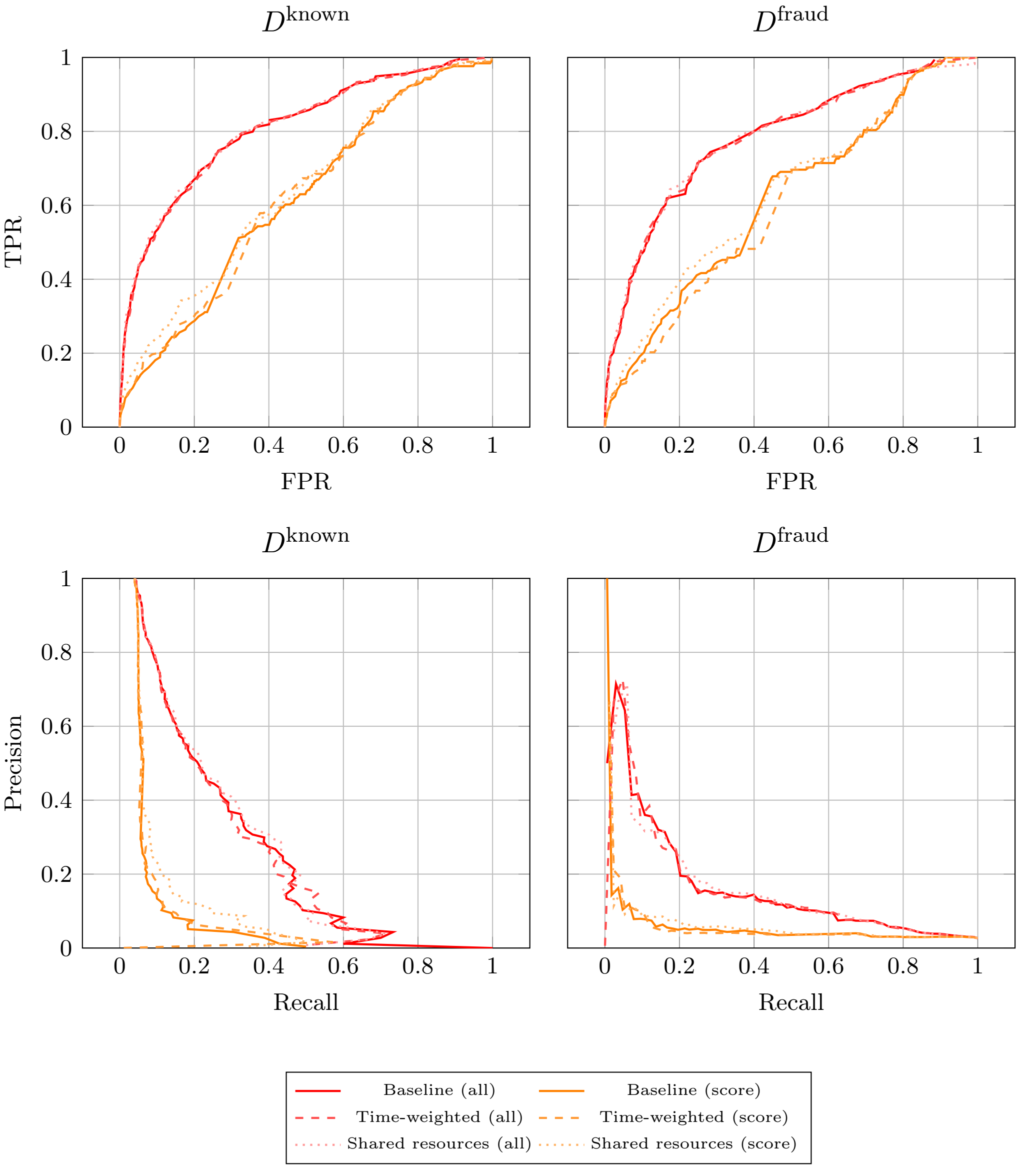


FIGURE 6.8: AUC and PR Curves using all features or score features

Table 6.4: Fraud experts' claim evaluation results

Feature set	Model	Evaluated	Interesting	Precision-at-k
Intrinsic	All	20	11	0.550
Score	Baseline	20	16	0.800
	Time-weighted	20	16	0.800
	Shared resources	20	14	0.700
Neighbourhood	All	20	15	0.750
All	Baseline	20	12	0.600
	Time-weighted	20	13	0.650
	Shared resources	20	12	0.600

of the top-20 *unlabelled* claims with the highest predicted probability of fraud, conducted in line with the experimental setup described in Section 5.7. For each combination of model and feature set, we report on both the number of newly evaluated claims and the number of claims subsequently labelled 'interesting', together facilitating the computation of a 'precision-at-k'.

The table shows that classification based on intrinsic features alone yielded the lowest percentage of claims deemed interesting (55.0%), followed by classification based on all features from the baseline or shared resources model (60.0%) and all features from the time-weighted model (65.0%). The highest percentage was achieved by classification based on score features from the baseline model and time-weighted models (80.0%), closely followed by classification based on neighbourhood features alone (75.0%).

6.3.3 Model Summaries

For an understanding of how the logistic regression models determined each claim's classification, the models' summaries were extracted from the statsmodels classifiers described in Section 5.4.6. Tables 6.5 to 6.12 present these summaries for the baseline logistic regression models corresponding to each feature set and data set, facilitating a comparison to the results presented in the original work [19]. For summaries of the adapted models, the reader is referred to Appendix C instead.

In the tables, the variable coefficients in the 'Coef' columns signify the average change in the log odds of the target variable associated with a one-unit increase of the feature. These can be transformed into the average change in odds using the equation e^{Coef}. Positive coefficients signify that an increase in the independent variable yields an increase in model output, i.e., an increase in the predicted probability of the claim being known or fraudulent, assuming all other independent variables remain constant. The opposite is true for negative coefficients. For enhanced interpretability, the tables include only variables with a p-value (column '$P > |z|$') of less than 0.05, indicative of the fact that there exist a significant relationship (at a significance level of 5%) between the variable and the target variable given a null hypothesis that the coefficient would be equivalent to zero.

Considering intrinsic features, Table 6.5 reveals positive coefficients in the D^{known} model for intrinsic features `amount`, `amount1`, `refused1`, `responsibilityCode_3`, and `responsibilityCode_x`. For D^{fraud}, Table 6.6 shows a similar list of variables with positive coefficients. However, compared to the model for D^{known}, `refused1` is replaced with `lastClaim_x`, `nClaims1`, and `sameSits5`. For both D^{known} and D^{fraud}, the tables present significant negative coefficients for `age`, `claimAge`, `lastClaim`, `numContracts`, and `organisations`. For D^{known}, this list is extended with `people` and `responsibilityCode_2`, whereas the list for D^{fraud} additionally includes `nClaims5` and `sameSits1`.

Table 6.5: Summary of baseline LR model for D^{known} with intrinsic features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`Intercept`	-2.4366	0.065	-37.725	0.000
`age`	-0.3502	0.027	-12.813	0.000
`amount`	0.3201	0.021	15.305	0.000
`amount1`	0.0718	0.029	2.504	0.012
`claimAge`	-0.7651	0.034	-22.475	0.000
`lastClaim`	-0.2040	0.040	-5.138	0.000
`numContracts`	-0.1468	0.029	-5.065	0.000
`organisations`	-0.2129	0.025	-8.382	0.000
`people`	-0.1540	0.030	-5.217	0.000
`refused1`	0.0982	0.028	3.543	0.000
`responsibilityCode_2`	-1.0704	0.478	-2.239	0.025
`responsibilityCode_3`	0.8545	0.125	6.842	0.000
`responsibilityCode_x`	0.2635	0.076	3.459	0.001

Table 6.6: Summary of baseline LR model for D^{fraud} with intrinsic features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`age`	-0.2922	0.026	-11.082	0.000
`amount`	0.1606	0.018	9.140	0.000
`amount1`	0.0907	0.022	4.176	0.000
`claimAge`	-0.6619	0.033	-20.076	0.000
`daysReport`	0.0704	0.017	4.122	0.000
`lastClaim`	-0.3049	0.042	-7.187	0.000
`lastClaim_x`	0.0997	0.029	3.454	0.001
`nClaims1`	0.0952	0.040	2.365	0.018
`nClaims5`	-0.1059	0.046	-2.306	0.021
`numContracts`	-0.2686	0.032	-8.375	0.000
`organisations`	-0.0874	0.024	-3.671	0.000
`responsibilityCode_3`	0.9359	0.119	7.875	0.000
`responsibilityCode_x`	0.2918	0.072	4.040	0.000
`sameSits1`	-0.1527	0.030	-5.026	0.000
`sameSits5`	0.1637	0.031	5.360	0.000

Considering score features in Tables 6.7 and 6.8 for D^{known} and D^{fraud} instead, both reveal negative coefficients for `n1.max` and positive coefficients for `scores0`. For D^{known}, the list of variables with a positive coefficient is extended with `n1.q1`. For D^{fraud}, positive coefficients are added for `n2.max` and `n2.q1`, along with a negative coefficient for `n2.med`.

TABLE 6.7: Summary of baseline LR model for D^{known} with score features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`Intercept`	-1.8077	0.023	-77.758	0.000
`n1.max`	-0.4478	0.046	-9.757	0.000
`n1.q1`	0.4231	0.113	3.742	0.000
`scores0`	0.1805	0.035	5.177	0.000

TABLE 6.8: Summary of baseline LR model or D^{fraud} with score features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`Intercept`	-1.8278	0.023	-77.814	0.000
`n1.max`	-0.5383	0.048	-11.266	0.000
`n2.max`	0.0787	0.036	2.171	0.030
`n2.med`	-0.5826	0.296	-1.971	0.049
`n2.q1`	0.8991	0.304	2.956	0.003
`scores0`	0.2377	0.029	8.232	0.000

Moving on to neighbourhood features in Tables 6.9 and 6.10, we observe only two significant features for D^{fraud}: `n2.ratioFraud` with a positive coefficient and `n2.size` with a negative coefficient. The same features with equivalently signed coefficients are observed for D^{known}, along with negatively signed coefficients for `n1.size` and `n2.size` and positively signed coefficients for `n2.ratioFraud` and `n2.ratioNonFraud`.

TABLE 6.9: Summary of baseline LR model for D^{known} with neighbourhood features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`Intercept`	-1.9095	0.028	-68.770	0.000
`n1.size`	-0.142	0.024	-5.912	0.000
`n2.binFraud`	-0.0854	0.024	-3.525	0.000
`n2.ratioFraud`	0.3677	0.039	9.382	0.000
`n2.ratioNonFraud`	0.083	0.029	2.902	0.004
`n2.size`	-0.6197	0.055	-11.194	0.000

Lastly, we use Tables 6.11 and 6.12 to make a comparison between the significant features in the models constructed using the set of all features and the models constructed using only a subset of the features. We start with a comparison of intrinsic features and observe that, for D^{known}, nine out of twelve intrinsic features in Table 6.5 are also present in Table 6.11, with the same signs for the coefficients. Meanwhile, unavailable in the latter table are `amount1`, `organisations`, `people`, whereas `refused1` and

TABLE 6.10: Summary of baseline LR model for D^{fraud} with neighbourhood features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`Intercept`	-2.9405	0.191	-15.370	0.000
`n2.ratioFraud`	0.3368	0.036	9.440	0.000
`n2.size`	-3.6121	0.529	-6.828	0.000

`responsibilityCode_2` are observed significant in the all-features model but not in the intrinsic-features model. Considering D^{fraud} instead, we observe that thirteen out of fifteen features in Table 6.6 are also present in Table 6.12, again with the same signs for the coefficients. Contrary to the intrinsic features model, `organisations` and `daysReport` are insignificant in the all features model, whereas it adds `responsibilityCode_1` and `sameSits5`.

TABLE 6.11: Summary of baseline LR model for D^{known} with all features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`Intercept`	-2.8265	0.071	-39.667	0.000
`age`	-0.3718	0.028	-13.23	0.000
`amount`	0.2628	0.020	12.928	0.000
`claimAge`	-0.8041	0.035	-22.713	0.000
`lastClaim`	-0.1873	0.040	-4.642	0.000
`n1.max`	0.2047	0.079	2.596	0.009
`n1.size`	-0.3149	0.155	-2.026	0.043
`n2.binFraud`	-0.2352	0.057	-4.129	0.000
`n2.q1`	-0.2862	0.119	-2.414	0.016
`n2.ratioFraud`	0.3515	0.055	6.405	0.000
`n2.size`	-0.9146	0.079	-11.53	0.000
`numContracts`	-0.1138	0.029	-3.876	0.000
`refused1`	0.0831	0.030	2.795	0.005
`responsibilityCode_2`	-1.234	0.545	-2.264	0.024
`responsibilityCode_3`	1.0233	0.125	8.191	0.000
`responsibilityCode_x`	0.5949	0.08	7.479	0.000

Moving on to score features, we observe noticeably larger differences. For D^{known}, only `n1.max` is present in both Table 6.7 and 6.11, but with a negative coefficient in the former and a positive coefficient in the latter. For D^{fraud}, Tables 6.8 and 6.11 reveal a similar result for `n1.max`, although equivalent signs are observed for `n2.max`,

Considering neighbourhood features instead, we observe that, for D^{known}, Tables 6.9 and 6.11 match in terms of five significant features, with equivalent signs in the two tables. The only difference is that the former adds `n2.ratioNonFraud`. For D^{fraud}, the only difference in significant neighbourhood features is that `n2.size` feature is also deemed significant in the all features model.

6.4 Conclusion

In this chapter, we have presented the results of our study. This includes information required to assert whether our data establishes empirical evidence for the homophily as-

Table 6.12: Summary of baseline LR model for D^{fraud} with all features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`Intercept`	-4,3814	0,729	-6,013	0
`age`	-0,3385	0,028	-12,295	0
`amount`	0,0809	0,018	4,428	0
`amount1`	0,0717	0,022	3,194	0,001
`claimAge`	-0,6634	0,034	-19,526	0
`lastClaim`	-0,2628	0,043	-6,17	0
`lastClaim_x`	0,1142	0,03	3,802	0
`n1.max`	0,2377	0,072	3,316	0,001
`n2.binFraud`	-0,1466	0,055	-2,679	0,007
`n2.max`	0,1561	0,07	2,242	0,025
`n2.ratioFraud`	0,1187	0,027	4,327	0
`n2.size`	-5,0401	0,67	-7,524	0
`nClaims1`	0,1016	0,042	2,44	0,015
`nClaims5`	-0,1157	0,048	-2,435	0,015
`numContracts`	-0,1966	0,032	-6,077	0
`responsibilityCode_1`	0,2766	0,077	3,577	0
`responsibilityCode_3`	1,0521	0,121	8,7	0
`responsibilityCode_x`	0,7382	0,076	9,718	0
`sameSits1`	-0,1709	0,032	-5,365	0
`sameSits5`	0,1839	0,03	6,036	0

sumption, as well details pertaining to the importance of individual features and the classification performance of the baseline, time-weighted and shared resource models on the held-out test set. In the next chapter, we will interpret these findings and compare our results to ones reported by authors in the original study, which lays the foundation for answering this study's research questions.

Chapter 7

Discussion

In this chapter, we present an interpretation of our results, building upon the objective presentation provided in Chapter 6. Section 7.1 presents a detailed analysis of our findings concerning the research questions posited in this study. Section 7.2 compares these findings to the hypotheses presented earlier. Then, Section 7.3 sheds light on the main limitations of this study. Concerning these constraints, Section 7.4 offers recommendations for future research endeavours.

7.1 Answering the Research Questions

The analyses conducted as part of this study are founded on the research questions initially presented in Section 1.2 and repeated here:

RQ1 To what extent can empirical evidence for the homophily assumption presented in Óskarsdóttir et al. [19] be found in a different real insurance data set?

RQ2 How do the results presented in Óskarsdóttir et al. [19] generalise to a different real insurance data set?

RQ3 How can the social network analysis-based insurance fraud detection approach presented in Óskarsdóttir et al. [19] be adapted to enhance its classification performance?

- a What is the impact of extending the approach with time-weighted fraud influence and edges?
- b What is the impact of extending the bipartite network with party–party relations based on shared resources?

RQ4 How do the baseline and adapted models along with different combinations of feature sets compare in terms of highlighting interesting and/or suspicious claims that had not been investigated previously?

Hypotheses for these questions were that: 1) our data set also suggests some empirical evidence for the homophily assumption; 2) the main findings of the original work generalise to a different data set; 3) both adaptations yield a positive impact on the classification performance of the fraud detection model; and 4) both adaptations show enhanced results.

In this section, we consider these research questions to interpret the results presented in Chapter 6. First, Section 7.1.1 elucidates our findings in relation to RQ1. Then, Section 7.1.2 presents a comparison between our results and those reported by Óskarsdóttir et al. [19] to answer RQ2. Last, Section 7.1.3 provides an interpretation of the results achieved using the adapted models to answer RQ3.

7.1.1 Homophily Assumption

Óskarsdóttir et al. [19] report on their data establishing some empirical evidence for the homophily assumption, i.e., the idea that closely related instances are likely to behave in the same way [27]. This evidence is derived from analysing the average relative frequency of fraudulent claims in the second-and fourth-order neighbourhoods of both non-fraudulent and fraudulent claims. However, the authors did not report on conducting any significance test, raising concerns about their findings' robustness. Additionally, their reporting focuses solely on the average relative frequency of fraudulent claims in comparison to *all* claims in the respective neighbourhood. This approach may lead to an unjustly elevated frequency when a larger proportion of claims in the neighbourhood of fraudulent claims is investigated, even if the exact proportion of investigated claims revealing fraud remains consistent. Such a larger proportion of investigated claims in the neighbourhood of fraudulent claims is not inconceivable due to, for example, the influence of existing fraud detection models.

