Insurance Fraud Detection Using Supervised Machine Learning and Explainable Artificial Intelligence

University of Turku
Department of Computing
Master of Science (Tech) Thesis
Data Analytics
April 2026
Paavo Laine

Supervisors:
Jukka Heikkonen

UNIVERSITY OF TURKU
Department of Computing

Paavo Laine: Insurance Fraud Detection Using Supervised Machine Learning and Explainable Artificial Intelligence

Master of Science (Tech) Thesis, 65 p.
Data Analytics
April 2026

Insurance fraud is a significant problem that causes financial losses for insurance companies and higher prices for honest customers. Machine learning has been widely used in insurance fraud detection systems. However, fraudulent claims often represent only a small percentage of the data, and standard machine learning models struggle with this extreme data imbalance. Additionally, the most advanced models are often black boxes, meaning their predictions are not interpretable to outside observers. This inability to explain decisions is problematic given the highly regulated nature of the insurance industry.

This thesis aims to develop an insurance fraud detection system by utilizing machine learning models and tools from explainable artificial intelligence (XAI) research. A publicly available, labeled dataset of vehicle insurance fraud is used to train and evaluate the models. Based on the literature, four commonly used machine learning models are selected, trained, and evaluated. A logistic regression model and a decision tree are used as transparent baseline models, and they are compared with the more advanced black-box ensemble models: random forest and eXtreme Gradient Boosting (XGBoost). The class imbalance problem is addressed with cost-sensitive learning. To make the black-box models interpretable, this thesis uses Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP). The goal is not only to effectively detect insurance fraud, but also to provide explanations for why specific claims are suspicious.

The trained models were evaluated using various metrics that account for the imbalanced nature of the data. XGBoost was found to be the highest-performing model, while random forest also outperformed logistic regression and decision tree. Both SHAP and LIME were successfully applied to the XGBoost model to generate explanations for the predictions. SHAP was found to be a more robust and reliable method compared to LIME.

Keywords: insurance fraud, machine learning, cost-sensitive learning, explainable artificial intelligence, XAI, random forest, XGBoost, LIME, SHAP

Contents

List of Figures

List of Tables

List of acronyms

AUC-PR Area Under the Precision-Recall Curve

AUC-ROC Area Under the Receiver Operating Characteristic Curve

CRISP-DM Cross-Industry Standard Process for Data Mining

FDS Fraud Detection System

GDPR General Data Protection Regulation

LIME Local Interpretable Model-agnostic Explanations

SHAP SHapley Additive exPlanations

SMOTE Synthetic Minority Over-sampling Technique

TPE Tree-structured Parzen Estimator

XAI Explainable Artificial Intelligence

XGBoost eXtreme Gradient Boosting

1 Introduction

The insurance industry plays a crucial role in the modern economic system. It provides individuals and businesses with ways to manage risk, thereby increasing overall economic stability. In 2024, the global economic impact of the insurance industry reached $7 trillion in gross written premiums, representing approximately 7.4% of the global GDP [1]. However, the industry faces a persistent problem: fraudulent claims.

An insurance claim is a demand by a policyholder to an insurance provider for financial compensation for a loss covered by their insurance policy. Insurance companies handle millions of claims each year. Estimating the occurrence of fraud is notoriously difficult. However, it is estimated that combined detected and undetected insurance fraud accounts for 10% of all claim expenditures [2] [3]. This figure includes both hard fraud (premeditated fabrication of losses) and soft fraud (opportunistic exaggeration of legitimate claims).

Even when considering only the fraud that has been detected, insurance companies in the United Kingdom, for example, prevent £1 billion in fraudulent claims each year [4]. In 2022, insurance companies in Finland had 2500 unclear claims under investigation, with a total value of 147 million euros [5]. Fraud is a problem not only for insurance companies but also for their customers, as it increases prices for honest policyholders. The FBI has estimated that insurance fraud costs the average U.S. family between $200 and $300 each year [6].

These figures represent significant economic losses. However, they also present an opportunity to apply advanced machine learning algorithms to automate fraud detection and reduce financial losses. The insurance industry has widely adopted machine-learning-based solutions due to the vast amounts of data processed in the industry. However, academic research on using machine learning for fraud detection has been more limited, primarily due to limited access to insurance datasets.

1.1 Problem Statement

Insurance fraud forces insurers to raise premiums for honest policyholders, and manual claim review complicates the claims settlement process. Traditional methods, such as manual auditing and static rule-based systems, cannot scale to the growing volume of claims. The most effective fraud detection methods currently in use are based on machine learning.

However, implementing machine learning based solutions comes with its own challenges. First, fraud is a rare event compared to legitimate claims. Without suitable modifications, machine learning algorithms tend to be biased toward the majority class. This means that they fail to detect actual fraud, which is often a small minority in the dataset. This is a problem because the costs of false positives and false negatives are not equal. One incorrectly flagged legitimate customer may only require some manual labor from a claims handler and cause a minor delay. In contrast, one undetected case of fraud can cost an insurance company thousands of euros.

Secondly, most advanced machine learning models are black boxes, meaning that their internal decision-making processes are not interpretable by humans. The lack of interpretability of these models limits their practical deployment, especially in highly regulated industries like insurance. For instance, the European Union's General Data Protection Regulation (GDPR) [7] and the Artificial Intelligence Act [8] limit the

use of black-box artificial intelligence models in high-risk domains such as insurance. While numerous studies focus on maximizing the performance of machine learning models, research addressing interpretability and the use of Explainable Artificial Intelligence (XAI) for fraud detection is more scarce.

1.2 Research Methods and Sources

This study first conducts a literature review of fraud detection methods used in the insurance industry. This is done to gain a clear picture of the current state of the field and to contextualize the research within the broader academic literature.

In the empirical part of the thesis, the performance of various tree-based machine learning algorithms is compared in a fraud classification task. The included models are logistic regression, decision tree, random forest, and eXtreme Gradient Boosting (XGBoost). Logistic regression and decision tree models serve as baselines and are compared to the more advanced random forest and XGBoost. To address the dataset imbalance, the study utilizes class-sensitive learning.

For training and evaluation, a publicly available dataset on vehicle insurance claim fraud [9] is used. The dataset contains 33 features and consists of 15 420 claims, of which 6% have been labeled as fraudulent. The models are evaluated based on various performance metrics, including recall, precision, F1-score, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), and the Area Under the Precision-Recall Curve (AUC-PR).

Logistic regression and decision trees are intrinsically interpretable, but random forests and XGBoost are not. The study incorporates XAI techniques, mainly SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME), to make these advanced models more interpretable.

1.3 Purpose, Scope and Research Questions

The primary purpose of this research is to develop an interpretable machine learning framework that is capable of effectively detecting fraudulent insurance claims. The scope is limited to tabular data consisting of numerical and categorical features related to insurance claims. No unstructured data sources, like claim descriptions or images, are incorporated. Used methods are limited to supervised machine learning, mainly tree-based models. Unsupervised and deep-learning-based methods are excluded. Tree-based models better align with the interpretability goal of the thesis, since they do not require data to be standardized.

Models are evaluated based on their classification performance. Evaluation of real-time speed and costs in a production environment is outside the scope of this research. Interpretation of the model results with SHAP and LIME is limited to feature importance evaluation. Evaluation of the actual usefulness of these explanations with real claims handlers falls outside the scope of this research.

The following three research questions can be formulated based on the purpose and scope of the research:

RQ1: How do specific machine learning models (logistic regression, decision tree, random forest, and XGBoost) compare in their predictive performance when detecting vehicle insurance fraud?

RQ2: How can the imbalanced nature of insurance fraud data be addressed when using machine learning based methods?

RQ3: Can explainable artificial intelligence methods be used to interpret the predictions of fraud detection machine learning models?

1.4 Thesis Structure

This thesis consists of six chapters. The chapters after the introduction are organized as follows:

Chapter 2 presents the scientific research on insurance fraud detection. The chapter briefly discusses the concept of insurance fraud and its different typologies. Then it explores the literature on machine learning based fraud detection.

Chapter 3 presents the technical details of the machine learning concepts and methods that will later be used in the empirical part of the thesis.

Chapter 4 introduces the research process and the vehicle insurance dataset. Exploratory data analysis is performed, and the data is prepared for training the machine learning model. This chapter also explains the training process and the application of SHAP and LIME to the trained models.

Chapter 5 evaluates and compares the performance of the models. Model interpretability with SHAP and LIME is also evaluated. The chapter concludes with a discussion section.

Chapter 6 concludes the thesis. First, the findings are summarized, followed by answers to the research questions. The limitations are then addressed, and finally, future research directions are proposed.

2 Literature Review

This chapter reviews the literature on insurance fraud detection. First, insurance fraud is discussed at a general level, and various typologies of insurance fraud are presented. After that, the qualities of typical fraud detection systems used in insurance companies are discussed. This is followed by a review of machine-learning-based fraud detection methods. The machine learning methods found in the literature are classified as either supervised or unsupervised. Solutions to the class imbalance problem and XAI methods found in the literature are also discussed. Technical details of the machine learning methods used in this thesis are explained later in Chapter 3.