In this work, we have addressed both of the issues as mentioned above. For one, we have conducted T-tests to assert the significance of the observed differences in average relative frequencies. Furthermore, we have reported on the average frequency of fraudulent claims not only among *all* claims but also among all *investigated* claims in the respective neighbourhoods.

The results in Section 6.1 have affirmed a significant difference between fraudulent and non-fraudulent claims when comparing the average relative frequency of fraudulent claims in their second-order neighbourhoods, whether considering all claims or only investigated claims. Especially the observed significance when considering only investigated claims suggests some empirical evidence for homophily in the bipartite network of claims and parties. However, to further substantiate the existence of homophily in such network, additional analyses in consultation with domain experts are required. There might be alternative explanations that explain our observations, which are not accounted for in this study. For example, knowledge of an existing case of fraud might yield a positive impact on the chance that a fraud investigation into a related claim unveils perpetrated fraud by pointing the investigator into a certain direction.

The need for caution in drawing conclusions is further emphasised by the lack of a significant difference when considering fourth-order neighbourhoods instead of second-order neighbourhoods, though this might also be explained by the substantial size of these neighbourhoods. Fourth-order neighbourhoods are likely to be very large, such that the inclusion of a small sample of *known* fraudulent claims exhibits insufficient impact on the overall relative frequency.

In conclusion, referring back to RQ1, we have demonstrated that our data suggests evidence supporting the homophily assumption when considering the second-order neighbourhoods of claims, although alternative explanations might exist. Meanwhile, our analysis has highlighted limitations in the robustness of the homophily analysis conducted in the original work.

7.1.2 Generalisability of Findings

Along with other findings, Óskarsdóttir et al. [19] reported enhanced classification performance achieved by their social network analysis-based automobile insurance fraud detection model in comparison to models that rely on intrinsic features alone. To evaluate whether their findings generalise to alternative real insurance claims data sets, we compare the results reported in their study to the results achieved by a similar model on a similar real insurance data set in this study. We consider feature importance, sequential feature addition results, test set evaluations, and logistic regression model summaries.

Feature importance

First, note that our use of dummy encoding and the resulting loss of a one-to-one correspondence between individual dummy variables and responsibility code types suggests we can assign little meaning to the importance of individual `responsibilityCode` features. Correspondingly, the `responsibilityCode` feature importances are omitted from subsequent analyses.

Then, comparing the Gini importance of intrinsic features as reported in Figure 6.1 to those presented by authors of the original work, we notice that the features included in the top five features with the highest importance align, albeit in a different order. Meanwhile, a discrepancy is observed in the Gini importance of features pertaining to the policyholder's responsibility in historical claims. Notably, whereas these features exhibit noticeably small Gini importance in our model, their importance in the original work is average. In Section 5.1, we revealed that the `responsibilityCode` feature, which serves as the basis for these other properties, is available to only half of all claims in our data set. This might serve as an explanation for the diminished importance of historical responsibility properties in our model relative to the model in the original work.

Looking at neighbourhood features instead (Figure 6.2), we observe that the feature with the highest Gini importance in our study (`n1.size`) differs from the one in the original paper (`n2.size`). Conversely, both studies report very small Gini importance for the `n2.binFraud` feature. The latter presents some inconsistency with the idea of homophily, but might be partially explained by Gini importance's bias towards continuous variables and variables with many categories [111], since `n2.binFraud` is a binary feature. Nevertheless, it cannot be conclusively stated whether this represents the full cause.

Considering score features (Figure 6.3), our results revealed a mostly uniform distribution of Gini importances over features for D^{known} with slight variation for D^{fraud}, suggesting that none of the score features are significantly more important than others in making predictions in our study. This observation contrasts with the findings in the original work [19]. The original work's authors report a larger variation for D^{known}, with the highest importance attributed to `n2.max`. Meanwhile, for D^{fraud}, the authors reveal a major spike in the importance of the `scores0`, exceeding the importance of the other features by a factor of 1.5–2.

Focusing on all features (Figure 6.4) and considering D^{known} specifically, we observe that both the top five features displayed in our results and the top five features in the original work include `amount`, `daysReport`, `n1.size`, and `age`. In contrast, for D^{fraud}, a match is observed only in `amount` and `claimAge`. The original work reports that score feature `scores0` attains the highest Gini importance for D^{fraud}, followed by neighbourhood features `n1.size` and `n2.ratioNonFraud`. In our work, we instead see intrinsic features `claimAge`, `age`, and `daysReport` being attributed the largest feature importance.

The observed difference in top five features across the set of all features aligns with our more general observation that network features yield comparatively larger Gini importances in the original work than in our study. Our endeavours thus far have not yielded a comprehensive elucidation for this result. Nevertheless, we propose several hypotheses that might explain the substantial Gini importance assigned to the top five variables in our full feature set.

A conceivable rationale for the significant Gini importance of `amount` concerning D^{known} is that claims with higher values pose a larger financial burden for the insurer yet are more lucrative for the claimants. This dynamic might fuel increased attention from the insurance company toward investigating higher-value claims, paralleled by heightened interest from fraudsters in submitting such claims. Simultaneously, the large Gini importance assigned to the `claimAge` property could be ascribed to existing business rules. For instance, the existence of a business rule flagging a claim submitted shortly after obtaining a new insurance contract would sensibly detect past-posting, a fraudulent activity previously referred to in Section 5.1. A plausible rationale for for the pronounced Gini importance attributed to `numContracts` is that policyholders with a large history of contracts with the insurer might have somehow bolstered their perceived trustworthiness. Meanwhile, large Gini importance attributed to `age` might indicate overrepresentation of certain age groups in investigated and confirmed fraudulent cases. No hypothesis is currently posited for the elevated importance linked to `n1.size` and `daysReport`.

To establish whether there is any truth in the aforementioned hypotheses, a comprehensive data analysis would be necessary. However, this activity falls beyond the scope of this study. Confirmation of the existence of business rules contributing to the observed results could further validate the hypotheses, but this has been impeded by confidentiality constraints.

Sequential feature addition

Moving on, we focus on the results achieved by logistic regression classifiers constructed using different subsets of each feature set's most important features.

Our finding that the classification of D^{known} yielded enhanced or at least equivalent performance to the classification of D^{fraud} contrasts with the original work's results, where three out of four feature sets yielded equivalent or even improved performance in classifying D^{fraud} compared to D^{known}. Notably, Óskarsdóttir et al. [19] reported enhanced classification performance for D^{known} only when considering intrinsic features alone.

In addition, whereas our observation that utilising only intrinsic features led to enhanced performance compared to using only score or neighbourhood features aligns with the original paper's results for D^{known}, it diverges for D^{fraud}. In the case of D^{fraud}, the original work suggested improved classification performance using only score or neighbourhood features compared to using only intrinsic features.

Meanwhile, our observation that employing all features resulted in enhanced classification performance over the use of specific feature sets proved consistent with original work's findings, although the reported differences in their study are more substantial. Our observed maximum differences between all features and intrinsic features in AUC-ROC, AUC-PR, and TDL are in the range of 0.05, 0.05, and 0.5, respectively, contrasting with differences of approximately 0.1, 0.2, and 2 reported in the original work. For D^{known}, the differences are less pronounced.

Regarding the absolute values attained for the three performance metrics, we note that the range of AUC-ROC and TDL values reported by the original authors is comparable with the observations in this study. However, their reported AUC-PR values are substantially larger compared to our findings. Their maximum AUC-PRs, reaching approximately 0.5, demonstrate a two-fold improvement over the 0.25 observed in our findings.

Investigating when maximum performance is attained, our comparatively consistent performance from $n = 14$ features onwards deviates from the original paper. Figures in the original paper reveal that consistent performance was achieved at values of around $n = 10$, though the AUC-PR of their model for D^{known} further increased at $n = 15$, while their TDL for D^{fraud} declined as the number of features grew past $n = 9$.

To elucidate the slight discrepancies in outcomes between the current investigation and the original research, it is essential to consider our observations concerning the experimental

Table 7.1: Performance of baseline model compared to performance in [19]

		D^{known}			D^{fraud}		
Features	Study	AUC-ROC	AUC-PR	TDL	AUC-ROC	AUC-PR	TDL
Intrinsic	This study	0.767	0.207	4.445	0.735	0.105	3.985
	[19]	0.691	0.1214	2.85	0.662	0.0301	2.137
Score	This study	0.617	0.084	1.810	0.618	0.058	1.903
	[19]	0.634	0.0883	2.25	0.660	0.0402	2.812
Neighbourhood	This study	0.684	0.123	3.344	0.668	0.117	2.617
	[19]	0.681	0.1051	2.65	0.719	0.0481	3.262
All	This study	0.803	0.255	4.839	0.777	0.153	4.044
	[19]	0.725	0.1312	3.457	0.792	0.0810	3.824

setup employed by the original authors, as delineated in Section 5.4.4. In that section, we underscored that their publication suggests the utilisation of an inappropriate oversampling strategy, potentially leading to biased and excessively optimistic results. The proposition that their cross-validation outcomes are overly optimistic gains further support from the fact that, while the AUC-PRs demonstrate a twofold enhancement over the same metric in our study (as delineated in this section), their performance on the test set is comparable if not inferior to ours, as detailed in Section 7.1.2.

Evaluation on test set

Moving on to findings related to the evaluation on the test, Table 7.1 presents a comparison between the classification performance of our baseline logistic regression model and the results presented in the original work [19]. We observe that our baseline model consistently outperformed theirs in classifying D^{known} across all three metrics (AUC-ROC, AUC-PR, and TDL) when considering intrinsic, neighbourhood, or all features. However, the original work reported superior performance on D^{known} when considering score features specifically.

Examining the classification of D^{fraud} instead, we achieved a higher AUC-ROC using intrinsic features (0.735 versus 0.662), whereas the original work's authors achieved a higher AUC-ROC using score features (0.660 versus 0.618), neighbourhood features (0.719 versus 0.668), and all features (0.792 versus 0.777). Shifting our focus to the AUC-PR, our work delivered elevated performance compared to the model constructed by Óskarsdóttir et al. [19] across all feature sets. These enhancements were especially prevalent for intrinsic features (0.105 vs. 0.0301) and neighbourhood features (0.117 versus 0.0481), with smaller differences for score features (0.058 versus 0.0402) and all features (0.153 vs 0.0810). Regarding TDL, we reported enhanced performance compared to the original study for both D^{known} and D^{fraud} considering intrinsic features and the set of all features. Conversely, the original work's authors demonstrated elevated TDL across both data sets when considering score features or neighbourhood features exclusively.

Taking a different perspective, we note that both studies achieved the highest AUC-ROC, AUC-PR, and TDL when utilising the combined set of all features. For D^{known}, both studies also share equivalent rankings for the second to fourth place (intrinsic, neighbourhood, and score), whereas these rankings differ for D^{fraud}. Notably, whereas in our study, the ranking of feature sets for D^{fraud} aligns with that for D^{known}, the results in the original work reveal the set of only neighbourhood features as the second-best performing feature set across all metrics for D^{fraud}.

In conclusion, considering AUC-PR, we achieved consistently enhanced performance compared to the work in Óskarsdóttir et al. [19] across all feature sets and for both D^{known}

and D^{fraud}, except for the combination D^{fraud} with score features. No such general statement can be made for the other metrics. Both studies achieved the best performance across all metrics using the set of all features, but the rankings of subsequent feature sets differed.

One plausible explanation for the significantly enhanced AUC-PR in our study compared to the original work is the higher percentage of fraudulent samples in our data set: 2.69% versus 1.8%. This 50% increase indicates a less imbalanced class distribution, potentially influencing performance. Additionally, we note that the data used in our research spans a smaller number of years—four years compared to six years. Considering the suggested time-evolving nature of fraud [16, 30], this could imply that our classifier is less influenced by historical 'outdated' information, potentially enhancing performance. However, we lean towards the smaller time range negatively impacting performance instead, since our data encompasses significantly fewer claims than the few million claims reported in the original paper, providing our model with less data to learn from. For confidentiality reasons, we omit specifying the exact number of claims in our data set.

Model summaries

When comparing the model summaries from the original work to those presented in our study, we identify certain similarities, such as consistently matching coefficient signs for variables related to amounts, policyholder age, and claim age. However, an equal number of inconsistencies emerge, leading us to the conclusion that details regarding the internal workings of the models do not seamlessly generalise to ours.

For example, Óskarsdóttir et al. [19] reveal that their model for D^{fraud}, fitted on the set of all features, is dominated by score features. In contrast, our model is predominantly influenced by intrinsic features, aligning with earlier observations that underscored the heightened importance of network features in the original work compared to ours. Additionally, while the original study reveals disparities in variable coefficient signs between D^{known} and D^{fraud}, these coefficients consistently match in our study. Moreover, we observe discrepancies in the number of significant variables reported between their model and ours, possibly stemming from variations in how significance is interpreted in both studies. Unlike the authors of the original work, who may have considered significance in the context of stepwise forward and backward feature selection, we opted not to conduct such feature selection, as delineated in Section 5.4.6. Consequently, we rely on significance derived from the summaries of models constructed using all features within each feature set instead.

Having compared the model summaries in this work to those presented by the original authors, we assess whether the coefficients of the variables presented in our model summaries align with our intuition and the hypotheses regarding Gini importance discussed earlier in this section. To accomplish this, we take a holistic view of the model summaries provided in Tables 6.5 to 6.12, leveraging the consistency in coefficients across the individual summaries.

When examining variables linked to our hypotheses, we find that their coefficients align with our earlier intuition. Firstly, we observe a negative coefficient for the `age` feature in the model summaries related to the set of intrinsic or all features. In our earlier hypothesis, we suggested that the importance of the policyholder age feature could be attributed to an overrepresentation of certain age groups among investigated or confirmed fraudulent cases. The negative coefficient in the model summaries implies that the overrepresented age group might skew towards younger ages. Next, we observe positive coefficients for `amount` and `amount1`, matching our hypothesis regarding increased interest in claims associated with larger financial burdens. Then, we observe a negative coefficient for `claimAge`, which matches our previous hypothesis regarding past-posting: a larger disparity between the contract's begin date and the claim registration date decreases the predicted probability of fraud. Lastly, the negative coefficient for `numContracts` matches our hypothesis regarding perceived trustworthiness: a higher number of (historical) contracts a yields a smaller predicted probability of fraud in our model.

Hypotheses regarding score and neighbourhood features were not explicitly presented in this section, but the idea of homophily that is at the foundation of this research suggests that indicators signifying fraud in the neighbourhood of a claim should increase that claim's fraud probability. The positive coefficients for `scores0`, `n2.max` and `n2.ratioFraud` align with this proposed idea. However, we observe negative coefficients for score features `n1.max`, `n2.med`, `n2.q1` and neighbourhood feature `n2.binFraud` whereas, based on homophily, one would expect these to increase the fraud score.

In addition to revealing coefficients that defy intuition, our model summaries also indicate conflicting coefficients between similar features. For instance, while the positive coefficient for `n2.ratioFraud` aligns with the idea of homophily, a positive coefficient is also assigned to the opposite `n2.ratioNonFraud` feature. Similarly, features `nClaims1` and `nClaims5`, representing the same features over different time spans, exhibit coefficients that do not align. Intuitively, one might anticipate a frequent claimer being considered more suspicious and expect a positive coefficient, but this expectation does not hold for `nClaims5`. A similar inconsistency is observed for `sameSits1` and `sameSits5`, which also depict the same characteristic but for different periods. However, we lack intuition for the expected coefficients for these values, except through their suspected correlation to `nClaims1` and `nClaims5`.

In light of these findings, we repeat Óskarsdóttir et al. [19] in stating that the coefficient estimates might be unreliable due to multicollinearity—the phenomenon where two or more independent variables in a regression model are highly correlated, making it challenging to isolate the individual effects of each variable on the dependent variable [147]. However, an evaluation of multicollinearity was out-of-scope for this study.

Conclusion

In conclusion, we revisit RQ2 and summarise the generalisability of findings from the original work [19]. Initially, we note that the intrinsic features with the highest Gini importance in their original work align with those in our study. However, their results reveal abundantly larger Gini importance for network (i.e., score and neighbourhood) features compared to ours, indicating a potentially larger contribution of network features in their model.

This observation is consistent with the sequential feature addition results: while both studies achieved the best results with the inclusion of all features, the original work identified neighbourhood and score features as the top-performing feature sets, in contrast to intrinsic features in our study. Furthermore, whereas our study showed enhanced performance in classifying D^{known} compared to D^{fraud}, the opposite held true in the original work. A crucial limitation surrounding these findings is that the experimental setup in the original work suggests their sequential feature addition results might be biased and overly optimistic, which should be considered when interpreting their findings. This might explain their two-fold increase in achieved AUC-PR compared to our results.

Moving on to model summaries, we again observe discrepancies between the model summaries presented in the original paper and in this report. This suggests limited generalisability of these details, notwithstanding difficulties in model interpretation following potential multicollinearity issues.

Finally, considering the evaluation on the test set, a consistent observation is made that the models using all features exhibited the best performance in both studies. However, when ranking individual feature sets based on performance, discrepancies emerge, with network features emphasised in the original work compared to our study. Overall, our study consistently demonstrated enhanced AUC-PR compared to the existing study, while differences in AUC-ROC and TDL are less uniform.

7.1.3 Impact of Adaptations

In Table 6.3, we presented the results of evaluating the adapted models on the test set. When using all features, the results revealed slight differences in performance between the baseline models and our adapted models, in favour of the adapted models. However, these are insufficiently large to conclusively state that the adaptations present a valuable contribution over the baseline model that was originally proposed by Óskarsdóttir et al. [19].