2.1 Insurance Fraud

Insurance fraud is a prevalent problem, but estimates of its extent vary. While many sources claim that 10% of claim expenditure is lost to fraud [2] [3], this figure has also been questioned [10] [11]. In a study, where experts manually reviewed 42000 auto injury claims, it was estimated that hard fraud and buildup added 13 to 17 percent excess to claim payments, most of which was caused by buildup [12]. Buildup refers to inflating the damages of a claim, and it very rarely leads to criminal consequences. According to the Association of British Insurers (ABI), insurance companies in the United Kingdom prevent over £1 billion in fraudulent claims each year [4]. So even when considering only the actual verified fraud, the financial impact is massive.

Insurance fraud can be classified based on multiple typologies. One of the most common classifications is to categorize insurance fraud as either soft or hard fraud [13] [11]. Soft fraud, also called buildup, refers to situations in which policyholders take advantage of an opportunity to inflate damages in an otherwise legitimate claim. The more serious hard fraud is premeditated and involves staging or inventing an incident.

Hard fraud can also include organized fraud rings, which are complex networks designed to systematically gain financial benefit through fraudulent activities. In the case of automobile fraud, these networks can include drivers, mechanics, insurance workers, healthcare professionals, lawyers, and police officers. [14]

It has been argued that soft fraud is a far greater financial problem than hard fraud because soft fraud is more socially acceptable [10]. However, soft fraud is also harder to detect because the claim so closely resembles a legitimate claim [11]. In automobile claims, in particular, exaggerating the damages and committing soft fraud may seem like a small, harmless lie with little risk of facing consequences.

Fraud can occur during underwriting or when a claim is made [13]. Underwriting fraud happens when the insurance contract is created or renewed. Underwriting fraud can include, for example, hiding information or existing insurance contracts. However, insurance fraud is most often associated with false, inflated, or fictitious claims [13]. Fraud can also be classified as either internal or external [15]. Internal fraud is committed by an employee against their own company, while external fraud is committed by outside actors, such as customers or vendors [16].

2.2 Fraud Detection Systems

Insurance companies use Fraud Detection Systems (FDS) to detect insurance fraud [16]. These systems encompass the entire claim-handling pipeline, from the moment a claim is filed through various automated insurance detection methods, and finally

to manual review by human investigators.

Fraud detection is part of the broader research field of anomaly detection [17] [18]. Fraud is an anomaly in the data. Related fields of anomaly detection include intrusion detection, medical anomaly detection, and industrial damage detection, among others [17]. In addition to insurance fraud, other subfields of fraud detection include credit card fraud, mass marketing fraud, and corporate fraud, among others [19].

Historically, the insurance industry has relied on manual labor and rule-based systems as the primary methods for detecting fraud. These methods were the industry standard for decades. However, they depend on the assumption that fraud follows a predictable pattern that can be identified by human experts. The most basic method of detecting fraud is having experts manually review the claims. Insurance companies often maintain a list of fraud indicators that claim handlers train on and use to assess the suspicion of a claim [13].

The problem is that humans are susceptible to cognitive biases and fatigue. A meta-analysis of 36 experimental studies concludes that even experienced human auditors struggle with fraud cues [20]. Manually checking each claim is also slower and more costly compared to automated methods. However, using purely automated systems is impractical and is often prohibited by regulation. Claim processing systems typically work by first using automatic methods to flag suspicious claims, which are then later reviewed by humans [10].

Before more advanced methods, rule-based systems were the primary approach to automating fraud detection. Rule-based systems use static business rules in the form of "if-then" to identify specific claims that require further review [21]. For instance, a claim may get flagged if the dollar amount exceeds a certain threshold. These rules are often based on the experience of professionals [21]. However, the utility of rule-based systems is limited because detection is limited to patterns already known

by experts [22].

There are multiple factors that often hinder the performance of fraud detection systems. The main problems are the high volume of claims filed, the imbalance between legitimate claims and fraudulent ones, and the complexity and variety of fraud patterns [16]. Additionally, fraud patterns constantly change, a phenomenon known as concept drift [16]. In the fraud detection task, it is not necessarily important to perfectly separate the two classes, but rather to prioritize the most suspicious claims ahead of the others [21]. The human investigators with their limited resources can then focus on these most important cases [23].

An important factor to consider is also the cost of false negatives and false positives in fraud detection. In the case of a suspicious claim, insurance companies need to constantly balance the cost of paying the settlement versus the operational cost of auditing the claim [24]. Delaying an honest customer's payout also damages reputation, which must be considered in the cost of flagging a claim [24].

When suspicion of fraud is strong enough, the insurer can decide to reduce or deny compensation, and in the most serious cases, press charges. However, soft fraud is often settled due to the financial and reputational costs involved in a lawsuit. [13]

2.3 Machine Learning in Fraud Detection

Due to the complexity of insurance fraud schemes, especially those involving organized rings spanning multiple claims, manual review and rule-based systems are often not sufficient compared to more modern methods. This is why machine learning models are often incorporated into the fraud detection systems [16].

Machine learning has been widely used in fraud detection, but the scarcity of publicly available real data has limited the amount of academic research [23] [15] [19]. Insurance companies are reasonably hesitant to share their private data for academic research. Labeled datasets are particularly difficult to obtain due to the

effort required to detect fraud.

Machine learning models are typically deployed as a part of the overall fraud detection system. After the fraud detection machine learning model has been deployed, it can be updated in two ways. It can use online learning, where the model is updated as new data arrives, or the model can be retrained periodically, such as every month or year [23]. However, in insurance fraud detection, determining whether a claim is fraudulent can take months of investigation. Online learning might be more suitable for other types of fraud-detection systems with shorter delays, such as credit card fraud.

Different approaches to fraud detection using machine learning can be categorized as either supervised or unsupervised learning. Supervised learning refers to training a model with labeled data, while unsupervised learning refers to training with unlabeled data. The literature also presents different XAI methods and different approaches to the class imbalance issue. The following subchapters summarize the research in each of these areas.

2.3.1 Unsupervised Learning

Many researchers use unsupervised learning because reliable fraud labels are often hard to obtain. Concept drift can also, over time, decrease the performance of an online detection model if it has learned from historical labeled data and if it is not utilizing adaptive learning [23]. As noted in the comprehensive survey [15], unsupervised methods are important in detecting new fraud types that have not been flagged in the historical data.

Isolation forest [25] is one of the most commonly used methods for unsupervised fraud detection. For example, isolation forests have been shown to be effective in detecting workers' compensation insurance fraud in a practical insurance industry setting [26]. Isolation forest is often claimed to be the state-of-the-art method for

anomaly detection [27].

Various other unsupervised learning methods have also been applied to insurance fraud detection, like autoencoder and variational autoencoder [28], unsupervised spectral ranking [29], self-organizing feature maps [30], and many others [31]. Since this thesis focuses on supervised learning, these unsupervised methods are not further discussed.

Unsupervised methods often suffer from high false positive rates because unusual behavior is not always fraudulent [21]. Supervised methods often beat unsupervised methods, even when limited labeled data is available [32]. However, unsupervised and supervised methods are suitable for different tasks. Supervised methods detect fraud patterns that have already occurred, and unsupervised methods detect novel fraudulent behavior [27]. This means that the methods should be thought of as complementary rather than substitutes [33]. This justifies the need for supervised methods when labels are available.

2.3.2 Supervised Learning

In one of the first successful real-world applications of machine learning based fraud detection, Ghosh and Reilly [34] used neural networks to detect credit card fraud. Their system reduced false positives by 20% compared to the bank's old rule-based system.

In another early study [35], the performance of multiple machine learning algorithms was compared. It was found that simple models, such as logistic regression, achieved results nearly as good as state-of-the-art models of the time, such as early neural networks. Aslam et al. [36] also found that logistic regression achieved the best F1-score when comparing logistic regression, support vector machine, and naïve Bayes in auto insurance fraud detection.

The decision tree performed poorly in the study by Viene et al. [35]. However,

when Hassan et al. [37] applied decision trees, support vector machines, and artificial neural networks to a car insurance fraud detection, they found that decision trees performed slightly better than the other algorithms.

Viene et al. also discussed how ensemble tree methods could be more suitable for fraud detection compared to the simpler models they used in their study [35]. Ensemble tree models, like random forests and XGBoost, have since become the state-of-the-art methods for supervised fraud detection. While deep learning dominates in images and text, tree-based models still often dominate in tabular data [38] [39]. Multiple studies have demonstrated the performance of the XGBoost model specifically in fraud detection [40] [41] [27].

Nabrawi and Alanazi used random forests, logistic regression, and artificial neural networks to detect healthcare insurance fraud [42]. In their study, random forests acquired the best performance. Random forest also had the highest performance in a study that compared 10 different supervised learning algorithms for detecting vehicle insurance fraud [43]. Random forest also beat neural networks and support vector machines in a credit card fraud detection study [23].

Multiple models can be combined into an ensemble to achieve better performance. In one study, researchers achieved higher performance in healthcare fraud detection with an ensemble model consisting of Catboost, XGBoost, LightGBM, and a random forest than with any of the individual models [44].

While supervised algorithms like random forests and XGBoost are the standard for fraud detection, they face a problem because the training data inevitably contains undetected fraud that has been recorded as legitimate instances [45]. This is called label noise. In fact, it could be more accurate to think of this type of training data as labeled and unlabeled classification [46]. In this framework, the positive class consists of verified fraud, while the negative class consists of unlabeled data that includes both legitimate claims and undetected fraud. Label noise also demonstrates

another utility of making the models more interpretable. Explanations of false positive instances could reveal fraud patterns and help determine whether the false positives are actually undetected fraud.

2.3.3 Class Imbalance

Fraud data is typically imbalanced because there are far fewer fraudulent cases compared to honest ones [15]. Many techniques have been proposed in the literature that aim to address the problem of imbalanced data. Most of the approaches can be classified as either cost-sensitive learning or sampling techniques [47]. This section presents the literature on handling imbalanced classes, while the technical aspects of the class imbalance solutions used in this thesis are further explained in Section 3.1.

Cost-sensitive learning refers to techniques that penalize the model's loss function during training. By applying a higher cost to misclassifying a rare event, such as fraud, compared to a common event, the algorithm is forced to prioritize finding the minority class rather than simply maximizing overall accuracy. [48]

Sampling techniques, on the other hand, aim to address severe class imbalance by physically altering the training dataset's distribution before a model is trained. Sampling techniques can be divided into undersampling and oversampling. In undersampling, examples from the majority class are discarded using various strategies to balance the distribution. In oversampling, the minority class is expanded, for example, by duplicating the minority observations or by generating synthetic observations. [49]

The problem with undersampling is that it can discard potentially useful instances from the majority class, thereby hindering the performance of the classification model [50]. This is why oversampling methods are often preferred over undersampling methods, especially in insurance fraud research, where the class imbalance is severe. With severe imbalance, a large proportion of the majority class would have to be

discarded. While undersampling-based approaches are rare in the fraud detection literature, some studies have examined them. For example, Hassan et al. [37] compared different undersampling approaches in their research on car insurance fraud.

Synthetic Minority Over-sampling Technique (SMOTE) [51] is one of the most popular oversampling techniques. SMOTE generates new synthetic examples of the minority class. It works by selecting a minority observation, finding its k-nearest neighbors, and then using linear interpolation to create new data points along the vector connecting them in the feature space [51]. SMOTE and its variations have been widely used in insurance fraud research. For example, Nabrawi and Alanazi successfully utilized SMOTE in their research on health insurance fraud [42].

SMOTE-ENN is one of the SMOTE variations that have been successfully used in insurance fraud research [52]. SMOTE-ENN adds an additional step to the standard SMOTE algorithm. It uses Edited Nearest Neighbors (ENN) to remove overlapping class boundaries, thereby reducing the noise introduced by the standard SMOTE.

Khan et al. [53] argue that the Adaptive Synthetic (ADASYN) sampling approach is more effective than SMOTE when the class imbalance is severe. ADASYN focuses on the samples that are harder to classify and generates synthetic copies of them, unlike uniform sampling with SMOTE. It can, however, be argued that ADASYN is not safe for fraud datasets, since they often have a lot of noise, and ADASYN risks amplifying noise [54].

While SMOTE is a popular technique for addressing class imbalance, many studies suggest that cost-sensitive learning can be equally or more effective [55]. Weiss et al. [56] demonstrated that, especially with a dataset of more than 10,000 examples, the cost-sensitive learning algorithm often outperforms the sampling methods. It has also been shown, specifically for insurance fraud, that weighted XGBoost can acquire better performance than using oversampling techniques [41].

Standard SMOTE also does not work with categorical data, since it uses Euclidean distance, which requires continuous dimensions [51]. If SMOTE is applied to encoded nominal data, the algorithm synthesizes statistically impossible values for discrete categories [57]. A variation of SMOTE called SMOTE-NC [51] exists, but it requires a mixed dataset that has both continuous and nominal features [57]. Furthermore, SMOTE is not effective for high-dimensional data because it decreases the true variability and introduces artificial correlations between samples [58].

Furthermore, since SMOTE alters the data distribution, it has been argued that using SMOTE in combination with SHAP and LIME can decrease the faithfulness of the post-hoc explanations [59] [60]. This happens because generated synthetic examples may distort the actual patterns in the data and, in turn, affect the explanations generated by SHAP and LIME [59].

2.3.4 Explainable Artificial Intelligence (XAI)

Explainable AI refers to a collection of techniques to improve the interpretability of machine learning models [61]. The goal is to make the models easier for humans to understand, trust, and maintain [61]. Explainable AI can be separated into intrinsically interpretable models (such as logistic regression and decision trees) and post-hoc explainability tools (such as SHAP and LIME) [62].

Most advanced machine learning algorithms are black-box models. The internal logic of these models is not interpretable, meaning that outside observers cannot understand the rationale behind their decisions. Their decision-making logic consists of millions of abstract calculations. Post-hoc explanation tools aim to open the black box by assigning a numerical value to each feature, indicating how much each feature affects the model's prediction.

However, in insurance fraud detection, like in many other applications, it is important and useful to understand the principles on which a machine learning

model bases its predictions. Article 22 of the GDPR [7] from the European Union states that individuals have the right to understand how and why an automated system made a specific decision about them. GDPR gives individuals a "right to explanation", which means that they can ask for the specific reasons why an algorithm made a specific decision that affected them [63]. Decisions made by automated systems should not be based on potentially discriminatory features such as age, gender, or race [63].

Simply removing these features from the model is not sufficient, since features that correlate with the discriminatory features may still remain [64]. For example, a residential area can correlate with socioeconomic status or race, and occupation can correlate with gender.

The EU AI Act [8] presents further limitations to the use of machine learning in high-risk domains. Article 13 of the act requires high-risk systems to be sufficiently transparent so that human reviewers, such as insurance claims adjusters, can understand and monitor the system. Furthermore, article 14 of the act mandates that high-risk AI systems should always be overseen by humans.

There are also practical reasons to adopt explainable AI methods in the insurance industry. The original LIME study demonstrated that accuracy alone is often not enough for users to adopt a model, but they also need to understand why the prediction was made [65]. Further research has also shown that interpretability is a critical factor for experts to trust machine learning models [66].

Explainable AI has also been shown to benefit experts in practice in various fields. SHAP has been used in a system that predicts hypoxaemia during surgery to generate explanations of why the system was sounding an alarm [67]. The Bank of England has demonstrated the utility of SHAP values for predicting mortgage defaults [68].

Automated screening systems in insurance companies often work by giving

indications of what makes the claim suspicious [13]. This means that a single suspicion score is not enough, and explanations from the machine learning models are needed. Applying explainable AI methods to insurance fraud detection has been quite limited, though some studies have examined it.

For healthcare insurance fraud detection, SHAP and LIME were found to be effective in explaining the model predictions [44]. The researchers argue that the interpretability improves the model's practical application, since non-technical stakeholders can use it to focus on specific areas of concern and make more informed decisions. In another study, SHAP was used to identify a novel pattern of health insurance fraud previously unknown to experts [31].

In fraud detection, claims flagged by a machine learning algorithm still need to be manually reviewed by a human expert. Post-hoc explanations have been shown to be useful for fraud detection experts. One study found that using SHAP and LIME improved the accuracy with which experts detected fraud compared to using only the score from the machine learning model [69]. Decision time was also faster than when decisions were made using only raw data [69].

There are differences in the utility of these common post-hoc tools. LIME can be used to get local explanations, while SHAP can be used for both local and global explanations. In a practical study, users found LIME less helpful than SHAP because its explanations were less varied between different cases [69].

XAI methods come with their own challenges. Research has shown that by intentionally building a malicious machine learning system, the bias can be hidden from post-hoc explanation tools [70]. In the study, LIME was more susceptible to the intentional manipulation than SHAP [70].

It is also unclear whether the explanations are useful to claims adjusters. It has been shown that explanations can lead experts to over-trust the predictions made by machine learning models while dismissing their own intuition and expertise [71]. It is

a known challenge how to evaluate the predictions made by post-hoc explainers [59].

Due to the limitations of post-hoc tools, it has even been argued that their use should be stopped, and that the focus should instead be on fully transparent models [72]. Rudin argues that transparent models can achieve results similar to those of black-box models [72]. However, as was seen in Section 2.3.2, black-box models such as random forest and XGBoost seem to outperform simpler models, at least in the domain of fraud detection.

3 Technical Framework

This chapter introduces the machine learning concepts, methods, and algorithms that will be used later in the thesis. The first section discusses the problem of imbalanced data and how it is addressed in this thesis. Next, the hyperparameter tuning strategy and model evaluation metrics are presented. After this, the technical details of the machine learning algorithms used are presented. Logistic regression, decision tree, random forest, and XGBoost were selected as the machine learning models. Finally, the selected post-hoc explanation methods, LIME and SHAP, are introduced.