To explain these observations, a first suggestion would be that the limited impact that the adaptations have had on the performance of the all-features model might be partially explained by the fact that the model adaptations are only propagated through the score feature set and thus represented in the score feature set and the set of all features only. Accordingly, they might be overshadowed by the large number of alternative features. If this were the case, larger differences in performance should still be noted when considering models that use the score features only. For the shared resources model, this holds when considering relative performance differences, although the absolute differences remain small. For D^{known}, we observed a difference in AUC-PR and TDL of approximately 25% (0.105 vs 0.084) and 28% (2.321 versus 1.810), respectively. For D^{fraud}, the difference in AUC-PR was approximately 12% (0.065 versus 0.058) and the difference in TDL was approximately 22% (2.312 versus 1.903) instead. For the time-weighted model, both the relative and absolute differences are significantly less convincing.

One hypothesis for the consistently negligible impact of time-weighting is related to our experimental setup. As described in Section 5.5, we adopt a fraud decay constant that gradually reduces the influence of a fraud occurrence to 50% over eight years. Our training data encompasses three years only. As a result, the most historical fraudulent claim in our data is given an influence of

$$e^{-\frac{\ln 0.5}{365.25\times 8}\times 365.25\times 4} \approx 0.707, \tag{7.1}$$

representing approximately 70.7% of its full influence in the baseline model. This difference in fraud influence might be insufficiently large to yield a substantial impact on the fraud scores in the network.

An additional hypothesis would be that the proposed value of emphasising recent fraud over historical fraud does not weigh up to the ‘loss’ of fraud information this introduces. One of the major complexities in automatically detecting fraud is its uncommon nature, i.e., there are few known fraudulent cases to learn from. Introducing time-weighting might, to some extent, lead to a further effective decrease in the available information.

7.1.4 Fraud Expert Evaluations

In Section 6.3.2, we presented the results of different combinations of models and feature sets in recalling interesting, previously unknown (i.e., unlabelled) claims, based on fraud experts’ evaluations of the top twenty claims from each model that were assigned the highest probability of fraud. It was discovered that the score and neighbourhood network features produced the highest precision-at-k (0.800), while classification based solely on intrinsic features produced the lowest precision-at-k (0.550). Comparing the different models, we observe that the time-weighted model consistently yields the best achieved performance, albeit together with or with only a slight edge over the baseline model.

The superior performance achieved by applying network features (score or neighbourhood) over intrinsic features contradicts our previous findings, which suggested that intrinsic features increased performance instead. This could imply that intrinsic characteristics are particularly good at identifying claims that have previously been discovered by existing fraud detection systems, whereas score and neighbourhood features are better at recalling further fraud. It should be emphasised, however, that the classification of claims by fraud experts was based on whether the claim was sufficiently interesting for them to want the claim reported for further study, rather than whether the claim had been discovered fraudulent. Furthermore, as explained in Section 7.3.3, the robustness of the experimental setting used for the fraud expert evaluations was limited.

7.2 Re-Evaluating Our Hypotheses

Comparing the hypotheses introduced in Section 1.3 to our findings presented in the previous section (Section 7.1), we make a few observations.

For RQ1, we hypothesised that empirical evidence supporting the homophily assumption is also present in our data set. Section 7.1.1 revealed that our findings are consistent with this hypothesis when considering second-order neighbourhoods of claims. However, no such evidence was found when considering fourth-order neighbourhoods instead.

Considering RQ2 instead, we hypothesised that the original work's [19] main findings generalise to the data set used in this. Indeed, Section 7.1.2 showed that both studies report consistent findings in terms of network features enhancing the classification performance of considered fraud detection models. More granular results showed inconsistencies with differences pertaining to, for example, the importance of individual feature sets. This aligns with our prior expectations.

Focusing on RQ3, we hypothesised that both adaptations yield a positive impact on the classification performance of the model. This hypothesis is not corroborated by our findings in Section 7.1.3. In terms of time-weighted fraud influence, our approach yielded no substantial impact on classification performance for any of the feature sets, whether considering AUC-ROC, AUC-PR, or TDL. Focusing on shared resources instead, positive differences were only observed in the classification performance achieved using score features.

For RQ3, our hypothesis was that the adapted models would show enhanced performance compared to the baseline model. Considering the time-weighted model, this held true when considering the set of all features only. Focusing on the shared resources model instead, the presented results showed consistently diminished performance when compared to the baseline.

7.3 Limitations

Having elucidated the primary findings of our study concerning the research questions, we outline several limitations that might have influenced the validity of the reported results. Initially, Section 7.3.1 addresses inconsistencies in our replication of the existing work. Subsequently, Section 7.3.2 illuminates (potential) inconsistencies within our data set. Then, Section 7.3.3 assesses the robustness of our results. Following that, Section 7.3.4 scrutinises potential limitations in our implementation and evaluation of the adapted models. Lastly, Section 7.3.5 highlights major limitations in the generalisability of the results on the test set to the practical deployment of the model.

7.3.1 Replication Inconsistencies

Despite our best efforts in replicating the data set and models employed in the original work, the original paper provided insufficient details to ensure full confidence that our models and data set match their exact characteristics. For example, we have had to make assumptions regarding the value of the `lastClaim` feature when there was no previous claim (Section 5.4.4) and regarding model parameters for both the random forest classifier (Section 5.4.5) and the logistic regression classifier (Section 5.4.6). This uncertainty limits the validity of our findings regarding the generalisation of their results, presented in Section 7.1.2.

A further discrepancy between the two studies might involve how fraud information was used in the experimental setup. As delineated in Section 5.4.1, we adopted the same approach as Óskarsdóttir et al. [19] by constructing a query vector that excluded information on fraud pertaining to claims registered in the last year that is covered by the data set. However, the original work did not specify whether this information was also excluded during the extraction of network and neighbourhood features. In this study, we considered all information during the extraction of score and neighbourhood, which might have had some impact on the achieved results due to a minor form of data leakage. For example, consider a situation where the feature `n2.binFraud` for a claim in the test set holds true because a claim in their second-order neighbourhood was fraudulent, whereas this claim was only investigated *because* the claim in the test set was proven fraudulent before. In a practical setting, this type of situation would not occur.

In addition to these unknown differences between the models and data sets employed in the two studies, we also established some explicit differences between the two works. These discrepancies will have further limited the validity of the comparison between the two works, especially when considering detailed results instead of taking a broader perspective. They include our omission of the `police` intrinsic feature (Section 4.1.1), our use and corresponding representation of one-hot encoded features (Section 5.4.4), and the lack of sequential feature selection employed for the construction of the logistic regression classifier that was used to classify the test set (Section 5.4.6). The latter might have especially contributed to diverging results regarding the evaluation on the test set and the model summaries, as previously discussed in Section 7.1.2. We also reiterate a discrepancy in the total number of years spanned by the employed data sets. Their potential impact on the results was previously delineated in Section 7.1.2.

7.3.2 Data Inconsistencies

In addition to inconsistencies in the replication of the original work, there might have also been inconsistencies in the data set employed in this study that were not uncovered during our analyses.

One known inconsistency is related to the amalgamation of parties in the bipartite network based on the matching of postal codes and names, as detailed in Section 4.3.1. This merging process was not applied during the construction of the intrinsic feature set, suggesting the computed values for intrinsic features related to the policyholder, such as their number of claims in the past years or their number of contracts, may not always be accurate. Other potential errors in the data set might have also gone unnoticed in our research. These might involve errors in the raw data used to compile the data set, or inaccuracies in the sometimes intricate transformations from raw data into the specified features that were not uncovered during our evaluations on correctness.

7.3.3 Robustness of Results

In Section 6.2, we detailed the hyper-parameters that produced optimal performance across all parameters considered in a grid search and directed readers to Appendix A for insights into the results attained by alternative configurations. An examination of the results presented in Appendix A reveals that, while the configurations in Table 6.2 indeed yielded the best mean AUC-PR, the mean results for each set of hyper-parameters were associated with substantial standard deviations. For instance, the mean AUC-PR of the worst-performing model on the $\mathcal{D}^{\text{fraud}}$ data set with all features falls within the range of the mean AUC-PR plus or minus the corresponding standard deviation. These large standard deviations suggest potential limitations in the robustness of the individual models. Consequently, repetitions of the same type of grid search might provide different results with varying Gini importance of individual features. Unfortunately, Óskarsdóttir et al. [19] did not present their grid search results. As a result, we cannot establish if similar variability existed in their study.

Our study has also revealed potential limitations regarding the methodology in the original work [19] in terms of feature importance, which was replicated in this study. We have assumed that the importance evaluation in the original work was conducted based on the features' Gini importance. As delineated in Section 4.3.6, it is argued that the Gini importance can yield misleading results. This might have introduced bias in the findings presented in both this and the original work, limiting their value.

A similar limitation relates to the issue of multicollinearity, previously delineated in Section 7.1.2 and also acknowledged in the original work. This factor may have introduced complexities in correctly interpreting the coefficients in the logistic regression models, potentially compromising the validity of the associated findings.

Moreover, a significant limitation affecting all studies on fraud detection is that the evaluations of fraud detection models are often based on data sets where ground truth labels are known for only a small fraction of claims. There is a common implicit assumption that any claim not labelled as fraudulent is legitimate, an assumption that is also at the core of our classification of D^{fraud}. However, for the majority of the claims in these data sets, it is unknown whether they involve fraud or not, as they have not undergone prior investigation. We attempted to address this issue through fraud expert evaluations of previously unknown (i.e., unlabelled) claims. Nonetheless, due to regulatory constraints and its labour-intensive nature, this evaluation was restricted in its contribution to resolving this matter.

More specifically, the corresponding findings were derived from an evaluation of a small sample of only twenty claims per model, limiting the generalisability of the results. Additionally, each claim was evaluated by a single expert only, disallowing an assessment of intra-rater reliability Lastly, we reiterate that fraud experts' evaluations involved classification based on suspicious or anomalous characteristics, rather than proven fraud.

7.3.4 Implementation and Evaluation of Adapted Models

Concerning our implementation of the adapted models, Section 7.1.3 already reported that the influence of time-weighting might have been too small due to the relatively short period covered by our data set compared to the chosen decay constant. However, there are also potential limitations in our implementation of the shared resources.

Unlike the hyper-parameters used in the equations adopted for BiRank (Section 5.4.2), the hyper-parameters employed in the equation that is used to iteratively update $\boldsymbol{p}$ in TriRank sum to 2 (Section 5.6). In this study, we have observed that this configuration yields convergence and produces fraud scores that align with intuition. However, the sum of hyper-parameters resembles a deviation from the requirement set for hyper-parameters in BiRank, which should sum to one [29]. Consequently, the theoretical proof of convergence might not immediately extend to the approach adopted in this work, and situations deviant from the ones considered might yield undesired results.

7.3.5 Generalisation of Results to Practice

Regarding the applicability of the findings to the practical application of the model, a significant limitation stems from a flaw identified in the original experimental design, which has persisted in our own experimental setup. Notably, as outlined in Section 5.4.4, the experimental setup involves constructing an analytical model data set by combining all labelled claims with a random sample of 20,000 unlabelled claims. This data set is then employed to form both training and testing sets. Consequently, the class distribution in the test set fails to accurately mirror the class distribution observed in practice, which suggests that the reported performance metrics may not accurately reflect the model's performance in real-world applications. Therefore, although relative comparisons of the reported AUC-ROC, AUC-PR, and TDL can be meaningful, the absolute values do not accurately represent the true performance metrics that would be achieved following deployment of the model.

7.4 Future Work

Acknowledging the limitations in the preceding section, we first present directions for future research that focus on addressing some of these issues. Then, we propose several alternative suggestions for future research that shall inspire readers to conduct further exploration into the direction of automobile insurance fraud detection.

7.4.1 Suggestions for Addressing the Limitations

To address some of the limitations of this study, we first propose adopting an alternative approach to establishing feature importance, such as the permutation importance approach detailed in Section 2.3.2. This shall enhance the validity of the importance evaluation, facilitating a deeper understanding of the features that are likely indicators of fraud. The implementation of this alternative feature importance ranking approach should be relatively straightforward given existing implementations in, for example, the popular *scikit-learn* Python library that has been extensively used in this study. However, its use requires the construction of a validation data set, as delineated in Section 4.3.6, and thus a change in the experimental setup.

In conjunction with the previous suggestion, we also propose analysing the multicollinearity of the included features using the variable importance factor (VIF) [147], which can indicate whether a variable is included in a linear dependency. Understanding the presence of multicollinearity in the data set allows for the implementation of appropriate measures to address the issue, subsequently enhancing the interpretability of coefficients in logistic regression models. In turn, this yields a positive impact on the interpretability of classifications made by the corresponding models.

Additionally, we propose suggestions that shall enhance the robustness of findings concerning the adapted models. Regarding an evaluation of the impact of introducing time-weighted fraud influence, we suggest employing a data set spanning a more extensive timeframe. This shall increase its influence on the fraud scores in the network, yielding better insight into the value of this adaptation. Considering shared resources instead, we propose to establish theoretical proof that the chosen parameters for the TriRank model yield expected results or to conduct more extensive evaluations based on a larger number of sample network configurations. This shall aid in determining whether the chosen approach is generally suitable or not. Taking a broader perspective and considering both adaptations, we propose making adjustments to the considered features to more explicitly incorporate the adaptations. By doing so, the adaptations shall have a more significant impact on the model, which shall more clearly present whether they comprise a valuable contribution over the baseline model that was considered.

Lastly, we suggest practitioners who are interested in the practical deployment of this model to first evaluate its performance on a test set whose class distribution is representative of the class distribution observed in the insurers' full set of claims.

7.4.2 Alternative Suggestions

In addition to suggestions that address some of the mentioned limitations in the current study, we also propose alternative directions that might prove interesting.

One direction would be to conduct an extensive evaluation of the impact of alternative hyper-parameter configurations on the performance of the model. As highlighted by Benedek, Ciumas and Nagy [31] and recognised in our systematic literature review, this type of hyper-parameter optimisation is often omitted from studies on automobile insurance fraud detection. Meanwhile, in the context of this study, for example, it can be expected that a change in hyper-parameters employed in the ranking algorithms yields a substantial impact on the final fraud scores assigned to nodes in the network.

An alternative direction would be to explore the impact of adopting alternative resampling methods on the performance, in line with an earlier recommendation in the original work [19]. In this study, we used SMOTE to construct synthetic samples that increase the percentage of minority class samples in the training data set from approximately 2.69% and 4.06% to 15% (Section 5.4.4), such that the majority of the minority class samples on which the classifiers were trained were synthetic. This prompts the idea that the approach taken to construct these synthetic samples might have a large influence on the classifier and as such, an exploration of alternative resampling methods might be an interesting direction for future research.

Last, in line with the recommendations in our literature review, we propose considering graph representation learning approaches for automobile insurance fraud detection. For that purpose, inspiration can be taken from an existing literature review on graph neural networks and their applications in other domains [148]. By adopting a graph representation learning approach, the need to conduct manual network feature engineering is diminished. However, the limited interpretability of graph neural network approaches when compared to the approaches taken in this study might yield concerns for its practical deployment.

7.5 Conclusion

In this chapter, we have presented an interpretation of the study's results concerning the research questions. Additionally, we presented a discussion regarding the limitations that warrant consideration when evaluating these results, along with suggestions for future research aimed at either addressing some of the limitations identified in this study or contributing valuable insights in alternative ways.

In the next chapter, we conclude this report by reiterating the main findings of both this study and the systematic literature review that preceded this.

Chapter 8

Conclusion

Throughout this work, we have been exploring the domain of automobile insurance fraud detection by focusing on data mining-based practices that assist fraud experts in choosing the right claims to attend to. This process was initiated by conducting a systematic literature review of recent literature to gain insight into the current state-of-the-art, followed by practical research into one direction specifically.

Our systematic literature review was exclusively focused on the data sets, detection methods, and resampling methods employed in recent studies on automobile insurance fraud detection. Its findings were summarised in Chapter 3. Through an analysis of fifty academic articles published between January 2019 and March 2023, it corroborated earlier notions that suggested severe limitations in publicly available automobile insurance fraud data sets to conduct research on, and revealed widespread use of resampling methods to address the issue of imbalanced class distributions in insurance fraud data sets. In terms of detection methods, the literature review showed a predominant focus on the detection of insurance *claims* fraud using supervised learning methods, although unsupervised methods were also considered. Reportedly novel types of detection approaches that were proposed include methods that utilise unstructured textual data and the use of graph-based techniques, though a variety of other types of methods was also evaluated.

In this practical part of our research, we took inspiration from an existing article [19] on a graph-based method evaluated in our literature review. This article suggested homophily in their bipartite network of claims and associated parties, proposed leveraging BiRank to compute fraud scores within this network, and reported enhanced claims classification performance by using features extracted from this network. The focus of our research was on first analysing the generalisability of the article's findings by constructing a mostly equivalent model and data set, followed by exploring the value of two distinct adaptations to their proposed model. Our research also sought to address the prevalent issue that uninvestigated claims cannot be confidently assigned the 'legitimate' status. To tackle this, we enlisted fraud experts to evaluate suspicion in previously unlabelled claims identified as suspicious by our fraud detection models.

Considering generalisability, our results in Chapter 6 matched our hypothesis in Section 1.3 that the original authors' main findings generalise to a different real insurance data set. This concerns suggested evidence for homophily in the bipartite network of claims and parties, presented in Section 7.1.1, and enhanced performance when also considering network features during supervised classification, as discussed in Section 7.1.2. Concerning more granular findings instead, inconsistencies emerged. These encompass variation in the importance of both feature sets and individual features, elaborated upon in Section 7.1.2.

Turning attention to proposed adaptations, the findings outlined in Section 7.1.3 revealed that our initial hypothesis, suggesting that both changes would enhance classification performance, was not supported by our results. To elaborate further, our implementation of time-weighted fraud influence yielded negligible impact on the classification performance of the model. Extending the claims–parties network with shared resources yielded a marginally more positive impact, yet still insufficient to conclusively establish its worth.

In the context of fraud experts' evaluation of unlabelled claims, our results again contradicted our hypothesis that both adaptations would enhance performance. Disregarding major limitations in the robustness of the evaluation, our findings in Section 7.1.4 suggested that while marginally better performance was observed for the time-weighted model, the shared resources model performed worse than the baseline.