3.1 Cost-Sensitive Learning for Class Imbalance

Fraud is rare compared to legitimate claims, so datasets representing fraud tend to be imbalanced. This is also the case with the dataset used in this thesis. This is a problem because most machine learning algorithms expect an equal distribution of classes by default. When there is a major imbalance between the classes, the algorithms tend to bias towards the majority class to maximize accuracy. As a result, the minority class is not prioritized and is often treated as noise by the algorithm, even though it is the actual object of interest. This resulting model has high accuracy but poor precision in detecting the minority class, making it unlikely to be useful in a practical setting.

From the literature discussed in Section 2.3.3, it was determined that cost-sensitive learning and sampling methods are the most commonly used methods to address

class imbalance. As was discussed, cost-sensitive learning can be as effective as sampling methods [55] [56] [41]. It is also better for the post-hoc explanations if synthetic data is not created with oversampling methods [59] [60]. Standard SMOTE also does not work with categorical data [51]. Furthermore, the dataset used in this thesis is high-dimensional because one-hot encoding is used for categorical features, and SMOTE is less effective on high-dimensional datasets [58].

Undersampling methods are also not suitable in the context of this thesis. The dataset size is too small to delete enough rows to match the number of normal cases to fraud cases. Removing this many legitimate claims would negatively affect the performance of the models.

For these reasons, cost-sensitive learning is used in this thesis. This approach addresses the imbalance at the algorithmic level by modifying the loss function to give different weights to different classes. The weights apply a penalty to the minority-class (fraud) errors while reducing the penalty for the majority-class (non-fraud) errors. This forces the model to heavily penalize errors made on fraudulent instances. The formula for balancing class weights across the dataset is defined as:

$$w_j = \frac{n}{K \cdot n_j} \tag{3.1}$$

Where:

w_j: The penalty weight assigned to class j.

n: The number of samples in the training dataset.

K: The number of unique classes (in fraud binary classification $K = 2$).

n_j: The count of samples belonging to class j.

For the XGBoost model, the thesis uses cost-sensitive boosting [47]. The gradient of the positive class is multiplied by the weight at each boosting iteration. The gradients are used to build the next tree, so scaling them forces the algorithm to

focus more heavily on correcting mistakes it made in classifying the minority class in previous iterations.

3.2 Hyperparameter Tuning

The hyperparameter optimization was performed using Bayesian optimization, utilizing the Optuna framework [73]. An Optuna run consists of trials, each evaluating a specific set of hyperparameter values. Optuna chooses hyperparameters for each trial by utilizing the Tree-structured Parzen Estimator (TPE) algorithm. TPE models the search space by dividing the past hyperparameter combinations into two separate density distributions. The distribution $l(\theta)$ has the hyperparameters that achieve high performance, and $g(\theta)$ has the hyperparameters that achieve low performance. In each iteration, the algorithm proposes a new hyperparameter configuration θ which maximizes the ratio $\frac{l(\theta)}{g(\theta)}$. This way, the algorithm steers the search towards an optimal hyperparameter combination.

3.3 Evaluation Metrics

Because fraud data is imbalanced, accuracy is not a suitable performance metric [49]. For instance, a model could achieve high accuracy by incorrectly predicting that all instances are non-fraudulent. This is why different metrics are needed for imbalanced data. First, the following foundational elements are needed to construct suitable evaluation methods:

- **True Positives (TP):** Fraud cases correctly classified as fraud.
- **True Negatives (TN):** Legitimate cases correctly classified as legitimate.
- **False Positives (FP):** Legitimate cases incorrectly classified as fraud (Type I Error).

- **False Negatives (FN):** Fraud cases incorrectly classified as legitimate (Type II Error).

The performance is evaluated using precision, recall, F1-score, AUC-ROC, and AUC-PR. These metrics account for the data imbalance. The metrics are introduced below, and their definitions are based on the framework established by Powers [74].

Precision is the proportion of predicted positive cases that are actually positive. In the context of fraud detection, it measures how often the model is correct when it classifies a transaction as fraudulent. Low precision means that many legitimate customers are incorrectly flagged, leading to wasted labor and customer dissatisfaction.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

Recall is the proportion of positive cases that are correctly predicted as positive. In fraud detection, it measures the model's ability to find all the fraud cases in the dataset. A low recall would indicate that the model is missing fraudulent claims, resulting in financial losses for the insurer.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

F1-Score is the harmonic mean of precision and recall. Because increasing recall typically lowers precision and vice versa, the F1-score balances the two. The harmonic mean penalizes extreme values, so the model cannot achieve a high score by sacrificing one metric for the other.

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

AUC-ROC ROC curve plots the true positive rate (recall) against the false positive rate at every threshold (from 0% to 100%). AUC measures the area under the curve, condensing the graph into a single, easily comparable number. An AUC score of 0.5 means random guessing, while a score of 1.0 means perfect predictions.

$$AUC = \int_0^1 TPR(FPR)\, d(FPR) \tag{3.5}$$

where

$$TPR = \text{Recall} = \frac{TP}{TP + FN} \tag{3.6}$$

$$FPR = \frac{FP}{FP + TN} \tag{3.7}$$

AUC-PR However, with highly imbalanced datasets, AUC-ROC can present overly optimistic results due to the high number of true negatives, which artificially suppresses the false positive rate. Because of this, the PR curve might be better suited for the context of this thesis. It removes the true negatives from the equation entirely and only plots the trade-off between the model's precision and recall. [75]

3.4 Machine Learning Algorithms

As was seen in Chapter 2, tree-based models have achieved state-of-the-art results in previous research on fraud detection. That is why tree-based models were also chosen for this research. They align well with the interpretability goal of the thesis, since they do not require standardized data. Also, for tree-based models, SHAP values can be calculated with TreeSHAP, which is more computationally efficient [76]. In Chapter 2, logistic regression was shown to be a popular benchmark model in previous research [35] [36] [42], and that is why it was also chosen as the baseline

model for this research.

3.4.1 Logistic Regression

Logistic regression is one of the most widely used predictive models to identify relationships between categorical variables and one or more independent variables. It estimates the probability of an instance belonging to a particular group. The description of logistic regression presented here follows the principles detailed in [77].

Logistic regression computes a linear combination of the input features and then passes the score through a non-linear sigmoid activation function to transform the output into a probability distribution between 0 and 1. The probability of an observation belonging to the positive (fraud) class is defined as:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n)}} \tag{3.8}$$

Where:

β_0: The baseline log-odds of fraud.

$\beta_1, \beta_2 ... \beta_n$: The learned coefficients (weights) of the features.

$X_1, X_2 ... X_n$: The features of a claim.

During training, logistic regression optimizes the coefficients by minimizing an objective function known as binary cross-entropy or log-loss:

$$\mathcal{L}(\beta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \tag{3.9}$$

Where:

N: The number of claims.

y_i: The true class label (fraud or legitimate).

p_i: The model's predicted probability of fraud.

Because logistic regression calculates the log-odds coefficients for every feature, the contribution of each variable can be precisely explained. This makes the logistic regression an intrinsically interpretable model.

Logistic regression assumes a linear relationship between the features and the fraud. But fraud is rarely linear. For example, a fraudster can avoid suspicious high-value claims and instead target mid-tier claims. In this case, the relationship would be an upside-down U-shape. Logistic regression only draws straight lines, so it cannot capture this type of relationship.

3.4.2 Decision Tree

A decision tree is a classic supervised machine learning method. It breaks a complex problem into a series of "Yes or No" questions until it reaches a final decision. The definitions and properties presented here follow the principles detailed in [78].

The decision tree is built from the top down. It begins with the root node and the entire dataset. The algorithm evaluates all possible splits across all features and selects the one that best separates the classes of the target feature. Based on the split at the root node, the tree splits into two internal nodes, each containing smaller subsets of the data. These internal nodes iterate the process and find the split that best separates the classes. The same process iterates until the tree reaches the leaf nodes, which provide the final prediction.

Each potential split is evaluated with the Gini index. A Gini score of 0 would mean perfect purity, and a score of 0.5 would mean maximum impurity. The Gini index is calculated with the following formula:

$$Gini = 1 - \sum_{i=1}^{C} p_i^2 \tag{3.10}$$

Where:

C: The total number of classes.

p_i: The probability of an item belonging to class i.

A decision tree is inherently interpretable, meaning its internal logic is understandable to humans. These types of models are often called white-box models. Their logic does not rely on millions of abstract calculations as black-box models do, but rather logic that maps directly to human reasoning. A decision tree can be traced from the root to the leaves to see the exact reasoning behind the model's decision.

A common problem with decision trees is that they tend to overfit when they are not pruned. The tree grows until it has classified every instance in the training data. If the dataset contains noise, the tree builds highly specific branches to memorize the noise and will subsequently fail on unseen test data. Decision trees can also be highly unstable, meaning that a small change in the data can completely change the tree structure.

3.4.3 Random Forest

Random forest [79] is an ensemble learning method. Instead of relying on a single complex decision tree, a random forest averages predictions across hundreds of independent, shallow trees. Cost-sensitive learning can also be applied to random forest [55]. The technical description here is based on the original random forest paper [79]. To ensure that the trees are different from one another, a random forest relies on two core mechanics. They are bagging and random feature selection.