In our scrutiny of the existing paper, several limitations in the robustness of their employed methodology and experimental setup surfaced, influencing the validity of their results. These limitations encompassed inappropriate ratios and the absence of significance tests for establishing homophily in their data (Section 7.1.1), the improper use of oversampling before ten-fold cross validation (Section 5.4.4), potential bias in the employed feature importance ranking mechanism (Section 4.3.6) and interpretation (Section 7.1.2), and limited generalisability of their test results to the practical deployment of the model (Section 7.3.5). The former two limitations were explicitly addressed in this study through the adoption of alternative strategies. However, the latter two persisted in this research, albeit explicitly acknowledged.

Further limitations in our study stem from imperfections in our replication of the original model due to missing details in the original work or resource constraints, as outlined in Section 7.3.1. Consequently, inconsistencies in results may not be attributed to the data set alone, but also to model inconsistencies. Additional limitations involve potential inconsistencies in our data set, as discussed in Section 7.3.2. Moreover, limitations in the evaluation of the proposed adaptations prevent us from drawing definitive conclusions regarding their value, awaiting further research instead, as mentioned in Section 7.3.4.

Based on this practical research, a primary recommendation for future work is to address the limitations in this and the preceding research by exploring alternative feature importance evaluations, tackling multicollinearity in the data set, and evaluating the model's performance on a test set whose class distribution resembles the one observed in a practical setting. This would provide a more profound understanding of the proposed model's value to insurers, greatly enhancing the practical contribution of both this research and the study based upon which this research was founded. Subsequent steps could encompass investigating the impact of time-weighted fraud influence using data spanning a larger timeframe, exploring the generalisability and theoretical validity of the proposed shared resources model, and evaluating the effect of different hyperparameters on the models' classification performance. Alternatively, one could explore a different type of graph-based approach by considering graph representation learning techniques for automobile insurance fraud detection. This eliminates the need for manual feature engineering, albeit sacrificing some interpretability.

Bibliography

[1] Stijn Viaene and Guido Dedene. 'Insurance Fraud: Issues and Challenges'. In: *The Geneva Papers on Risk and Insurance - Issues and Practice* 29.2 (1st Apr. 2004), pp. 313–333. ISSN: 1468-0440. DOI: 10.1111/j.1468-0440.2004.00290.x.

[2] Centrum Bestrijding Verzekeringscriminaliteit. *CBV Factsheet - November 2022*. Nov. 2022. URL: https://www.verzekeraars.nl/media/10724/cbv_factsheet_nov_2022.pdf.

[3] Het Verbond van Verzekeraars. *About the Association*. Verbond van Verzekeraars. URL: https://www.verzekeraars.nl/en/about-the-association (visited on 15/02/2023).

[4] Centrum Bestrijding Verzekeringscriminaliteit. *Verzekeringscriminaliteit*. URL: https://www.verzekeraars.nl/branche/verzekeringscriminaliteit (visited on 24/07/2023).

[5] Centrum Bestrijding Verzekeringscriminaliteit. *CBV Factsheet - september 2018*. Sept. 2018. URL: https://www.verzekeraars.nl/media/5143/factsheet-verzekeringsfraude-najaar-2018.pdf.

[6] Centrum Bestrijding Verzekeringscriminaliteit. *CBV Factsheet - Oktober 2019*. Oct. 2019. URL: https://www.verzekeraars.nl/media/6611/factsheet-verzekeringsfraude-oktober-2019.pdf.

[7] Centrum Bestrijding Verzekeringscriminaliteit. *CBV Factsheet - Oktober 2021*. Oct. 2021. URL: https://www.verzekeraars.nl/media/9513/verbondvanverzekeraars_factsheet_fraude_2021.pdf.

[8] Centrum Bestrijding Verzekeringscriminaliteit. *CBV Factsheet - Oktober 2020*. Oct. 2020. URL: https://www.verzekeraars.nl/media/7947/cbv_factsheet_fraude_oktober-2020.pdf.

[9] Insurance Europe. *Insurance fraud: not a victimless crime*. Nov. 2019. URL: https://www.insuranceeurope.eu/mediaitem/2bf88e16-0fe2-4476-8512-7492f5007f3c/Insurance%20fraud%20-%20not%20a%20victimless%20crime.pdf.

[10] Insurance Europe. *Mission statement*. Insurance Europe. URL: http://www.insuranceeurope.eu/ (visited on 07/02/2023).

[11] Insurance Europe. *Annual Report 2020-2021*. 31st May 2021. URL: http://www.insuranceeurope.eu/ (visited on 24/07/2023).

[12] David M. W. Powers. *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*. 10th Oct. 2020. DOI: 10.48550/arXiv.2010.16061. arXiv: 2010.16061[cs,stat].

[13] Het Verbond van Verzekeraars. *Recordaantal verzekeringsfraudeurs opgespoord*. Verbond van Verzekeraars. 24th Oct. 2019. URL: https://www.verzekeraars.nl/publicaties/actueel/recordaantal-verzekeringsfraudeurs-opgespoord (visited on 07/02/2023).

[14] *Artikel 3:17 Wet op het financieel toezicht*. 1st Jan. 2024. URL: http://wetten.overheid.nl/jci1.3:c:BWBR0020368&titeldeel=3&hoofdstuk=3.3&afdeling=3.3.3&paragraaf=3.3.3.1&artikel=3:17&lid=2.

[15] Tweede Kamer der Staten-Generaal. *Kamerstuk 29708, nr. 10*. Officiële bekendmakingen. 8th Mar. 2005. URL: https://zoek.officielebekendmakingen.nl/kst-29708-10.html?idp=LegalIntelligence (visited on 15/01/2024).

[16] Véronique Van Vlasselaer et al. 'GOTCHA! Network-Based Fraud Detection for Social Security Fraud'. In: *Management Science* 63.9 (Sept. 2017), pp. 3090–3110. ISSN: 0025-1909, 1526-5501. DOI: 10.1287/mnsc.2016.2489.

[17] A. Abdallah, M.A. Maarof and A. Zainal. 'Fraud detection system: A survey'. In: *Journal of Network and Computer Applications* 68 (2016). Publisher: Academic Press, pp. 90–113. ISSN: 10848045. DOI: 10.1016/j.jnca.2016.04.007.

[18] Lovro Šubelj, Štefan Furlan and Marko Bajec. 'An expert system for detecting automobile insurance fraud using social network analysis'. In: *Expert Systems With Applications* 38.1 (Jan. 2011). Place: USA Publisher: Pergamon Press, Inc., pp. 1039–1052. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2010.07.143.

[19] Maria Óskarsdóttir et al. 'Social Network Analytics for Supervised Fraud Detection in Insurance'. In: *Risk Analysis* 42.8 (2022), pp. 1872–1890. ISSN: 02724332. DOI: 10.1111/risa.13693.

[20] European Insurance and Occupational Pensions Authority. *Big data analytics in motor and health insurance: a thematic review*. Luxembourg: Publications Office, 21st May 2019. ISBN: 978-92-9473-142-5. URL: https://data.europa.eu/doi/10.2854/54208.

[21] Clifton Phua et al. 'A Comprehensive Survey of Data Mining-based Fraud Detection Research'. In: (2010). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1009.6119.

[22] A.K. Jain, Jianchang Mao and K.M. Mohiuddin. 'Artificial neural networks: a tutorial'. In: *Computer* 29.3 (Mar. 1996). Conference Name: Computer, pp. 31–44. ISSN: 1558-0814. DOI: 10.1109/2.485891.

[23] Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik. 'A training algorithm for optimal margin classifiers'. In: *Proceedings of the fifth annual workshop on Computational learning theory*. COLT '92. New York, NY, USA: Association for Computing Machinery, 1st July 1992, pp. 144–152. ISBN: 978-0-89791-497-0. DOI: 10.1145/130385.130401.

[24] Chen Liang et al. 'Uncovering Insurance Fraud Conspiracy with Network Learning'. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR'19. New York, NY, USA: Association for Computing Machinery, 18th July 2019, pp. 1181–1184. ISBN: 978-1-4503-6172-9. DOI: 10.1145/3331184.3331372.

[25] S Boccaletti et al. 'Complex networks: Structure and dynamics'. In: *Physics Reports* 424.4 (Feb. 2006), pp. 175–308. ISSN: 03701573. DOI: 10.1016/j.physrep.2005.10.009.

[26] Association of Certified Fraud Examiners. *Insurance Fraud Handbook*. Austin, TX: Association of Certified Fraud Examiners, 2009. URL: https://web.actuaries.ie/sites/default/files/erm-resources/insurance_fraud_handbook.pdf.

[27] Ravi Bapna and Akhmed Umyarov. 'Do Your Online Friends Make You Pay? A Randomized Field Experiment on Peer Influence in Online Social Networks'. In: *Management Science* 61.8 (Aug. 2015), pp. 1902–1920. ISSN: 0025-1909, 1526-5501. DOI: 10.1287/mnsc.2014.2081.

[28] Armen S. Asratian, Tristan M. J. Denley and Roland Häggkvist. *Bipartite Graphs and their Applications*. Google-Books-ID: l8fLCgAAQBAJ. Cambridge: Cambridge University Press, 13th Aug. 1998. 274 pp. ISBN: 978-1-316-58268-8.

[29] Xiangnan He et al. 'BiRank: Towards Ranking on Bipartite Graphs'. In: *IEEE Transactions on Knowledge and Data Engineering* 29.1 (Jan. 2017). Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 57–71. ISSN: 1558-2191. DOI: 10.1109/TKDE.2016.2611584.

[30] David Jensen. 'Prospective Assessment of AI Technologies for Fraud Detection: A Case Study'. In: *Proceedings of AAI-7 Workshop on AI Approaches to Fraud Detection & Risk Management*. AAAI Press, 1997. URL: https://www.semanticscholar.org/paper/Prospective-Assessment-of-AI-Technologies-for-Fraud-Jensen/012248dd502ace444aae276bfb353c8c7821de1b?sort=is-influential.

[31] B. Benedek, C. Ciumas and B.Z. Nagy. 'Automobile insurance fraud detection in the age of big data – a systematic and comprehensive literature review'. In: *Journal of Financial Regulation and Compliance* 30.4 (2022). Publisher: Emerald Group Holdings Ltd., pp. 503–523. ISSN: 13581988. DOI: 10.1108/JFRC-11-2021-0102.

[32] N.J. Morley, L.J. Ball and T.C. Ormerod. 'How the detection of insurance fraud succeeds and fails'. In: *Psychology, Crime and Law* 12.2 (2006), pp. 163–180. ISSN: 1068316X. DOI: 10.1080/10683160512331316325.

[33] Karen M. Gill, Adrian Woolley and Martin Gill. 'Insurance fraud: the business as a victim?' In: *Crime At Work: Studies in Security and Crime Prevention Volume I*. Ed. by Martin Gill. London: Palgrave Macmillan UK, 2005, pp. 73–82. ISBN: 978-1-349-23551-3. DOI: 10.1007/978-1-349-23551-3_6.

[34] Leman Akoglu, Hanghang Tong and Danai Koutra. 'Graph based anomaly detection and description: A survey'. In: *Data Mining and Knowledge Discovery* 29.3 (May 2015). Place: USA Publisher: Kluwer Academic Publishers, pp. 626–688. ISSN: 1384-5810. DOI: 10.1007/s10618-014-0365-y.

[35] Michele Tumminello et al. 'Insurance fraud detection: A statistically validated network approach'. In: *Journal of Risk and Insurance* (2022). Publisher: John Wiley and Sons Inc, pp. 1–39. ISSN: 00224367. DOI: 10.1111/jori.12415.

[36] R.J. Bolton and D.J. Hand. 'Statistical fraud detection: A review'. In: *Statistical Science* 17.3 (2002), pp. 235–255. ISSN: 0883-4237. DOI: 10.1214/ss/1042727940.

[37] Object Management Group. *Business Process Model And Notation 2.0*. Dec. 2010. URL: https://www.omg.org/spec/BPMN/2.0/ (visited on 04/01/2024).

[38] Leo Breiman. 'Random Forests'. In: *Machine Learning* 45.1 (1st Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324.

[39] Tin Kam Ho. 'The random subspace method for constructing decision forests'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.8 (Aug. 1998). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 832–844. ISSN: 1939-3539. DOI: 10.1109/34.709601.

[40] Robert Bryll, Ricardo Gutierrez-Osuna and Francis Quek. 'Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets'. In: *Pattern Recognition* 36.6 (1st June 2003), pp. 1291–1302. ISSN: 0031-3203. DOI: 10.1016/S0031-3203(02)00121-8.

[41] scikit-learn developers. *1.11. Ensembles: Gradient boosting, random forests, bagging, voting, stacking*. scikit-learn. URL: https://scikit-learn/stable/modules/ensemble.html (visited on 09/11/2023).

[42] Leo Breiman et al. *Classification And Regression Trees*. 1st ed. New York: Routledge, 25th Oct. 2017. ISBN: 978-1-315-13947-0. DOI: 10.1201/9781315139470. URL: https://www.taylorfrancis.com/books/9781351460491.

[43] Gilles Louppe et al. 'Understanding variable importances in forests of randomized trees'. In: *Advances in neural information processing systems*. Ed. by C.J. Burges et al. Vol. 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/e3796ae838835da0b6f6ea37bcf8bcb7-Paper.pdf.

[44] Vicente García, José Salvador Sánchez and Ramón A. Mollineda. 'Exploring the Performance of Resampling Strategies for the Class Imbalance Problem'. In: *Trends in Applied Intelligent Systems*. Ed. by Nicolás García-Pedrajas et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 541–549. ISBN: 978-3-642-13022-9. DOI: 10.1007/978-3-642-13022-9_54.

[45] Gary M. Weiss. 'Foundations of Imbalanced Learning'. In: *Imbalanced Learning*. Ed. by Haibo He and Yunqian Ma. Hoboken, NJ, USA: John Wiley & Sons, Inc., 10th June 2013, pp. 13–41. ISBN: 978-1-118-64610-6. DOI: 10.1002/9781118646106.ch2.

[46] Ivan Tomek. 'Two Modifications of CNN'. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-6.11 (Nov. 1976). Conference Name: IEEE Transactions on Systems, Man, and Cybernetics, pp. 769–772. ISSN: 2168-2909. DOI: 10.1109/TSMC.1976.4309452.

[47] N. V. Chawla et al. 'SMOTE: Synthetic Minority Over-sampling Technique'. In: *Journal of Artificial Intelligence Research* 16 (1st June 2002), pp. 321–357. ISSN: 1076-9757. DOI: 10.1613/jair.953.

[48] Sayash Kapoor and Arvind Narayanan. *Leakage and the Reproducibility Crisis in ML-based Science*. 14th July 2022. DOI: 10.48550/arXiv.2207.07048. arXiv: 2207.07048[cs,stat].

[49] Ravi K. Samala et al. 'Hazards of data leakage in machine learning: a study on classification of breast cancer using deep neural networks'. In: *Medical Imaging 2020: Computer-Aided Diagnosis*. Medical Imaging 2020: Computer-Aided Diagnosis. Vol. 11314. SPIE, 16th Mar. 2020, pp. 279–284. DOI: 10.1117/12.2549313.

[50] Bunty Shah. *Auto Insurance Claims Data*. 20th Aug. 2018. URL: https://www.kaggle.com/datasets/buntyshah/auto-insurance-claims-data.

[51] Sebastián Mauricio Palacio. *Outlier Detection*. Version 1. Publisher: Mendeley Data. 6th Dec. 2018. DOI: 10.17632/g3vxppc8k4.2. URL: https://data.mendeley.com/datasets/g3vxppc8k4/2.

[52] Angoss Knowledge Seeker. *carclaims.csv*. 30th Nov. 2022. URL: https://github.com/Rashmi-77/Vehicle-Insurance-Fraud-Detection.

[53] Angoss Knowledge Seeker. *carclaims.csv*. 16th July 2022. URL: https://www.kaggle.com/datasets/khusheekapoor/vehicle-insurance-fraud-detection.

[54] Chamal Gomes, Zhuo Jin and Hailiang Yang. 'Insurance fraud detection with unsupervised deep learning'. In: *Journal of Risk and Insurance* 88.3 (2021), pp. 591–624. ISSN: 00224367. DOI: 10.1111/jori.12359.

[55] Z. Shaeiri and S.J. Kazemitabar. 'Fast Unsupervised Automobile Insurance Fraud Detection Based on Spectral Ranking of Anomalies'. In: *International Journal of Engineering* 33.7 (2020). Publisher: Materials and Energy Research Center, pp. 1240–1248. ISSN: 17281431. DOI: 10.5829/ije.2020.33.07a.10.

[56] F. Vandervorst, W. Verbeke and T. Verdonck. 'Data misrepresentation detection for insurance underwriting fraud prevention'. In: *Decision Support Systems* 159 (2022). Publisher: Elsevier B.V. ISSN: 01679236. DOI: 10.1016/j.dss.2022.113798.

[57] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. 'Deep learning'. In: *Nature* 521.7553 (May 2015). Number: 7553 Publisher: Nature Publishing Group, pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539.

[58] Anuj Dimri et al. 'Enhancing claims handling processes with insurance based language models'. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. Boca Raton, Florida, USA: IEEE, Dec. 2019, pp. 1750–1755. DOI: 10.1109/ICMLA.2019.00284.

[59] Anuj Dimri et al. 'A multi-input multi-label claims channeling system using insurance-based language models'. In: *Expert Systems with Applications* 202 (2022), p. 117166. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2022.117166.

[60] Jacob Devlin et al. 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL-HLT 2019. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[61] Jeremy Howard and Sebastian Ruder. 'Universal Language Model Fine-tuning for Text Classification'. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ACL 2018. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 328–339. DOI: 10.18653/v1/P18-1031.

[62] Meryem Yankol-Schalck. 'The value of cross-data set analysis for automobile insurance fraud detection'. In: *Research in International Business and Finance* 63 (2022), p. 101769. ISSN: 02755319. DOI: 10.1016/j.ribaf.2022.101769.

[63] Long Zhang et al. 'Auto Insurance Knowledge Graph Construction and Its Application to Fraud Detection'. In: *Proceedings of the 10th International Joint Conference on Knowledge Graphs*. IJCKG '21. Virtual Event: Association for Computing Machinery, 2022, pp. 64–70. ISBN: 978-1-4503-9565-6. DOI: 10.1145/3502223.3502231.

[64] Chuxu Zhang et al. 'Heterogeneous Graph Neural Network'. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. New York, NY, USA: Association for Computing Machinery, 25th July 2019, pp. 793–803. ISBN: 978-1-4503-6201-6. DOI: 10.1145/3292500.3330961.

[65] Aidan Hogan et al. 'Knowledge Graphs'. In: *ACM Computing Surveys* 54.4 (2nd July 2021), 71:1–71:37. ISSN: 0360-0300. DOI: 10.1145/3447772.

[66] Yuri Zelenkov. 'Example-dependent cost-sensitive adaptive boosting'. In: *Expert Systems with Applications* 135 (2019), pp. 71–82. ISSN: 09574174. DOI: 10.1016/j.eswa.2019.06.009.

[67] Yoav Freund and Robert E. Schapire. 'A desicion-theoretic generalization of on-line learning and an application to boosting'. In: *Computational Learning Theory*. Ed. by Paul Vitányi. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1995, pp. 23–37. ISBN: 978-3-540-49195-8. DOI: 10.1007/3-540-59119-2_166.

[68] Michaela Baumann. 'Improving a rule-based fraud detection system with classification based on association rule mining'. In: *INFORMATIK 2021 - Computer Science & Sustainability*. Vol. P-314. Lecture Notes in Informatics (LNI), Proceedings. ISSN: 16175468. Berlin, Germany: Gesellschaft fur Informatik (GI), 2021, pp. 1121–1134. ISBN: 978-3-88579-708-1. DOI: 10.18420/informatik2021-091.

[69] Xi Liu et al. 'Automobile Insurance Fraud Detection using the Evidential Reasoning Approach and Data-Driven Inferential Modelling'. In: *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. Virtual Event: IEEE, July 2020, pp. 1–7. DOI: 10.1109/FUZZ48607.2020.9177589.

[70] Maleeha Qazi et al. 'Discovering temporal patterns from insurance interaction data'. In: *The Thirty-First AAAI Conference on Innovative Applications of Artificial Intelligence (IAAI-19)*. Vol. 33. Proceedings of the AAAI Conference on Artificial Intelligence. Place: Honolulu, Hawaii, USA. Honolulu, Hawaii, USA: AAAI Press, 2019, pp. 9573–9580. ISBN: 978-1-57735-809-1. DOI: 10.1609/aaai.v33i01.33019573.

[71] Neeraj Arora, Glenn Fung and Srinivas Tunuguntla. *T-Patterns in Business*. Rochester, NY, 6th Nov. 2017. DOI: 10.2139/ssrn.3066839.

[72] James C. Bezdek, Robert Ehrlich and William Full. 'FCM: The fuzzy c-means clustering algorithm'. In: *Computers & Geosciences* 10.2 (1st Jan. 1984), pp. 191–203. ISSN: 0098-3004. DOI: 10.1016/0098-3004(84)90020-7.

[73] Santosh Kumar Majhi et al. 'Fuzzy clustering using salp swarm algorithm for automobile insurance fraud detection'. In: *Journal of Intelligent & Fuzzy Systems* 36.3 (Jan. 2019). Place: NLD Publisher: IOS Press, pp. 2333–2344. ISSN: 1064-1246. DOI: 10.3233/JIFS-169944.

[74] Santosh Kumar Majhi. 'Fuzzy clustering algorithm based on modified whale optimization algorithm for automobile insurance fraud detection'. In: *Evolutionary Intelligence* 14.1 (2021), pp. 35–46. ISSN: 18645909. DOI: 10.1007/s12065-019-00260-3.

[75] Sharmila Subudhi and Suvasini Panigrahi. 'Two-Stage Automobile Insurance Fraud Detection by Using Optimized Fuzzy C-Means Clustering and Supervised Learning'. In: *International Journal of Information Security and Privacy (IJISP)* 14.3 (2020), pp. 18–37. ISSN: 19301650. DOI: 10.4018/IJISP.2020070102.

[76] Sharmila Subudhi and Suvasini Panigrahi. 'Use of optimized Fuzzy C-Means clustering and supervised classifiers for automobile insurance fraud detection'. In: *Journal of King Saud University - Computer and Information Sciences* 32.5 (2020). Publisher: King Saud bin Abdulaziz University, pp. 568–575. ISSN: 13191578. DOI: 10.1016/j.jksuci.2017.09.010.

[77] Pierre Baldi. 'Autoencoders, unsupervised learning and deep architectures'. In: *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop - Volume 27*. Vol. 27. UTLW'11. Washington, USA: JMLR.org, 2nd July 2011, pp. 37–50.

[78] Linda L. Golden et al. 'aPRIDIT Unsupervised Classification with Asymmetric Valuation of Variable Discriminatory Worth'. In: *Multivariate Behavioral Research* 55.5 (2020). Publisher: Routledge, pp. 685–703. ISSN: 00273171. DOI: 10.1080/00273171.2019.1665979.

[79] P.L. Brockett et al. 'Fraud classification using principal component analysis of RIDITs'. In: *Journal of Risk and Insurance* 69.3 (2002), pp. 341–371. ISSN: 00224367. DOI: 10.1111/1539-6975.00027.

[80] Gutha Jaya Krishna and Vadlamani Ravi. 'Anomaly Detection Using Modified Differential Evolution: An Application to Banking and Insurance'. In: *Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019)*. SoCPaR 2019. Ed. by Ajith Abraham et al. Vol. 1182. Advances in Intelligent Systems and Computing. ISBN: 9783030493448 Publisher: Springer. Hyderabad, India: Springer International Publishing, 2021, pp. 102–111. DOI: 10.1007/978-3-030-49345-5_11.

[81] Gutha Jaya Krishna and Vadlamani Ravi. 'Outlier Detection using Evolutionary Computing'. In: *Proceedings of the International Conference on Informatics and Analytics*. ICIA-16. New York, NY, USA: Association for Computing Machinery, 25th Aug. 2016, pp. 1–6. ISBN: 978-1-4503-4756-3. DOI: 10.1145/2980258.2980295.

[82] Charu C. Aggarwal and Philip S. Yu. 'Outlier detection for high dimensional data'. In: *ACM SIGMOD Record* 30.2 (1st May 2001), pp. 37–46. ISSN: 0163-5808. DOI: 10.1145/376284.375668.

[83] Ke Nian et al. 'Auto insurance fraud detection using unsupervised spectral ranking for anomaly'. In: *The Journal of Finance and Data Science* 2.1 (1st Mar. 2016), pp. 58–75. ISSN: 2405-9188. DOI: 10.1016/j.jfds.2016.03.001.

[84] Jiaqiu Wang et al. 'Fraud network identification model for insurance industry'. In: *Business Intelligence and Information Technology*. International Conference on Business Intelligence and Information Technology BIIT 2021. Lecture Notes on Data Engineering and Communications Technologies. Publisher: Springer Science and Business Media Deutschland GmbH. Harbin, China: Springer International Publishing, 2022, pp. 276–287. ISBN: 978-3-030-92632-8. DOI: 10.1007/978-3-030-92632-8_27.

[85] Bouzgame Itri et al. 'Empirical Oversampling Threshold Strategy for Machine Learning Performance Optimisation in Insurance Fraud Detection'. In: *International Journal of Advanced Computer Science and Applications* 11.10 (2020). Publisher: Science and Information Organization, pp. 432–437. ISSN: 2158107X. DOI: 10.14569/IJACSA.2020.0111054.

[86] Charles Muranda, Ahmed Ali and Thikozani Shongwe. 'Deep Learning Method for Detecting Fraudulent Motor Insurance Claims Using Unbalanced Data'. In: *2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*. Riga, Latvia: IEEE, Oct. 2021, pp. 1–5. DOI: 10.1109/ITMS52826.2021.9615264.

[87] Haibo He et al. 'ADASYN: Adaptive synthetic sampling approach for imbalanced learning'. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). ISSN: 2161-4407. June 2008, pp. 1322–1328. DOI: 10.1109/IJCNN.2008.4633969.

[88] Prateek Kate, Vadlamani Ravi and Akhilesh Gangwar. 'FinGAN: Chaotic generative adversarial network for analytical customer relationship management in banking and insurance'. In: *Neural Computing & Applications* 35.8 (Nov. 2022). Place: Berlin, Heidelberg Publisher: Springer-Verlag, pp. 6015–6028. ISSN: 0941-0643. DOI: 10.1007/s00521-022-07968-x.

[89] Ian Goodfellow et al. 'Generative adversarial networks'. In: *Communications of the ACM* 63.11 (22nd Oct. 2020), pp. 139–144. ISSN: 0001-0782. DOI: 10.1145/3422622.

[90] Bernhard Schölkopf et al. 'Estimating the Support of a High-Dimensional Distribution'. In: *Neural Computation* 13.7 (1st July 2001), pp. 1443–1471. ISSN: 0899-7667. DOI: 10.1162/089976601750264965.

[91] Tutikura Sreeja Reddy et al. 'An Analysis of Various Algorithmic Behaviors in Detecting a Financial Fraud'. In: *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. Kharagpur, India: IEEE, Oct. 2022, pp. 1–6. ISBN: 978-1-66545-262-5. DOI: 10.1109/ICCCNT54827.2022.9984399.

[92] Sonakshi Harjai, Sunil Kumar Khatri and Gurinder Singh. 'Detecting Fraudulent Insurance Claims Using Random Forests and Synthetic Minority Oversampling Technique'. In: *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*. Mathura, India: IEEE, Nov. 2019, pp. 123–128. DOI: 10.1109/ISCON47742.2019.9036162.

[93] Abhijeet Urunkar et al. 'Fraud detection and analysis for insurance claim using machine learning'. In: *2022 IEEE international conference on signal processing, informatics, communication and energy systems (SPICES)*. 2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES). Vol. 1. Thiruvananthapuram, India: IEEE, Mar. 2022, pp. 406–411. DOI: 10.1109/SPICES52834.2022.9774071.

[94] Iffa Maula Nur Prasasti, Arian Dhini and Enrico Laoh. 'Automobile Insurance Fraud Detection using Supervised Classifiers'. In: *2020 International Workshop on Big Data and Information Security (IWBIS)*. Virtual Event, Indonesia: IEEE, Oct. 2020, pp. 47–52. DOI: 10.1109/IWBIS50925.2020.9255426.

[95] Laiqa Rukhsar et al. 'Prediction of insurance fraud detection using machine learning algorithms'. In: *Mehran University Research Journal of Engineering and Technology* 41.1 (Jan. 2022), pp. 33–40. ISSN: 0254-7219. DOI: 10.22581/muet1982.2201.04.

[96] Serkan Türkeli et al. 'Enemy inside: salesperson fraud detection in the insurance industry'. In: *2020 15th iberian conference on information systems and technologies (CISTI)*. ISSN: 2166-0727. June 2020, pp. 1–5. DOI: 10.23919/CISTI49556.2020.9141105.

[97] Nirmala S.Patil et al. 'Vehicle Insurance Fraud Detection System Using Robotic Process Automation and Machine Learning'. In: *2021 International Conference on Intelligent Technologies (CONIT)*. Hubil, India: Institute of Electrical and Electronics Engineers Inc., June 2021, pp. 1–5. DOI: 10.1109/CONIT51480.2021.9498507.

[98] Naga Ramya Bhamidipati et al. 'ClaimChain: Secure Blockchain Platform for Handling Insurance Claims Processing'. In: *2021 IEEE International Conference on Blockchain (Blockchain)*. Melbourne, Australia: IEEE, Dec. 2021, pp. 55–64. DOI: 10.1109/Blockchain53845.2021.00019.

[99] S. Lloyd. 'Least squares quantization in PCM'. In: *IEEE Transactions on Information Theory* 28.2 (Mar. 1982). Conference Name: IEEE Transactions on Information Theory, pp. 129–137. ISSN: 1557-9654. DOI: 10.1109/TIT.1982.1056489.

[100] Sudipto Guha et al. 'Robust Random Cut Forest Based Anomaly Detection on Streams'. In: *Proceedings of The 33rd International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 1938-7228. PMLR, 11th June 2016, pp. 2712–2721. URL: https://proceedings.mlr.press/v48/guha16.html.

[101] Tianqi Chen and Carlos Guestrin. 'XGBoost: A Scalable Tree Boosting System'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: Association for Computing Machinery, 13th Aug. 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785.

[102] Faheem Aslam et al. 'Insurance fraud detection: Evidence from artificial intelligence and machine learning'. In: *Research in International Business and Finance* 62 (2022). Publisher: Elsevier Ltd. ISSN: 02755319. DOI: 10.1016/j.ribaf.2022.101744.

[103] Bouzgarne Itri et al. 'Performance comparative study of machine learning algorithms for automobile insurance fraud detection'. In: *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*. Marrakech, Morocco: IEEE, Oct. 2019, pp. 1–4. DOI: 10.1109/ICDS47004.2019.8942277.

[104] Michał Piesio, Maria Ganzha and Marcin Paprzycki. 'Applying machine learning to anomaly detection in car insurance sales'. In: *Big data analytics: 8th international conference, BDA 2020, sonepat, india, december 15–18, 2020, proceedings*. Number of pages: 21 Place: Sonepat, India. Berlin, Heidelberg: Springer-Verlag, 2020, pp. 257–277. ISBN: 978-3-030-66664-4. DOI: 10.1007/978-3-030-66665-1_17.

[105] Mohamed Hanafy and Ruixing Ming. 'Using machine learning models to compare various resampling methods in predicting insurance fraud'. In: *Journal of Theoretical and Applied Information Technology* 99.12 (2021). Publisher: Little Lion Scientific, pp. 2819–2833. ISSN: 19928645.

[106] Gustavo E. A. P. A. Batista, Ronaldo C. Prati and Maria Carolina Monard. 'A study of the behavior of several methods for balancing machine learning training data'. In: *ACM SIGKDD Explorations Newsletter* 6.1 (1st June 2004), pp. 20–29. ISSN: 1931-0145. DOI: 10.1145/1007730.1007735.

[107] Mabrouka Salmi and Dalia Atif. 'Using a data mining approach to detect automobile insurance fraud'. In: *Proceedings of the 13th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2021)*. Ed. by Ajith Abraham et al. Vol. 417. Lecture Notes in Networks and Systems. ISSN: 2367-3370. Virtual Event: Springer International Publishing, 2022, pp. 55–66. ISBN: 978-3-030-96302-6. DOI: 10.1007/978-3-030-96302-6_5.

[108] Giovanna Menardi and Nicola Torelli. 'Training and assessing classification rules with imbalanced data'. In: *Data Mining and Knowledge Discovery* 28.1 (1st Jan. 2014), pp. 92–122. ISSN: 1573-756X. DOI: 10.1007/s10618-012-0295-5.

[109] E. Soufiane et al. 'Automobile insurance claims auditing: A comprehensive survey on handling awry datasets'. In: *WITS 2020*. 6th International Conference on Wireless Technologies, Embedded, and Intelligent Systems. Ed. by Bennani, S. et al. Vol. 745. Lecture Notes in Electrical Engineering. ISBN: 9789813368927 Publisher: Springer Science and Business Media Deutschland GmbH. Fez, Morocco: Springer Singapore, 2022, pp. 135–144. DOI: 10.1007/978-981-33-6893-4_13.

[110] Edward A. Bender and S. Gill Williamson. 'Basic Concepts in Graph Theory'. In: *Lists, Decisions and Graphs*. 2010, pp. 147–202. URL: https://cseweb.ucsd.edu/~gill/BWLectSite/Resources/LDGbookCOV.pdf.

[111] Carolin Strobl et al. 'Bias in random forest variable importance measures: Illustrations, sources and a solution'. In: *BMC Bioinformatics* 8.1 (Dec. 2007), p. 25. ISSN: 1471-2105. DOI: 10.1186/1471-2105-8-25.

[112] Miriam Seoane Santos et al. 'Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier]'. In: *IEEE Computational Intelligence Magazine* 13.4 (Nov. 2018), pp. 59–76. ISSN: 1556-6048. DOI: 10.1109/MCI.2018.2866730.

[113] Terence Parr et al. *Beware Default Random Forest Importances*. 26th Mar. 2018. URL: https://explained.ai/rf-importance/ (visited on 13/11/2023).

[114] Christoph Molnar. '8.5 Permutation Feature Importance'. In: *Interpretable Machine Learning*. 2nd ed. 21st Aug. 2023. URL: https://christophm.github.io/interpretable-ml-book.

[115] 'Binary Logistic Regression'. In: Frank E. Harrell Jr. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. 2nd ed. Springer Series in Statistics. Cham: Springer International Publishing, 2015, pp. 219–274. ISBN: 978-3-319-19425-7. DOI: 10.1007/978-3-319-19425-7.

[116] Tom Minka. 'A comparison of numerical optimizers for logistic regression'. In: Mar. 2003. URL: https://www.microsoft.com/en-us/research/publication/comparison-numerical-optimizers-logistic-regression/.

[117] Lawrence Page et al. 'The PageRank Citation Ranking : Bringing Order to the Web'. In: The Web Conference. 11th Nov. 1999. URL: https://www.semanticscholar.org/paper/The-PageRank-Citation-Ranking-%3A-Bringing-Order-to-Page-Brin/eb82d3035849cd23578096462ba419b53198a556.