In bagging, every tree in the ensemble draws a random sample from the full training data with replacement. This ensures that each tree receives a slightly different version of the data, which helps prevent the ensemble from overfitting to outliers or noise.

Random feature selection means that at each node, the algorithm only has access to a random subset of features. By restricting the available features, the algorithm forces different trees to explore weaker patterns rather than always relying on the strongest pattern in the data.

The average prediction across all the trees in the ensemble is calculated with the following equation:

$$\hat{f}_{RF}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x) \tag{3.11}$$

Where:

$\hat{f}_{RF}(x)$: The final predicted probability of fraud for claim x.

B: The number of trees in the forest.

$\hat{f}_b(x)$: The probability predicted by the tree b.

Using an ensemble of trees loses the interpretability of a single decision tree. A random forest is a black-box model that requires post-hoc interpretability frameworks to generate explanations of the model predictions. Another problem with random forests is that the bagging technique is an independent process between different trees, which means that individual trees cannot learn from the mistakes of previous trees. This might limit the predictive power of the random forest.

3.4.4 XGBoost

XGBoost [80] is another ensemble machine learning method. Similar to random forests, XGBoost uses decision trees as the building blocks, and the trees are again very shallow trees compared to regular decision trees. The difference to random forest is that XGBoost builds trees sequentially. The technical details and equations described here are based on the original XGBoost paper by Chen and Guestrin [80].

XGBoost uses boosting as its training strategy. This means that the trees are

built sequentially, and each tree learns from the mistakes of the previous trees. This contrasts with the bagging strategy of random forest, where all trees are built in parallel and independently from each other. The final prediction for a given claim is the sum of the predictions from all K trees in the ensemble:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F} \tag{3.12}$$

Where:

$\hat{y}_i$: The predicted probability of fraud for claim i.

x_i: The feature vector of the claim.

f_k: One independent decision tree in the ensemble.

$\mathcal{F}$: The space of all possible trees.

The objective function of XGBoost consists of a loss function and regularization. The regularization term penalizes model complexity, which distinguishes XGBoost from traditional Gradient Boosting Machines (GBMs). At iteration t, the goal is to minimize the following objective:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{3.13}$$

The parameters are defined as:

l: The differentiable loss function.

y_i: The true label of the claim.

$\hat{y}_i^{(t-1)}$: The combined prediction from all the previous trees.

$f_t(x_i)$: The new tree that is currently built to improve the prediction.

$\Omega(f_t)$: The regularization penalty applied to the new tree.

The regularization term is defined as:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda||w||^2 \tag{3.14}$$

Where:

T: The total number of terminal leaves in the tree.

w: The score assigned to each leaf.

γ (gamma): Defines the minimum loss reduction required to make a further split on a leaf node. Acts as a complexity control.

λ (lambda): Smooths extreme attribution weights to prevent any single feature from dominating the model.

XGBoost uses a second-order Taylor expansion that involves both the first and second derivatives. This differentiates XGBoost from standard gradient boosting machines, which only optimize using the first derivative. This enables the XGBoost model to converge faster and produce more precise leaf weight estimates. The second-order Taylor approximation applied to the loss function at step t is:

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{3.15}$$

When training the XGBoost model in this thesis, cost-sensitive boosting [47] with the `scale_pos_weight` hyperparameter is applied to the loss function L before the Taylor expansion. This scales the gradient g_i and the Hessian h_i for all minority class (fraud) instances, which forces the algorithm to prioritize splits that isolate fraudulent claims.

While the sequential nature of XGBoost offers state-of-the-art performance, it is also susceptible to overfitting. While the first trees can capture the true fraud patterns, later trees might run out of real patterns to learn. The latter trees try to keep minimizing the objective function by building rules to catch random

anomalies in the specific training dataset. The weighting of fraudulent cases with `scale_pos_weight` hyperparameter further exacerbates overfitting to them. For this reason, it is important to optimize the γ and λ hyperparameters in the hyperparameter tuning phase. Tree growth is halted with γ when further splits would no longer significantly reduce error, while λ is used to shrink extreme leaf weights so the model does not overreact to outliers.

3.5 Explainable AI methods

While logistic regression and decision trees are intrinsically interpretable, the ensemble methods (random forest and XGBoost) are not. To interpret these black-box models, this thesis relies on two popular model-agnostic post-hoc explanation techniques: LIME and SHAP. Model-agnostic means that the techniques can be used with any machine learning model, and post-hoc means that the techniques are applied after the model has been trained. LIME can generate local explanations for individual instances, while SHAP can be used for both local explanations and global dataset-level explanations.

3.5.1 LIME

LIME [65] is a post-hoc, model-agnostic surrogate technique used to isolate and explain a single prediction (e.g., a specific insurance claim). LIME works by building a small interpretable model, typically a linear regression, that is used to predict the behavior of the more complex model. The technical description presented here is based on the original LIME research paper [65].

To generate explanations for a specific instance x, the LIME algorithm generates a synthetic dataset by perturbing the features of the sample x. The complex black-box model is then used to make predictions for these synthetic samples. The goal of LIME

is to find a surrogate model g that minimizes the loss L between the predictions of the complex model f and the simple model g. The objective function of LIME is as follows:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \ \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{3.16}$$

The parameters are defined as:

x: The instance that is being explained.

f: The black box model.

g: The simple and interpretable surrogate model chosen from a class of interpretable models G.

π_x: The proximity measure which defines how large the neighborhood around the instance is.

L: The loss function, which measures how close the explanation is to the prediction from the original model f.

$\Omega(g)$: The complexity penalty which forces the surrogate model to stay simple.

LIME has several limitations. Because the synthetic dataset is generated randomly, the resulting synthetic instances may contain impossible feature combinations. Because of its randomness, LIME is also inconsistent, so running it twice may result in different explanations. Additionally, LIME only explains the prediction of a single instance at a time. Therefore, it does not provide explanations of the feature importance at the global dataset level. However, due to its computational efficiency, LIME remains widely used across various domains, especially in those involving unstructured data and highly complex models.

3.5.2 SHAP

SHAP [81] is another commonly used post-hoc method for generating explanations from black-box models. The technical description presented here is based on the original SHAP paper by Lundberg et al. [81].

SHAP is based on cooperative game theory, specifically the Shapley values [82]. In the context of machine learning, the prediction task is formulated as a cooperative game in which the model's prediction serves as the payout, and the features are the players collaborating to achieve that specific prediction. The Shapley value aims to fairly distribute the prediction value among features based on their individual contributions.

SHAP aims to explain how much each feature in the data contributes to the final prediction. In practice, SHAP evaluates a feature's contribution by comparing the model's output across different feature combinations or coalitions. This process is repeated for all possible combinations of feature permutations, and in doing this, the exact predictive impact of every variable is calculated.

To calculate the Shapley value ϕ_i for feature i we can use the following formula:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right] \tag{3.17}$$

The parameters are defined as:

F: Set of all features in the model.

$|F|$: The number of features in the model.

S: A specific subset of features without the feature i.

$|S|$: The number of features in the current subset S.

$f_S(x_S)$: The models prediction with the subset S.

$f_{S \cup \{i\}}(x_{S \cup \{i\}})$: The models prediction when i is added to the subset S.

The term in the brackets $\left[f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)\right]$ calculates the change in the model's prediction when feature i gets added to the coalition of features S. Features also interact with each other, so the impact of a feature can change depending on the other features that are present. This means that the contribution of a feature needs to be calculated across all possible combinations of features. The term before the brackets, $\frac{|S|!(|F|-|S|-1)!}{|F|!}$, is the weighting mechanism where $|F|!$ is the number of possible feature permutations and $|S|!(|F| - |S| - 1)!$ is the number of possible feature permutations in S multiplied by the number of ways the remaining features can be added after feature i joins S. When these weighted contributions are summed across all subsets $S \subseteq F \setminus \{i\}$, it results in a single value ϕ_i, which represents the contribution of feature i to the fraud probability.

Compared to LIME, SHAP is mathematically guaranteed to be consistent. It can also be used to find the global importance of features by calculating the mean SHAP value of all claims in a dataset. This contrasts with LIME explanations, which cannot be reliably combined into global explanations. This is why SHAP will be used for both local and global explanations in this thesis, whereas LIME will be used only for local explanations.

The exact computation of Shapley values is NP-hard for general models [81]. However, the TreeExplainer method can be used for tree-based models. It leverages the internal structure of trees and computes the Shapley values in polynomial time [76]. Unlike the standard SHAP calculation, TreeSHAP does not iterate through all the coalitions. Instead, it explores the tree structure and only iterates the coalitions that would actually alter the predictions. All the models used in this research are tree-based, except for the logistic regression model, which is already interpretable without additional methods. For this reason, the TreeExplainer method can be used in this research to calculate the Shapley values.

4 Model Development

This chapter explains the overall methodological framework utilized to develop the fraud detection system. First, the research design and the dataset are introduced. Next, the data exploration and preparation phases are discussed. Finally, the model training process is detailed.

The machine learning pipeline development was conducted in the Google Colab environment using various Python libraries. The XGBoost was implemented with the official XGBoost Python library [80], and the other models were implemented with the scikit-learn library [83]. The hyperparameter optimization was done with the official Optuna framework [73].