[118] Jerome Fan, Suneel Upadhye and Andrew Worster. 'Understanding receiver operating characteristic (ROC) curves'. In: *Canadian Journal of Emergency Medicine* 8.1 (Jan. 2006). Publisher: Cambridge University Press, pp. 19–20. ISSN: 1481-8035, 1481-8043. DOI: 10.1017/S1481803500013336.

[119] Jesse Davis and Mark Goadrich. 'The relationship between Precision-Recall and ROC curves'. In: *Proceedings of the 23rd international conference on Machine learning*. ICML '06. New York, NY, USA: Association for Computing Machinery, 25th June 2006, pp. 233–240. ISBN: 978-1-59593-383-6. DOI: 10.1145/1143844.1143874.

[120] Paula Branco, Luís Torgo and Rita P. Ribeiro. 'A Survey of Predictive Modeling on Imbalanced Domains'. In: *ACM Computing Surveys* 49.2 (13th Aug. 2016), 31:1–31:50. ISSN: 0360-0300. DOI: 10.1145/2907070.

[121] Peter Flach and Meelis Kull. 'Precision-Recall-Gain Curves: PR Analysis Done Right'. In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/hash/33e8075e9970de0cfea955afd4644bb2-Abstract.html.

[122] Aurélie Lemmens and Christophe Croux. 'Bagging and Boosting Classification Trees to Predict Churn'. In: *Journal of Marketing Research* 43.2 (1st May 2006). Publisher: SAGE Publications Inc, pp. 276–286. ISSN: 0022-2437. DOI: 10.1509/jmkr.43.2.276.

[123] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. 'Evaluation in Information Retrieval'. In: *Introduction to information retrieval*. Cambridge: Cambridge University Press, 2008, pp. 151–175. ISBN: 978-0-521-86571-5. URL: https://nlp.stanford.edu/IR-book/.

[124] Apache Software Foundation. *Apache Hive*. URL: https://hive.apache.org/.

[125] Apache Software Foundation. *Apache Spark*. Version 3.3.1. 25th Oct. 2022. URL: https://spark.apache.org/docs/3.3.1/.

[126] Apache Software Foundation. *PySpark*. Version 3.4.1. 23rd June 2023. URL: https://spark.apache.org/docs/3.4.1/api/python/index.html#.

[127] Apache Software Foundation. *Apache Parquet*. Version 2.9. URL: https://parquet.apache.org/.

[128] The pandas development team. *pandas-dev/pandas: Pandas*. Version v1.5.3. Jan. 2023. DOI: 10.5281/zenodo.7549438. URL: https://doi.org/10.5281/zenodo.7549438.

[129] Wes McKinney. 'Data Structures for Statistical Computing in Python'. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[130] Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart. 'Exploring network structure, dynamics, and function using NetworkX'. In: *Proceedings of the 7th python in science conference*. Ed. by Gaël Varoquaux, Travis Vaught and Jarrod Millman. Pasadena, CA USA, 2008, pp. 11–15.

[131] Anshul Gupta and Toyotaro Suzumura. *Finding All Bounded-Length Simple Cycles in a Directed Graph*. 26th May 2021. DOI: 10.48550/arXiv.2105.10094. arXiv: 2105.10094[cs].

[132] Student. 'The Probable Error of a Mean'. In: *Biometrika* 6.1 (Mar. 1908), p. 1. ISSN: 00063444. DOI: 10.2307/2331554.

[133] B. L. Welch. 'The generalisation of 'Student's' problem when several different population variances are involved'. In: *Biometrika* 34.1 (1947), pp. 28–35. ISSN: 0006-3444, 1464-3510. DOI: 10.1093/biomet/34.1-2.28.

[134] Pauli Virtanen et al. 'SciPy 1.0: Fundamental algorithms for scientific computing in python'. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[135] Brian Aronson. *BiRank R and Python package*. Version 1.0. 3rd Nov. 2023. URL: https://github.com/BrianAronson/birankr.

[136] Charles R. Harris et al. 'Array programming with NumPy'. In: *Nature* 585.7825 (Sept. 2020). Publisher: Springer Science and Business Media LLC, pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[137] Rob J. Hyndman and Yanan Fan. 'Sample Quantiles in Statistical Packages'. In: *The American Statistician* 50.4 (1996). Publisher: [American Statistical Association, Taylor & Francis, Ltd.], pp. 361–365. ISSN: 0003-1305. DOI: 10.2307/2684934.

[138] Mike Van Ness et al. 'The Missing Indicator Method: From Low to High Dimensions'. In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '23. New York, NY, USA: Association for Computing Machinery, 4th Aug. 2023, pp. 5004–5015. ISBN: 9798400701030. DOI: 10.1145/3580305.3599911.

[139] John W. Graham. 'Missing Data Analysis: Making It Work in the Real World'. In: *Annual Review of Psychology* 60.1 (1st Jan. 2009), pp. 549–576. DOI: 10.1146/annurev.psych.58.110405.085530.

[140] Therese D. Pigott. 'A Review of Methods for Missing Data'. In: *Educational Research and Evaluation* 7.4 (1st Dec. 2001). Publisher: Routledge, pp. 353–383. ISSN: 1380-3611. DOI: 10.1076/edre.7.4.353.8937.

[141] F. Pedregosa et al. 'Scikit-learn: Machine learning in Python'. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[142] Guillaume Lemaître, Fernando Nogueira and Christos K. Aridas. 'Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning'. In: *Journal of Machine Learning Research* 18.17 (2017), pp. 1–5. URL: http://jmlr.org/papers/v18/16-365.html.

[143] Skipper Seabold and Josef Perktold. 'statsmodels: Econometric and statistical modeling with python'. In: *9th python in science conference*. 2010.

[144] Dong C. Liu and Jorge Nocedal. 'On the limited memory BFGS method for large scale optimization'. In: *Mathematical Programming* 45.1 (1st Aug. 1989), pp. 503–528. ISSN: 1436-4646. DOI: 10.1007/BF01589116.

[145] Insurance Fraud Bureau. *Frequently Asked Questions*. URL: https://www.theifr.org.uk/en/faqs (visited on 23/12/2023).

[146] Stichting Centraal Informatie Systeem. *Participating insurance companies*. URL: https://stichtingcis.nl/en-us/Members/Participating-insurance-companies (visited on 23/12/2023).

[147] Aylin Alin. 'Multicollinearity'. In: *WIREs Computational Statistics* 2.3 (2010). _eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.84, pp. 370–374. ISSN: 1939-0068. DOI: 10.1002/wics.84.

[148] Lingfei Wu et al., eds. *Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer Nature, 2022. ISBN: 9789811660542. DOI: 10.1007/978-981-16-6054-2.

Appendix A

Grid Search Results

Table A.1: Grid search results for D^{known} using intrinsic features

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
900	7	0.2815	0.0573	0.7927	0.0319	4.7800	0.6230
900	5	0.2811	0.0530	0.7969	0.0307	4.8472	0.4749
500	7	0.2807	0.0588	0.7921	0.0298	4.7627	0.6007
700	5	0.2807	0.0545	0.7963	0.0316	4.8302	0.4813
700	7	0.2804	0.0569	0.7926	0.0305	4.7802	0.6553
900	9	0.2801	0.0515	0.7922	0.0341	4.7125	0.5740
700	9	0.2801	0.0519	0.7925	0.0335	4.7122	0.5863
300	5	0.2799	0.0563	0.7938	0.0323	4.7969	0.6276
500	9	0.2797	0.0518	0.7917	0.0327	4.6447	0.6276
500	5	0.2794	0.0568	0.7947	0.0329	4.7966	0.4650
300	7	0.2779	0.0575	0.7906	0.0306	4.6958	0.6060
900	11	0.2773	0.0529	0.7893	0.0329	4.7450	0.6014
300	9	0.2772	0.0531	0.7899	0.0337	4.7461	0.5306
500	15	0.2764	0.0527	0.7879	0.0322	4.7783	0.5874
700	11	0.2763	0.0525	0.7888	0.0324	4.7794	0.5298
700	17	0.2756	0.0539	0.7854	0.0303	4.8622	0.5664
900	17	0.2756	0.0545	0.7859	0.0302	4.7619	0.5500
900	15	0.2754	0.0524	0.7872	0.0313	4.7278	0.5979
500	17	0.2753	0.0549	0.7856	0.0303	4.8289	0.6183
900	19	0.2753	0.0542	0.7839	0.0334	4.7947	0.6024
500	11	0.2752	0.0536	0.7873	0.0315	4.7789	0.5608
500	19	0.2751	0.0549	0.7833	0.0335	4.7447	0.6368
300	15	0.2751	0.0514	0.7869	0.0330	4.6944	0.5999
900	13	0.2750	0.0535	0.7898	0.0320	4.8463	0.5906
300	13	0.2749	0.0522	0.7873	0.0321	4.8805	0.5019
100	9	0.2749	0.0512	0.7863	0.0329	4.8635	0.5261
700	13	0.2748	0.0533	0.7887	0.0330	4.7963	0.6013
300	19	0.2747	0.0537	0.7828	0.0344	4.8122	0.6029
700	15	0.2747	0.0519	0.7875	0.0315	4.7450	0.6254
500	13	0.2742	0.0530	0.7884	0.0328	4.8130	0.5701
700	19	0.2742	0.0539	0.7834	0.0337	4.7788	0.5252
100	19	0.2741	0.0520	0.7810	0.0312	4.7947	0.6287
100	5	0.2732	0.0515	0.7867	0.0316	4.8141	0.5231
300	17	0.2732	0.0556	0.7846	0.0306	4.8294	0.5501
700	3	0.2724	0.0530	0.7960	0.0316	4.5609	0.5111
300	11	0.2723	0.0533	0.7875	0.0326	4.7622	0.5582
900	3	0.2717	0.0531	0.7959	0.0313	4.5778	0.4970
500	3	0.2716	0.0536	0.7967	0.0310	4.5447	0.4309
100	13	0.2715	0.0535	0.7815	0.0329	4.7452	0.5181

Table A.1: Grid search results for D^{known}, limited to intrinsic features (cont.)

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
900	21	0.2709	0.0537	0.7835	0.0310	4.7441	0.6639
300	21	0.2708	0.0537	0.7814	0.0353	4.7105	0.7142
700	21	0.2708	0.0536	0.7836	0.0317	4.7105	0.6817
500	21	0.2693	0.0538	0.7822	0.0337	4.7444	0.6908
300	3	0.2693	0.0527	0.7951	0.0321	4.5109	0.5073
100	15	0.2683	0.0514	0.7807	0.0326	4.7280	0.5796
100	11	0.2679	0.0538	0.7819	0.0313	4.7794	0.6366
900	23	0.2678	0.0547	0.7822	0.0307	4.7275	0.6277
100	17	0.2671	0.0557	0.7814	0.0292	4.7280	0.4843
100	7	0.2667	0.0622	0.7839	0.0286	4.7458	0.5701
100	21	0.2665	0.0582	0.7753	0.0329	4.7275	0.6844
700	23	0.2662	0.0531	0.7803	0.0303	4.6772	0.6164
300	23	0.2650	0.0555	0.7765	0.0309	4.5931	0.6496
500	23	0.2647	0.0537	0.7785	0.0316	4.6606	0.6460
100	3	0.2609	0.0567	0.7870	0.0355	4.4939	0.4848
100	23	0.2607	0.0549	0.7709	0.0336	4.5089	0.6965
900	1	0.2486	0.0533	0.7892	0.0339	4.4603	0.5085
700	1	0.2473	0.0524	0.7899	0.0334	4.5445	0.5062
500	1	0.2453	0.0495	0.7893	0.0332	4.5614	0.4929
300	1	0.2432	0.0504	0.7884	0.0326	4.4778	0.5488
100	1	0.2408	0.0551	0.7840	0.0313	4.4600	0.3934

Table A.2: Grid search results for D^{known} using neighbourhood features

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
900	5	0.1797	0.0245	0.7830	0.0282	3.7530	0.6563
900	3	0.1781	0.0264	0.7852	0.0278	3.8874	0.7110
700	3	0.1776	0.0274	0.7837	0.0280	3.8871	0.7445
500	5	0.1774	0.0244	0.7811	0.0300	3.7866	0.6990
700	5	0.1774	0.0241	0.7812	0.0291	3.7524	0.6312
300	5	0.1765	0.0240	0.7786	0.0307	3.8035	0.6657
100	5	0.1762	0.0256	0.7766	0.0324	3.7355	0.6725
500	3	0.1762	0.0266	0.7826	0.0276	3.8701	0.6828
300	3	0.1738	0.0249	0.7809	0.0288	3.8868	0.6242
100	3	0.1715	0.0263	0.7734	0.0313	3.8193	0.6275
900	1	0.1664	0.0251	0.7777	0.0270	3.7857	0.6780
700	1	0.1654	0.0248	0.7768	0.0259	3.8024	0.6369
500	1	0.1647	0.0254	0.7760	0.0264	3.8196	0.6437
300	1	0.1636	0.0257	0.7721	0.0282	3.8196	0.6347
100	1	0.1628	0.0300	0.7636	0.0308	3.7183	0.5570

Table A.3: Grid search results for D^{known} using score features

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
900	1	0.1003	0.0198	0.6846	0.0183	2.3884	0.3093
700	3	0.1001	0.0201	0.6848	0.0188	2.1197	0.5616
100	3	0.1001	0.0222	0.6785	0.0197	2.4054	0.5933
700	1	0.0998	0.0200	0.6835	0.0182	2.3213	0.4059
500	1	0.0996	0.0201	0.6825	0.0175	2.1538	0.3074
300	3	0.0995	0.0196	0.6836	0.0176	2.2035	0.5031
900	3	0.0994	0.0201	0.6843	0.0194	2.1199	0.5412
500	3	0.0994	0.0199	0.6841	0.0191	2.0530	0.3671

Table A.4: Grid search results for D^{known} using all features

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
700	9	0.3028	0.0560	0.8399	0.0173	5.4350	0.4333
900	11	0.3025	0.0562	0.8381	0.0172	5.4017	0.4480
300	9	0.3022	0.0566	0.8374	0.0196	5.3175	0.4139
500	9	0.3016	0.0561	0.8393	0.0175	5.3842	0.4287
900	9	0.3016	0.0548	0.8395	0.0174	5.3845	0.4515
900	7	0.3009	0.0541	0.8423	0.0173	5.4697	0.5291
700	11	0.3009	0.0571	0.8377	0.0174	5.3847	0.4172
500	7	0.3005	0.0562	0.8424	0.0165	5.3522	0.5242
500	11	0.3004	0.0572	0.8368	0.0176	5.4186	0.4511
700	7	0.3003	0.0562	0.8426	0.0172	5.4864	0.5055
500	5	0.2992	0.0551	0.8411	0.0176	5.3678	0.4885
700	15	0.2991	0.0571	0.8366	0.0202	5.3167	0.4234
300	11	0.2989	0.0557	0.8345	0.0192	5.4011	0.4712
700	5	0.2986	0.0548	0.8419	0.0179	5.3856	0.4591
900	13	0.2976	0.0574	0.8363	0.0183	5.1826	0.4061
500	15	0.2975	0.0562	0.8349	0.0204	5.3173	0.4303
900	5	0.2974	0.0551	0.8427	0.0175	5.4525	0.4928
900	15	0.2973	0.0564	0.8362	0.0200	5.3003	0.4486
700	13	0.2967	0.0562	0.8357	0.0180	5.2165	0.3593
300	5	0.2964	0.0570	0.8403	0.0186	5.3850	0.5284
300	7	0.2962	0.0537	0.8401	0.0159	5.3522	0.5077
700	19	0.2958	0.0540	0.8326	0.0204	5.1993	0.4288
900	19	0.2958	0.0539	0.8330	0.0205	5.2329	0.4786
500	13	0.2957	0.0566	0.8357	0.0181	5.1823	0.4305
500	19	0.2957	0.0547	0.8313	0.0203	5.2664	0.3600
100	9	0.2957	0.0583	0.8299	0.0189	5.2334	0.5063
300	15	0.2954	0.0580	0.8344	0.0227	5.3339	0.4762
900	17	0.2953	0.0541	0.8337	0.0202	5.2837	0.4499
300	19	0.2944	0.0524	0.8307	0.0203	5.2162	0.4404
700	17	0.2941	0.0534	0.8334	0.0211	5.2498	0.4698
300	13	0.2937	0.0542	0.8372	0.0193	5.2670	0.3587
500	17	0.2933	0.0550	0.8322	0.0221	5.2673	0.5185
900	23	0.2932	0.0538	0.8314	0.0225	5.2168	0.5884
700	21	0.2931	0.0505	0.8313	0.0219	5.2331	0.4819
900	21	0.2929	0.0509	0.8323	0.0215	5.2498	0.4575
700	23	0.2927	0.0545	0.8321	0.0226	5.1496	0.5809
500	21	0.2921	0.0509	0.8305	0.0209	5.1993	0.4358
300	17	0.2917	0.0544	0.8305	0.0225	5.1323	0.4564
500	23	0.2917	0.0542	0.8315	0.0227	5.1157	0.5740
700	25	0.2909	0.0514	0.8307	0.0193	5.1826	0.5581
900	25	0.2907	0.0513	0.8314	0.0198	5.1823	0.5562
300	21	0.2902	0.0508	0.8301	0.0206	5.1484	0.3500
500	25	0.2900	0.0517	0.8307	0.0195	5.1154	0.5463
300	25	0.2899	0.0500	0.8299	0.0194	5.0985	0.5883
300	23	0.2897	0.0507	0.8307	0.0218	5.2329	0.5192
700	27	0.2897	0.0525	0.8303	0.0211	5.1154	0.5809
900	27	0.2895	0.0522	0.8308	0.0206	5.1326	0.5597
100	11	0.2888	0.0551	0.8316	0.0173	5.2167	0.3910
700	33	0.2878	0.0509	0.8284	0.0194	5.1157	0.6081
900	3	0.2876	0.0524	0.8436	0.0159	5.3189	0.5665
500	27	0.2876	0.0520	0.8289	0.0214	5.1162	0.5783
500	3	0.2875	0.0528	0.8427	0.0159	5.2511	0.4618
100	15	0.2873	0.0617	0.8299	0.0260	5.1826	0.3447
700	3	0.2871	0.0530	0.8434	0.0163	5.2850	0.5234
900	29	0.2869	0.0488	0.8305	0.0211	5.1826	0.5643
900	31	0.2868	0.0517	0.8282	0.0236	4.9968	0.5895

Table A.4: Grid search results for D^{known} with all features (cont.)

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
300	29	0.2867	0.0496	0.8294	0.0224	5.1323	0.5373
700	31	0.2865	0.0518	0.8283	0.0244	4.9632	0.6188
100	21	0.2865	0.0513	0.8276	0.0220	5.0313	0.4794
900	33	0.2863	0.0510	0.8289	0.0204	5.0987	0.5774
500	29	0.2862	0.0487	0.8298	0.0217	5.1651	0.5054
700	29	0.2860	0.0481	0.8308	0.0218	5.1823	0.5715
500	33	0.2856	0.0513	0.8282	0.0206	5.1329	0.5054
300	3	0.2855	0.0537	0.8412	0.0150	5.2173	0.5018
300	27	0.2854	0.0497	0.8280	0.0228	5.1334	0.6285
100	29	0.2854	0.0516	0.8279	0.0247	5.1660	0.6101
100	7	0.2849	0.0555	0.8291	0.0167	5.1331	0.5289
300	31	0.2849	0.0520	0.8259	0.0242	4.9966	0.5520
100	17	0.2848	0.0549	0.8229	0.0199	5.0321	0.4352
700	35	0.2845	0.0510	0.8264	0.0208	5.0310	0.5846
500	31	0.2845	0.0512	0.8283	0.0251	4.9463	0.5498
100	5	0.2844	0.0576	0.8305	0.0193	5.0320	0.5333
500	35	0.2843	0.0506	0.8263	0.0205	5.0141	0.6141
100	13	0.2842	0.0571	0.8299	0.0221	5.1162	0.4510
300	33	0.2837	0.0503	0.8278	0.0201	5.1323	0.5001
900	35	0.2837	0.0501	0.8266	0.0208	5.0477	0.5769
100	19	0.2831	0.0431	0.8295	0.0224	5.0981	0.3894
100	25	0.2830	0.0533	0.8253	0.0177	5.0985	0.5110
100	3	0.2829	0.0601	0.8325	0.0161	5.1837	0.4644
100	31	0.2827	0.0522	0.8185	0.0251	4.9799	0.5838
300	35	0.2825	0.0493	0.8262	0.0206	5.0643	0.5210
100	35	0.2813	0.0470	0.8234	0.0195	5.0818	0.5815
100	27	0.2793	0.0491	0.8221	0.0232	5.1337	0.5976
100	23	0.2784	0.0485	0.8246	0.0216	5.0151	0.4783
100	33	0.2770	0.0458	0.8234	0.0211	5.0479	0.5634
700	1	0.2592	0.0510	0.8338	0.0158	4.7630	0.5281
500	1	0.2590	0.0526	0.8332	0.0167	4.7127	0.5087
900	1	0.2586	0.0525	0.8342	0.0159	4.8305	0.6192
300	1	0.2559	0.0511	0.8303	0.0151	4.7291	0.4416
100	1	0.2450	0.0487	0.8197	0.0159	4.7466	0.6262

Table A.5: Grid search results for D^{fraud} using intrinsic features

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
900	5	0.1550	0.0465	0.7698	0.0306	4.2754	0.9531
900	7	0.1546	0.0466	0.7664	0.0342	4.2486	0.7971
700	5	0.1544	0.0462	0.7702	0.0317	4.2767	0.9167
500	7	0.1543	0.0476	0.7647	0.0369	4.2479	0.7872
300	7	0.1536	0.0467	0.7619	0.0384	4.2492	0.8474
700	7	0.1533	0.0463	0.7655	0.0345	4.2479	0.8629
500	5	0.1533	0.0462	0.7691	0.0320	4.1736	0.9475
700	3	0.1529	0.0466	0.7713	0.0308	4.2736	0.9565
300	5	0.1527	0.0475	0.7677	0.0362	4.1979	0.7863
900	3	0.1527	0.0471	0.7720	0.0314	4.3498	0.9128
300	3	0.1526	0.0450	0.7641	0.0302	4.3517	0.8389
900	9	0.1524	0.0454	0.7666	0.0325	4.0967	0.8897
500	3	0.1523	0.0454	0.7699	0.0327	4.2992	0.9270
700	1	0.1519	0.0492	0.7710	0.0306	4.2973	0.8590

Table A.5: Grid search results for D^{fraud} with intrinsic features (cont.)

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
900	11	0.1514	0.0439	0.7593	0.0369	4.1217	0.8735
500	9	0.1513	0.0461	0.7653	0.0338	4.0955	0.8484
700	9	0.1511	0.0449	0.7662	0.0336	4.0455	0.9653
900	15	0.1505	0.0475	0.7599	0.0364	4.0955	0.8320
100	7	0.1503	0.0488	0.7526	0.0404	4.0705	0.9827
900	1	0.1499	0.0487	0.7718	0.0303	4.2717	0.8531
700	17	0.1496	0.0430	0.7537	0.0374	4.0711	0.7833
500	1	0.1493	0.0481	0.7710	0.0306	4.1961	0.7645
700	11	0.1493	0.0428	0.7586	0.0357	3.9705	0.7757
300	11	0.1492	0.0448	0.7550	0.0363	3.9698	0.8074
300	9	0.1491	0.0460	0.7656	0.0332	4.0948	0.7701
300	15	0.1489	0.0443	0.7615	0.0384	4.1211	0.8251
900	13	0.1483	0.0456	0.7595	0.0347	4.0717	0.9056
100	13	0.1483	0.0451	0.7544	0.0403	4.0455	0.8724
500	17	0.1482	0.0422	0.7534	0.0377	4.0448	0.7319
700	15	0.1482	0.0455	0.7611	0.0363	3.9942	0.7582
900	17	0.1482	0.0445	0.7541	0.0392	4.0705	0.8581
300	17	0.1479	0.0445	0.7499	0.0392	4.0198	0.8366
500	11	0.1479	0.0455	0.7583	0.0367	4.0217	0.7741
700	13	0.1479	0.0458	0.7583	0.0348	4.0723	0.8735
900	19	0.1477	0.0437	0.7515	0.0375	4.1205	0.8510
500	15	0.1475	0.0451	0.7613	0.0356	4.1717	0.8825
300	13	0.1475	0.0457	0.7573	0.0395	3.9955	0.7620
300	1	0.1473	0.0460	0.7680	0.0306	4.1961	0.6866
500	13	0.1468	0.0461	0.7591	0.0361	4.0473	0.7431
500	19	0.1468	0.0460	0.7512	0.0397	4.1723	0.8723
700	19	0.1467	0.0446	0.7515	0.0384	4.1467	0.7620
300	19	0.1463	0.0459	0.7485	0.0411	4.0198	0.8328
700	21	0.1462	0.0415	0.7492	0.0407	4.1986	0.6807
900	21	0.1461	0.0416	0.7491	0.0399	4.1229	0.7597
500	21	0.1446	0.0413	0.7490	0.0424	4.1992	0.8583
300	21	0.1446	0.0419	0.7474	0.0401	4.1467	0.7267
100	17	0.1444	0.0430	0.7477	0.0458	4.0467	0.8685
100	21	0.1441	0.0437	0.7447	0.0421	3.9936	0.8042
100	11	0.1440	0.0456	0.7481	0.0371	3.9692	0.7661
100	3	0.1437	0.0457	0.7579	0.0359	4.3286	1.0357
100	9	0.1433	0.0463	0.7548	0.0345	4.1967	0.9255
700	23	0.1432	0.0444	0.7485	0.0389	4.1979	0.7957
100	15	0.1429	0.0454	0.7526	0.0312	4.0698	0.9082
900	23	0.1427	0.0445	0.7474	0.0404	4.1986	0.7678
100	1	0.1424	0.0468	0.7520	0.0326	4.0942	0.6993
100	5	0.1423	0.0436	0.7583	0.0420	3.9923	0.8849
300	23	0.1420	0.0453	0.7433	0.0408	4.0961	0.7865
500	23	0.1410	0.0443	0.7463	0.0415	4.1992	0.8844
100	23	0.1401	0.0437	0.7442	0.0378	4.0698	0.7634
100	19	0.1397	0.0428	0.7452	0.0416	4.0461	0.8787

Table A.6: Grid search results for D^{fraud} using neighbourhood features

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
900	5	0.1677	0.0508	0.7772	0.0259	4.0111	0.9006
300	5	0.1667	0.0521	0.7706	0.0282	4.0367	0.8542
700	5	0.1660	0.0512	0.7772	0.0263	4.0880	0.9013
500	5	0.1656	0.0500	0.7762	0.0244	4.1136	0.8829
900	3	0.1621	0.0540	0.7785	0.0280	4.0873	0.8966
700	3	0.1618	0.0537	0.7788	0.0281	4.1123	0.9592
500	3	0.1614	0.0532	0.7765	0.0289	4.0617	0.8995
300	3	0.1608	0.0533	0.7748	0.0299	4.0617	0.8995
100	5	0.1582	0.0537	0.7625	0.0302	4.0123	0.7754
100	3	0.1569	0.0540	0.7676	0.0334	4.0117	0.8443
700	1	0.1439	0.0485	0.7698	0.0270	3.9361	0.7974
500	1	0.1438	0.0470	0.7690	0.0260	3.9361	0.8454
900	1	0.1425	0.0478	0.7697	0.0267	3.9361	0.8218
300	1	0.1417	0.0465	0.7668	0.0258	3.8336	0.8426
100	1	0.1390	0.0477	0.7636	0.0274	3.8849	0.8099

Table A.7: Grid search results for D^{fraud} using score features

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
300	5	0.0745	0.0228	0.6736	0.0253	2.3911	0.5518
500	5	0.0744	0.0223	0.6768	0.0234	2.3143	0.5619
900	5	0.0742	0.0215	0.6798	0.0211	2.3143	0.5641
700	5	0.0735	0.0207	0.6787	0.0214	2.2880	0.6719
100	7	0.0734	0.0219	0.6734	0.0248	2.3399	0.5338
700	1	0.0734	0.0192	0.6836	0.0206	2.0849	0.7270
900	7	0.0734	0.0216	0.6776	0.0263	2.3393	0.4365
700	3	0.0732	0.0208	0.6795	0.0205	2.2880	0.5563
700	7	0.0730	0.0220	0.6757	0.0270	2.4911	0.5142
300	7	0.0728	0.0217	0.6746	0.0284	2.3137	0.5380
500	7	0.0728	0.0218	0.6752	0.0279	2.2612	0.4905
900	3	0.0728	0.0204	0.6805	0.0217	2.3893	0.5781
300	1	0.0724	0.0187	0.6825	0.0195	2.1605	0.3718
500	3	0.0723	0.0203	0.6790	0.0218	2.3649	0.3926
100	1	0.0723	0.0180	0.6807	0.0218	2.4155	0.6266
300	3	0.0722	0.0208	0.6774	0.0227	2.3911	0.4480
100	5	0.0722	0.0238	0.6668	0.0297	2.1862	0.5182
500	1	0.0720	0.0187	0.6834	0.0200	2.2899	0.5859
900	1	0.0718	0.0182	0.6841	0.0208	2.0349	0.5718
100	3	0.0695	0.0211	0.6676	0.0275	2.2087	0.4763

Table A.8: Grid search results for D^{fraud} using all features

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
900	3	0.1862	0.0457	0.8254	0.0227	5.2135	0.6567
700	3	0.1848	0.0474	0.8237	0.0241	5.0860	0.5907
500	3	0.1826	0.0464	0.8213	0.0244	5.0873	0.5536
900	5	0.1821	0.0417	0.8274	0.0220	5.1123	0.5598
700	5	0.1810	0.0400	0.8276	0.0227	5.1379	0.5171
500	5	0.1807	0.0419	0.8269	0.0221	5.2392	0.5675
300	3	0.1802	0.0452	0.8201	0.0261	5.0617	0.5605
300	5	0.1794	0.0423	0.8265	0.0227	5.0860	0.6336
300	1	0.1781	0.0383	0.8151	0.0223	4.7311	0.4455
500	7	0.1780	0.0389	0.8225	0.0222	4.8323	0.6304
700	7	0.1775	0.0390	0.8240	0.0215	4.8073	0.6478
500	1	0.1772	0.0404	0.8160	0.0234	4.8323	0.5516
900	7	0.1765	0.0391	0.8254	0.0205	4.8073	0.6478
900	9	0.1765	0.0415	0.8226	0.0212	4.6786	0.6709
900	1	0.1760	0.0422	0.8155	0.0227	4.8579	0.5586
100	3	0.1757	0.0468	0.8115	0.0308	4.9354	0.6133
500	9	0.1753	0.0415	0.8222	0.0210	4.7804	0.7524
300	9	0.1749	0.0427	0.8235	0.0201	4.7036	0.6888
100	5	0.1741	0.0441	0.8159	0.0290	4.8335	0.5895
300	7	0.1741	0.0365	0.8206	0.0234	4.7561	0.5985
700	1	0.1741	0.0419	0.8162	0.0221	4.7823	0.5220
700	9	0.1736	0.0392	0.8219	0.0211	4.7548	0.6504
100	7	0.1726	0.0368	0.8127	0.0276	4.8335	0.4775
100	1	0.1722	0.0376	0.8051	0.0241	4.8817	0.4601
900	11	0.1710	0.0396	0.8229	0.0233	4.6286	0.4992
700	11	0.1707	0.0403	0.8222	0.0245	4.6017	0.5972
500	11	0.1707	0.0410	0.8226	0.0246	4.6536	0.5924
300	11	0.1706	0.0408	0.8210	0.0245	4.6536	0.5774
500	13	0.1699	0.0429	0.8175	0.0262	4.7054	0.6229
900	13	0.1692	0.0416	0.8199	0.0249	4.6286	0.6378
500	17	0.1690	0.0423	0.8161	0.0253	4.6304	0.5529
700	13	0.1690	0.0419	0.8191	0.0255	4.6292	0.6907
300	13	0.1689	0.0428	0.8168	0.0271	4.6548	0.6454
900	19	0.1679	0.0412	0.8179	0.0246	4.6292	0.6487
700	19	0.1677	0.0409	0.8175	0.0241	4.7048	0.5977
900	17	0.1677	0.0419	0.8152	0.0251	4.6811	0.5682
700	17	0.1671	0.0401	0.8160	0.0243	4.6042	0.5182
500	19	0.1670	0.0413	0.8171	0.0247	4.7054	0.6334
900	21	0.1662	0.0404	0.8158	0.0235	4.7317	0.6010
700	21	0.1657	0.0401	0.8149	0.0237	4.7054	0.5847
300	19	0.1657	0.0415	0.8164	0.0257	4.7048	0.6298
900	15	0.1656	0.0376	0.8170	0.0254	4.6298	0.6451
500	15	0.1654	0.0380	0.8163	0.0262	4.5517	0.5947
300	17	0.1653	0.0361	0.8146	0.0241	4.7567	0.5138
500	21	0.1640	0.0397	0.8150	0.0234	4.7304	0.5698
100	13	0.1637	0.0451	0.8044	0.0328	4.5536	0.7449
100	11	0.1635	0.0370	0.8110	0.0259	4.6548	0.5541
500	23	0.1635	0.0345	0.8132	0.0213	4.7561	0.6568
100	9	0.1635	0.0389	0.8138	0.0223	4.6023	0.5136
700	15	0.1633	0.0355	0.8157	0.0258	4.5267	0.6225
700	23	0.1630	0.0359	0.8130	0.0216	4.7811	0.6218
900	23	0.1629	0.0353	0.8138	0.0224	4.8317	0.6431
100	19	0.1629	0.0405	0.8165	0.0257	4.7823	0.6563
300	15	0.1625	0.0358	0.8144	0.0278	4.5786	0.5329
300	21	0.1623	0.0373	0.8173	0.0241	4.7567	0.5833
700	25	0.1620	0.0370	0.8126	0.0250	4.6779	0.6475

Table A.8: Grid search results for D^{fraud} with all features (cont.)