4.1 Research Design

This research design follows the widely used Cross-Industry Standard Process for Data Mining (CRISP-DM) framework [84]. CRISP-DM consists of six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. These phases do not necessarily follow this order, and the process can also return to previous phases, as seen in Figure 4.1.

In the context of this research, business understanding refers to the literature review conducted in Chapter 2. Data understanding is covered in Section 4.2 and Section 4.3, followed by data preparation in Section 4.4, modeling in Section 4.5, and finally evaluation in Chapter 5. The final phase, model deployment, is not included

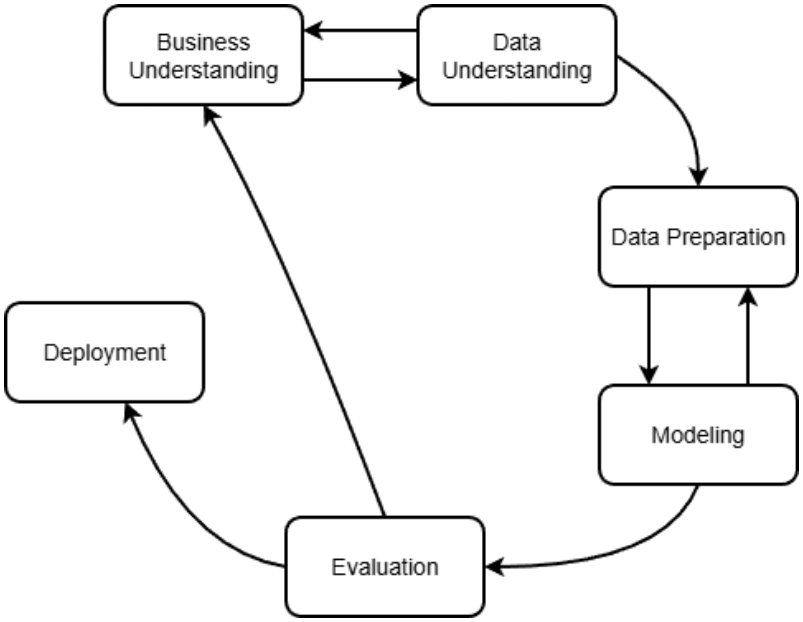

Figure 4.1: Phases of CRISP-DM. Adapted from [84].

in this research. The full machine learning pipeline development process is shown in Figure 4.2.

4.2 Dataset

This study uses a publicly available dataset acquired from Kaggle [9]. The dataset was originally published by Oracle, and it is from a US-based insurance company. The dataset does not contain any personally identifiable information, such as names, registration numbers, birth dates, or social security numbers.

The dataset consists of 33 features, which are listed in Table 4.1. The feature `FraudFound_P` defines if a specific instance is fraudulent or not. The data has 15 420 instances, of which 923 are fraudulent, and 14 497 are non-fraudulent. This means that in around 6% of the data instances fraud has been detected.

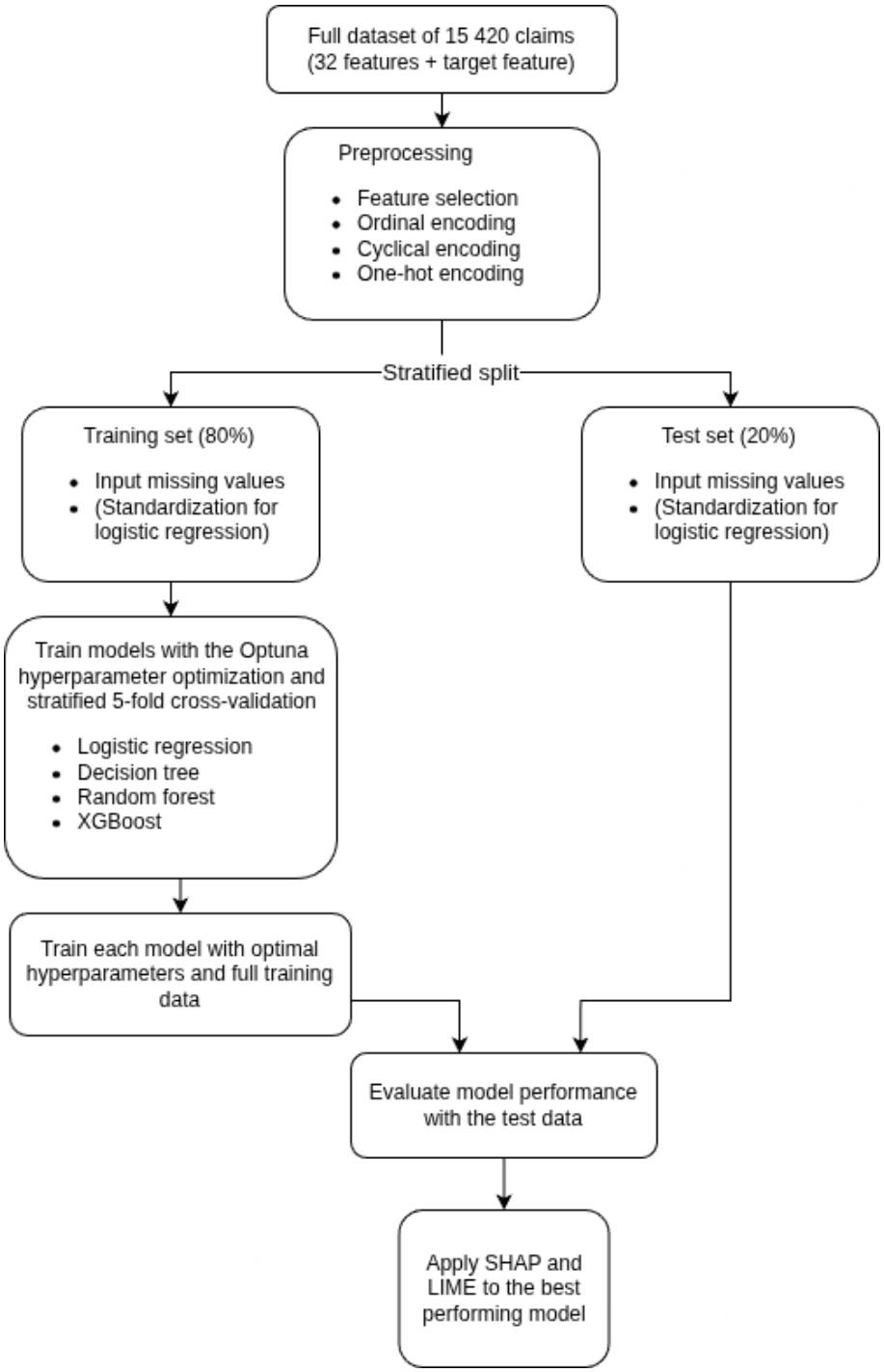

Figure 4.2: Full model training process.

Table 4.1: List of dataset features and their descriptions.

Feature	**Description**
FraudFound_P	Target variable (0 = Legitimate, 1 = Fraud)
Month	Month the accident occurred
WeekOfMonth	Week of the month the accident occurred
DayOfWeek	Day of the week the accident occurred
Make	Manufacturer of the vehicle
AccidentArea	Location of the accident (Urban, Rural)
MonthClaimed	Month the claim was filed
WeekOfMonthClaimed	Week of the month the claim was filed
DayOfWeekClaimed	Day of the week the claim was filed
Sex	Sex of the policyholder
MaritalStatus	Marital status of the policyholder
Age	Age of the driver in years
Fault	Who caused the accident (Policy Holder, Third Party)
PolicyType	Composite field (e.g., "Sport - Collision")
VehicleCategory	Classification of the vehicle
VehiclePrice	Estimated value range of the vehicle
PolicyNumber	Unique ID for each policy
RepNumber	ID of the claim handler
Deductible	Policyholder's out-of-pocket liability
DriverRating	Internal risk rating of the driver
Days_Policy_Accident	Days between policy purchase and the accident
Days_Policy_Claim	Days between policy purchase and the claim
PastNumberOfClaims	Number of claims filed prior to this incident
AgeOfVehicle	Age of the accident vehicle
AgeOfPolicyHolder	Categorical age range of the policyholder
PoliceReportFiled	Was a police report filed after the accident
WitnessPresent	Were there independent witnesses present
AgentType	Insurance handling agent type
NumberOfSuppliments	Supplemental items added to the claim
AddressChange_Claim	Policyholder's last address change
NumberOfCars	Number of vehicles registered to the policyholder
Year	Year the accident occurred (1994-1996)
BasePolicy	Type of insurance coverage

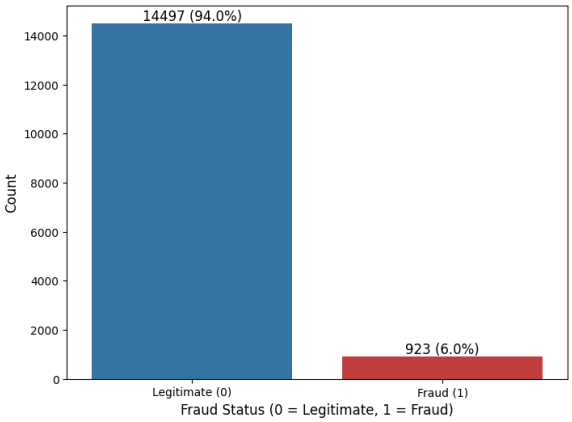

Figure 4.3: Class distribution of the target variable.