		AUC-PR		AUC-ROC		Top Decile Lift	
No. of trees	Features per split	Mean	Std.	Mean	Std.	Mean	Std.
900	27	0.1620	0.0363	0.8124	0.0242	4.7048	0.4355
700	27	0.1619	0.0357	0.8117	0.0255	4.7048	0.4355
500	25	0.1618	0.0362	0.8119	0.0255	4.7292	0.5640
100	21	0.1615	0.0368	0.8086	0.0211	4.6298	0.6772
300	23	0.1609	0.0355	0.8114	0.0228	4.5798	0.6121
900	25	0.1608	0.0351	0.8134	0.0243	4.6273	0.6642
300	27	0.1603	0.0335	0.8110	0.0275	4.7298	0.3825
500	27	0.1600	0.0337	0.8119	0.0255	4.6542	0.4485
300	25	0.1595	0.0350	0.8089	0.0264	4.7036	0.6077
100	17	0.1589	0.0396	0.8035	0.0240	4.7073	0.7782
100	15	0.1582	0.0350	0.8089	0.0239	4.6298	0.6416
900	29	0.1577	0.0342	0.8110	0.0242	4.8573	0.5636
100	27	0.1575	0.0344	0.8060	0.0263	4.7548	0.3764
700	29	0.1569	0.0324	0.8095	0.0249	4.7561	0.5155
100	25	0.1569	0.0370	0.8030	0.0305	4.7048	0.4384
500	29	0.1568	0.0323	0.8101	0.0257	4.7554	0.5793
700	31	0.1548	0.0336	0.8073	0.0251	4.7548	0.4083
900	31	0.1545	0.0329	0.8068	0.0256	4.7298	0.3985
300	29	0.1544	0.0364	0.8035	0.0287	4.8054	0.6285
500	33	0.1540	0.0312	0.8058	0.0224	4.6798	0.4379
900	33	0.1536	0.0313	0.8050	0.0237	4.8317	0.4414
100	23	0.1536	0.0313	0.8018	0.0266	4.6536	0.7398
700	33	0.1531	0.0318	0.8050	0.0232	4.7554	0.4300
300	33	0.1526	0.0332	0.8023	0.0223	4.5773	0.4148
500	31	0.1523	0.0304	0.8068	0.0248	4.7036	0.4808
100	33	0.1512	0.0293	0.8025	0.0176	4.5273	0.4767
700	35	0.1511	0.0306	0.8015	0.0223	4.6542	0.4885
300	31	0.1505	0.0310	0.8038	0.0285	4.7298	0.3492
900	35	0.1505	0.0294	0.8022	0.0227	4.7317	0.5493
100	31	0.1504	0.0332	0.7993	0.0273	4.7311	0.4005
500	35	0.1500	0.0285	0.8026	0.0224	4.6542	0.6041
300	35	0.1476	0.0301	0.7998	0.0242	4.7048	0.3877
100	35	0.1467	0.0303	0.7978	0.0203	4.7036	0.5700
100	29	0.1459	0.0347	0.7970	0.0333	4.7029	0.6122

Appendix B

Standard Deviations in Sequential Feature Addition Results

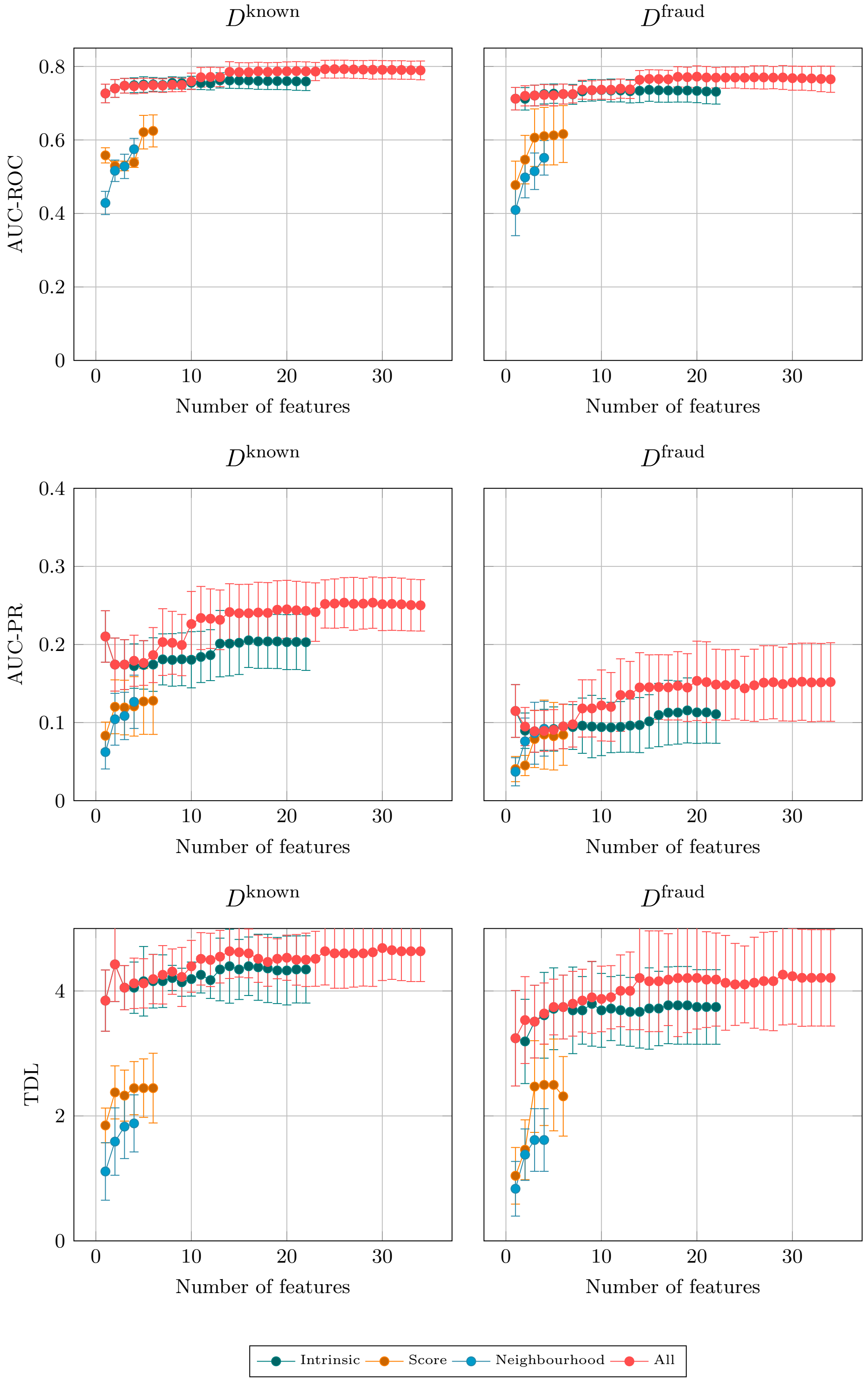


FIGURE B.1: Ten-fold cross-validated performance of LR models for sequential feature addition with error bars representing standard deviations

Appendix C

Model Summaries of Adapted Models

TABLE C.1: Summary of time-weighted LR model for D^{known} with intrinsic features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \lvert z \rvert$
`.const`	-2.4366	0.065	-37.725	0.000
`age`	-0.3502	0.027	-12.813	0.000
`amount`	0.3201	0.021	15.305	0.000
`amount1`	0.0718	0.029	2.504	0.012
`claimAge`	-0.7651	0.034	-22.475	0.000
`lastClaim`	-0.204	0.040	-5.138	0.000
`numContracts`	-0.1468	0.029	-5.065	0.000
`organisations`	-0.2129	0.025	-8.382	0.000
`people`	-0.154	0.030	-5.217	0.000
`refused1`	0.0982	0.028	3.543	0.000
`responsibilityCode_2`	-1.0704	0.478	-2.239	0.025
`responsibilityCode_3`	0.8545	0.125	6.842	0.000
`responsibilityCode_x`	0.2635	0.076	3.459	0.001

Table C.2: Summary of time-weighted LR model for D^{fraud} with intrinsic features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \lvert z \rvert$
`age`	-0.2922	0.026	-11.082	0.000
`amount`	0.1606	0.018	9.140	0.000
`amount1`	0.0907	0.022	4.176	0.000
`claimAge`	-0.6619	0.033	-20.076	0.000
`daysReport`	0.0704	0.017	4.122	0.000
`lastClaim`	-0.3049	0.042	-7.187	0.000
`lastClaim_x`	0.0997	0.029	3.454	0.001
`nClaims1`	0.0952	0.04	2.365	0.018
`nClaims5`	-0.1059	0.046	-2.306	0.021
`numContracts`	-0.2686	0.032	-8.375	0.000
`organisations`	-0.0874	0.024	-3.671	0.000
`responsibilityCode_3`	0.9359	0.119	7.875	0.000
`responsibilityCode_x`	0.2918	0.072	4.040	0.000
`sameSits1`	-0.1527	0.030	-5.026	0.000
`sameSits5`	0.1637	0.031	5.360	0.000

Table C.3: Summary of time-weighted LR model for D^{known} with score features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \lvert z \rvert$
`.const`	-1,8061	0,023	-77,708	0,000
`n1.max`	-0,5011	0,048	-10,486	0,000
`n1.q1`	0,3963	0,116	3,429	0,001
`scores0`	0,1923	0,036	5,298	0,000

Table C.4: Summary of time-weighted LR model for D^{fraud} with score features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \lvert z \rvert$
`.const`	-1.830	0.024	-77.764	0.000
`n1.max`	-0.5949	0.049	-12.080	0.000
`n2.max`	0.1735	0.037	4.705	0.000
`scores0`	0.2301	0.030	7.585	0.000

Table C.5: Summary of time-weighted LR model for D^{known} with neighbourhood features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \lvert z \rvert$
`.const`	-1.9095	0.028	-68.77	0.000
`n1.size`	-0.1420	0.024	-5.912	0.000
`n2.binFraud`	-0.0854	0.024	-3.525	0.000
`n2.ratioFraud`	0.3677	0.039	9.382	0.000
`n2.ratioNonFraud`	0.0830	0.029	2.902	0.004
`n2.size`	-0.6197	0.055	-11.194	0.000

Table C.6: Summary of time-weighted LR model for D^{fraud} with neighbourhood features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`.const`	-2.9405	0.191	-15.370	0.000
`n2.ratioFraud`	0.3368	0.036	9.440	0.000
`n2.size`	-3.6121	0.529	-6.828	0.000

Table C.7: Summary of time-weighted LR model for D^{known} with all features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`Intercept`	-2.8442	0.071	-39.826	0.000
`age`	-0.3698	0.028	-13.186	0.000
`amount`	0.2652	0.020	13.008	0.000
`claimAge`	-0.8025	0.035	-22.729	0.000
`lastClaim`	-0.1902	0.040	-4.707	0.000
`n1.size`	-0.3877	0.153	-2.529	0.011
`n2.binFraud`	-0.3209	0.051	-6.307	0.000
`n2.max`	0.3155	0.067	4.687	0.000
`n2.q1`	-0.4605	0.153	-3.013	0.003
`n2.ratioFraud`	0.3638	0.050	7.222	0.000
`n2.size`	-0.8181	0.077	-10.584	0.000
`numContracts`	-0.1253	0.029	-4.252	0.000
`organisations`	0.2703	0.125	2.154	0.031
`refused1`	0.0824	0.030	2.785	0.005
`responsibilityCode_1`	0.1609	0.082	1.962	0.050
`responsibilityCode_2`	-0.9858	0.494	-1.994	0.046
`responsibilityCode_3`	0.9897	0.126	7.843	0.000
`responsibilityCode_x`	0.6252	0.080	7.844	0.000

Table C.8: Summary of time-weighted LR model for D^{fraud} with all features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`.const`	-4.4838	0.667	-6.724	0.000
`age`	-0.3320	0.027	-12.080	0.000
`amount`	0.0780	0.018	4.235	0.000
`amount1`	0.0708	0.023	3.143	0.002
`claimAge`	-0.6737	0.034	-19.789	0.000
`lastClaim`	-0.2978	0.043	-6.890	0.000
`lastClaim_x`	0.1007	0.030	3.349	0.001
`n1.max`	0.1460	0.073	2.012	0.044
`n1.q1`	0.2576	0.119	2.166	0.030
`n2.binFraud`	-0.1681	0.049	-3.425	0.001
`n2.max`	0.2422	0.064	3.813	0.000
`n2.ratioFraud`	0.1239	0.029	4.267	0.000
`n2.size`	-5.2108	0.719	-7.243	0.000
`nClaims1`	0.0832	0.042	1.982	0.047
`nClaims5`	-0.1411	0.048	-2.946	0.003
`numContracts`	-0.1886	0.032	-5.885	0.000
`responsibilityCode_1`	0.2898	0.078	3.740	0.000
`responsibilityCode_3`	1.0379	0.122	8.517	0.000
`responsibilityCode_x`	0.7761	0.076	10.212	0.000
`sameSits1`	-0.1711	0.032	-5.361	0.000
`sameSits5`	0.1889	0.031	6.188	0.000

Table C.9: Summary of shared resources LR model for D^{known} with intrinsic features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \|z\|$
`.const`	-2.4366	0.065	-37.725	0.000
`age`	-0.3502	0.027	-12.813	0.000
`amount`	0.3201	0.021	15.305	0.000
`amount1`	0.0718	0.029	2.504	0.012
`claimAge`	-0.7651	0.034	-22.475	0.000
`lastClaim`	-0.2040	0.040	-5.138	0.000
`numContracts`	-0.1468	0.029	-5.065	0.000
`organisations`	-0.2129	0.025	-8.382	0.000
`people`	-0.1540	0.030	-5.217	0.000
`refused1`	0.0982	0.028	3.543	0.000
`responsibilityCode_2`	-1.0704	0.478	-2.239	0.025
`responsibilityCode_3`	0.8545	0.125	6.842	0.000
`responsibilityCode_x`	0.2635	0.076	3.459	0.001

Table C.10: Summary of shared resources LR model for D^{fraud} with intrinsic features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \lvert z \rvert$
age	-0.2922	0.026	-11.082	0.000
amount	0.1606	0.018	9.140	0.000
amount1	0.0907	0.022	4.176	0.000
claimAge	-0.6619	0.033	-20.076	0.000
daysReport	0.0704	0.017	4.122	0.000
lastClaim	-0.3049	0.042	-7.187	0.000
lastClaim_x	0.0997	0.029	3.454	0.001
nClaims1	0.0952	0.040	2.365	0.018
nClaims5	-0.1059	0.046	-2.306	0.021
numContracts	-0.2686	0.032	-8.375	0.000
organisations	-0.0874	0.024	-3.671	0.000
responsibilityCode_3	0.9359	0.119	7.875	0.000
responsibilityCode_x	0.2918	0.072	4.040	0.000
sameSits1	-0.1527	0.030	-5.026	0.000
sameSits5	0.1637	0.031	5.360	0.000

Table C.11: Summary of shared resources LR model for D^{known} with score features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \lvert z \rvert$
.const	-1.8212	0.023	-77.643	0.000
n1.max	-0.4476	0.046	-9.641	0.000
n1.med	-0.2401	0.120	-1.996	0.046
n1.q1	0.4982	0.121	4.116	0.000
scores0	0.1892	0.030	6.293	0.000

Table C.12: Summary of shared resources LR model for D^{fraud} with score features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \lvert z \rvert$
.const	-1.8672	0.024	-77.334	0.000
n1.max	-0.5730	0.050	-11.466	0.000
n2.med	-0.8076	0.175	-4.620	0.000
n2.q1	1.0339	0.194	5.334	0.000
scores0	0.2851	0.025	11.376	0.000

Table C.13: Summary of shared resources LR model for D^{known} with neighbourhood features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \|z\|$
.const	-1.9095	0.028	-68.77	0.000
n1.size	-0.1420	0.024	-5.912	0.000
n2.binFraud	-0.0854	0.024	-3.525	0.000
n2.ratioFraud	0.3677	0.039	9.382	0.000
n2.ratioNonFraud	0.0830	0.029	2.902	0.004
n2.size	-0.6197	0.055	-11.194	0.000

Table C.14: Summary of shared resources LR model for D^{fraud} with neighbourhood features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \|z\|$
.const	-2.9405	0.191	-15.370	0.000
n2.ratioFraud	0.3368	0.036	9.440	0.000
n2.size	-3.6121	0.529	-6.828	0.000

Table C.15: Summary of shared resources LR model for D^{known} with all features

	D^{known}			
Variable	Coef	Std. Err	z	$P > \|z\|$
.const	-2.8781	0.072	-39.854	0.000
age	-0.3845	0.028	-13.605	0.000
amount	0.2680	0.020	13.219	0.000
atFault5	-0.0790	0.040	-1.968	0.049
claimAge	-0.8011	0.035	-22.591	0.000
lastClaim	-0.1963	0.041	-4.817	0.000
n1.max	0.2710	0.084	3.236	0.001
n1.q1	0.2447	0.123	1.993	0.046
n1.size	-0.3159	0.156	-2.019	0.043
n2.binFraud	-0.1786	0.057	-3.134	0.002
n2.ratioFraud	0.3024	0.056	5.395	0.000
n2.size	-0.9936	0.084	-11.779	0.000
nClaims5	0.0953	0.045	2.133	0.033
numContracts	-0.1211	0.029	-4.108	0.000
refused1	0.0876	0.030	2.931	0.003
refused5	-0.0862	0.038	-2.264	0.024
responsibilityCode_1	0.1619	0.083	1.958	0.050
responsibilityCode_2	-1.0518	0.513	-2.052	0.040
responsibilityCode_3	1.0341	0.126	8.220	0.000
responsibilityCode_x	0.6396	0.080	7.975	0.000

Table C.16: Summary of shared resources LR model for D^{fraud} with all features

	D^{fraud}			
Variable	Coef	Std. Err	z	$P > \|z\|$
age	-0.3202	0.028	-11.586	0.000
amount	0.0795	0.018	4.363	0.000
amount1	0.0600	0.023	2.644	0.008
claimAge	-0.6654	0.034	-19.468	0.000
lastClaim	-0.2781	0.043	-6.462	0.000
lastClaim_x	0.0995	0.030	3.288	0.001
n1.max	0.2577	0.079	3.242	0.001
n1.q1	0.2359	0.110	2.143	0.032
n2.ratioFraud	0.0964	0.023	4.113	0.000
n2.size	-4.8314	0.638	-7.578	0.000
nClaims1	0.1005	0.042	2.403	0.016
nClaims5	-0.1149	0.048	-2.407	0.016
numContracts	-0.2045	0.033	-6.25	0.000
responsibilityCode_1	0.2300	0.078	2.951	0.003
responsibilityCode_3	1.0096	0.121	8.334	0.000
responsibilityCode_x	0.7029	0.076	9.208	0.000
sameSits1	-0.1594	0.032	-5.036	0.000
sameSits5	0.1752	0.031	5.704	0.000
scores0	0.0879	0.029	3.075	0.002

Appendix D

Declaration of Generative AI in the Writing Process

During the preparation of this work the author used OpenAI's ChatGPT based on GPT-3.5 for inspiration in paraphrasing in order to enhance readability. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.