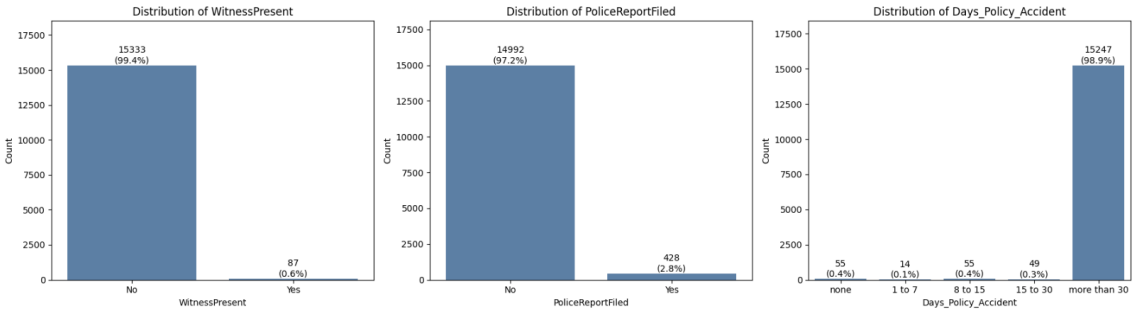

Figure 4.4: Distribution of three selected features (witness presence, police report, and time from policy purchase to accident).

4.3 Exploratory Data Analysis

Before preprocessing the data, an exploratory data analysis was conducted. This is part of the data understanding phase of the CRISP-DM framework. In this phase, feature values and distributions were explored, and this section highlights some key findings.

Figure 4.3 illustrates the distribution of the target variable `FraudFound_P`. As can be seen, there is a severe class imbalance, with fraudulent claims making up only 6% of the total set of claims. Some of the other features are also highly imbalanced. Figure 4.4 illustrates this with features `WitnessPresent`, `PoliceReportFiled` and `Days_Policy_Accident`.

Figure 4.5 shows the distribution of fraud by day of week and month. Fraud

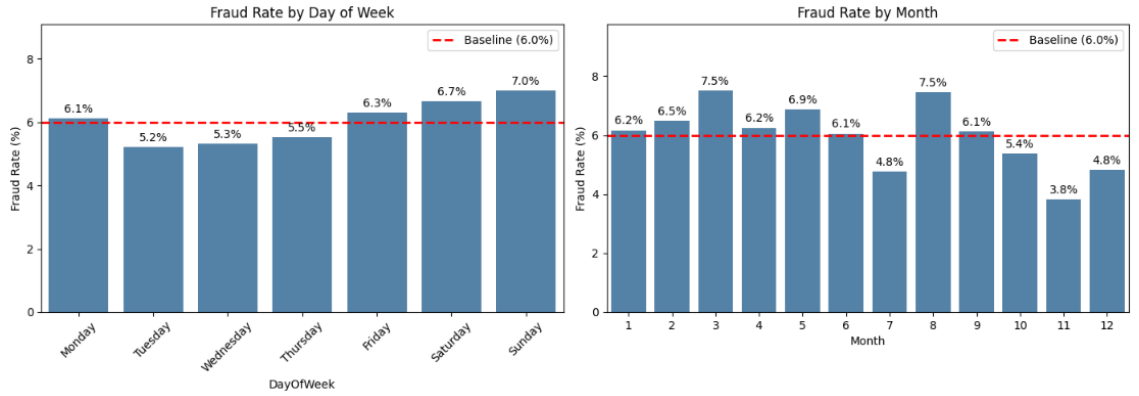

Figure 4.5: Distribution of fraud by day of the week and month of the accident.

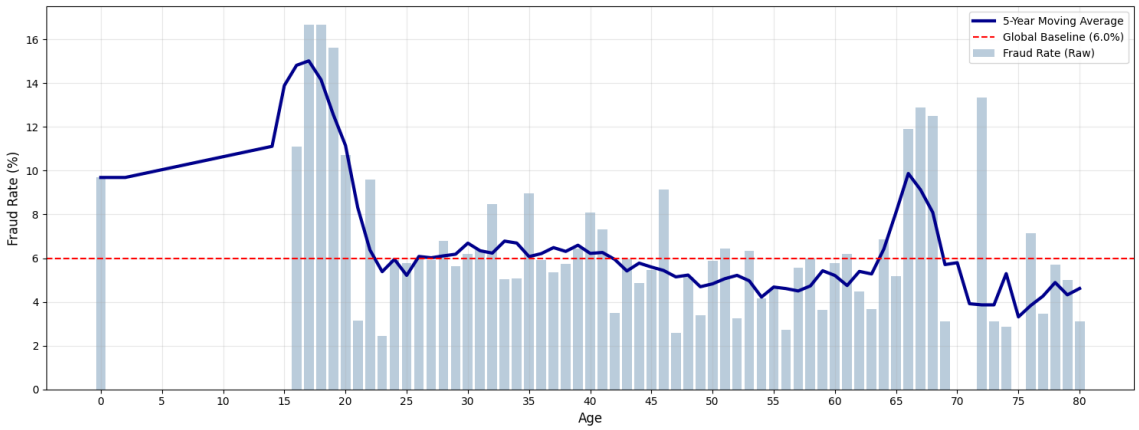

Figure 4.6: Fraud rate by age (with a 5-year moving average).

seems to somewhat increase around the weekend, with the highest rate of 7.0% on Sunday. Conversely, in the middle of the week, the fraud rate is somewhat reduced. The monthly distribution does not seem to show any clear pattern. Fraud rates are highest in March and August, with both having a rate of 7.5%, while November has the lowest rate of 3.8%.

Figure 4.6 shows the fraud rate by age. The fraud rate is mostly flat, but there are spikes among young and old drivers. The fraud rate is slightly higher among ages 25 to 40 than among ages 40 to 60. From the figure, we can also see that the feature includes some missing values represented by value 0.

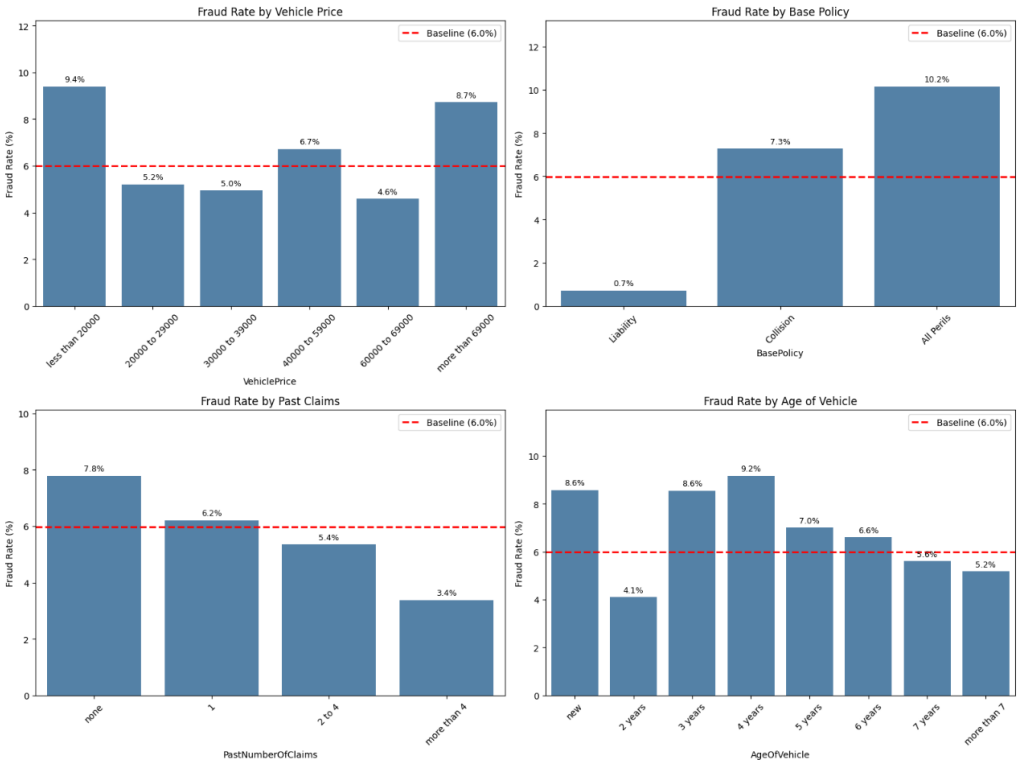

Figure 4.7: Fraud rates across four selected features.

Figure 4.7 shows the fraud rates for the features `VehiclePrice`, `BasePolicy`, `PastNumberOfClaims`, and `AgeOfVehicle`. For `VehiclePrice`, we can see that the fraud rate is highest for the least and most expensive vehicles. This may indicate very different types of fraud. For example, high-priced vehicle fraud may include hard fraud, while low-priced vehicle fraud may include more soft fraud. With `BasePolicy`, the fraud rate increases as the coverage of the policy increases. This seems logical because higher policies offer better payouts for accidents, which might encourage fraudulent behavior. For `PastNumberOfClaims`, the fraud rate decreases as the number of past claims grows. It could be speculated that long-time customers with more past claims are less likely to commit fraud. `AgeOfVehicle` is highest for four-year-old vehicles, after which the rate starts to decrease. However, there is also a spike in the fraud rate for brand-new vehicles, which may indicate some kind of organized fraud.

The apriori algorithm was also used as a part of data exploration to investigate potential rule-based patterns in the claims data. The output was filtered to show only the rules that predict fraud. The objective was to determine whether fraudulent behavior could be isolated using commonly occurring feature combinations. The majority class was undersampled before running the algorithm. The top found rules are presented in Table 4.2.

Table 4.2: Top association rules mined via apriori algorithm.

Antecedent (Itemset)	**Support**	**Conf.**	**Lift**
`Sex`: Male ∧ `BasePolicy`: All Perils ∧ `Fault`: Policy Holder ∧ `NumberOfSuppliments`: none ∧ `PoliceReportFiled`: No	0.113	0.783	1.566
`Sex`: Male ∧ `BasePolicy`: All Perils ∧ `WitnessPresent`: No ∧ `Fault`: Policy Holder ∧ `NumberOfSuppliments`: none	0.114	0.781	1.561
`NumberOfSuppliments`: none ∧ `Sex`: Male ∧ `BasePolicy`: All Perils ∧ `Fault`: Policy Holder	0.114	0.781	1.556

As shown in Table 4.2, the generated rules were very similar to each other. The rest of the rules also followed a similar pattern. The generated rules were mainly redundant variations of the same set of features, and the algorithm was ineffective at detecting different fraud typologies. Because the apriori algorithm constructs the rules based on the occurrence rates of feature combinations, the most common attributes of the dataset dominate the results. This inability to move beyond repetitive feature combinations justifies the need for more complex machine learning models that can also capture nonlinear relationships.

4.4 Data Preparation

The data preparation process included feature selection, converting categorical variables to numeric values, splitting the data into training and test sets, and addressing missing and incorrect values. The final part of this section describes the additional preprocessing steps performed on the data for the logistic regression model.

Feature selection: The `PolicyNumber` feature is a unique identifier and lacks predictive power, so it was removed. The `RepNumber` feature contains the identifier of the person who processed the claim. Since it is not relevant to the predictions, it was also excluded.

The `PolicyType` feature consisted of composite values (e.g., "Sport - Liability"), which combine the vehicle category and the coverage type. The second part of the value was always the same as the `BasePolicy` feature. The first part of the value had the same possible labels as the `VehicleCategory`, but the actual values of these two features were not identical across all claims. For these reasons, only the first part of `PolicyType` (e.g., "Sport") was retained, and the second part of the value was removed.

Categorical variables: The dataset contains multiple categorical variables. Manual ordinal encoding was used for the variables specified in the first part of Table 4.3. One-hot encoding was used for non-nominal categorical variables. These are listed in the second part of Table 4.3.

The time-related features like `Month` specified in the third part of Table 4.3 are inherently cyclical. In standard ordinal encoding, the distance between December (12) and January (1) would be 11, even though the months are actually adjacent. This is why cyclical encoding was used for these features. Cyclical encoding maps

Table 4.3: Features that were encoded during the preprocessing.

Feature name	**Original values (abbreviated)**
Categorical features *(ordinal encoding required)*	
`VehiclePrice`	<20k, 20k-29k, 30k-39k, 40k-59k, 60k-69k, >69k
`Days_Policy_Accident`	none, 1-7, 8-15, 15-30, more than 30
`Days_Policy_Claim`	none, 1-7, 8-15, 15-30, more than 30
`PastNumberOfClaims`	none, 1, 2-4, more than 4
`AgeOfVehicle`	new, 2, 3, 4, 5, 6, 7, more than 7
`AgeOfPolicyHolder`	16-17, 18-20, 21-25, ..., 41-50, 51-65, >65
`NumberOfSuppliments`	none, 1-2, 3-5, more than 5
`AddressChange_Claim`	<6 months, 1 year, 2-3 years, 4-8 years, no change
`NumberOfCars`	1, 2, 3-4, 5-8, more than 8
`BasePolicy`	Liability, Collision, All Perils
Categorical features *(one hot encoding required)*	
`Make`	Ford, Toyota, Honda, ...
`MaritalStatus`	Single, Married, Divorced, Widow
`VehicleCategory`	Sport, Sedan, Utility
`PolicyType`	e.g. "Sport - Liability"
Cyclical features *(cyclical encoding required)*	
`Month`	Jan, Feb, Mar, ...
`WeekOfMonth`	1, 2, 3, 4, 5
`DayOfWeek`	Monday, Tuesday, ...
`MonthClaimed`	Jan, Feb, Mar, ...
`WeekOfMonthClaimed`	1, 2, 3, 4, 5
`DayOfWeekClaimed`	Monday, Tuesday, ...

Table 4.4: Features that were not modified during preprocessing.

Feature name	**Original values (abbreviated)**
Binary features *(leave as is)*	
`FraudFound_P`	0 = Legitimate, 1 = Fraud
`AccidentArea`	Urban, Rural
`Sex`	Male, Female
`Fault`	Policy Holder, Third Party
`PoliceReportFiled`	Yes, No
`WitnessPresent`	Yes, No
`AgentType`	Internal, External
Ordinal features *(leave as is)*	
`Age`	16, 17, 18, ..., 80
`Deductible`	400, 500, 600, 700
`DriverRating`	1, 2, 3, 4
`Year`	1994, 1995, 1996

the variables onto a unit circle using sine and cosine functions. This preserves the proximity between the end of one cycle (December) and the start of the next (January).

The features listed in Table 4.4 are not modified in the preprocessing for the tree-based models. Special preprocessing is applied to the data for logistic regression, and that is explained at the end of this section.

Train-test split: Next, the dataset was partitioned into training (80%) and testing (20%) subsets using stratified sampling to preserve the class distribution (6% fraud) in both subsets. All other preprocessing steps were performed separately for the training and test sets to prevent information leakage between the sets.

Missing and incorrect values: The `Age` feature contained 320 missing values, which were represented by the value 0. Missing values were replaced with the

median value of the training set. The `MonthClaimed` feature contains a single missing value, which was replaced with the mode of the variable in the training set. The `DayOfWeekClaimed` feature also contains a single missing value, which was imputed with the mode of the training set. The `Fault` feature contains a single incorrect value: "Policy Ho". This value was replaced with the expected correct value: "Policy Holder".

Preprocessing for logistic regression: Data for the logistic regression required additional preprocessing compared to the tree-based models. First, the binary variables were encoded numerically as 0s and 1s. Standardization was fitted to all except binary variables in the training set before training the model. The parameters from the training set were also used to standardize the test set. This was done because logistic regression requires standardized data, unlike the tree-based models.

Also, the one-hot encoding for features `Make`, `MaritalStatus`, `PolicyType`, and `VehicleCategory` required dropping one of the encoded categories. This was done to avoid perfect multicollinearity, in which redundant features perfectly predict each other, preventing logistic regression from calculating independent feature weights.

4.5 Model Training

Using the prepared data, hyperparameter tuning was conducted on the models with the Optuna framework [73]. Hyperparameter search spaces for logistic regression, decision tree, random forest, and XGBoost are shown in Table 4.5. Optuna was run for 100 trials per model. Five-fold cross-validation with stratification was used for each model in each trial. The AUC-PR was used as the scoring function during the Optuna hyperparameter selection. After identifying the best settings, the models were retrained using the entire training set. The models were then evaluated using the unseen test set.

Table 4.5: Hyperparameter search space for each model.

Hyperparameter	Value Range	Scale
Logistic regression		
c_value	[1e-4, 1000]	Float (logarithmic)
penalty	{'l1', 'l2'}	Categorical
class_weight	'balanced'	Fixed float
Decision tree		
max_depth	[3, 15]	Integer (linear)
min_samples_split	[5, 50]	Integer (linear)
min_samples_leaf	[2, 20]	Integer (linear)
class_weight	'balanced'	Fixed float
Random forest		
n_estimators	[50, 500]	Integer (linear)
max_depth	[3, 15]	Integer (linear)
min_samples_split	[5, 50]	Integer (linear)
min_samples_leaf	[2, 20]	Integer (linear)
max_features	{'sqrt', 'log2'}	Categorical
class_weight	'balanced_subsample'	Fixed float
XGBoost		
n_estimators	[50, 500]	Integer (linear)
max_depth	[3, 10]	Integer (linear)
learning_rate	[0.01, 0.3]	Float (logarithmic)
subsample	[0.6, 1.0]	Float (linear)
colsample_bytree	[0.5, 1.0]	Float (linear)
gamma	[0, 5]	Float (linear)
reg_lambda	[1e-3, 10]	Float (logarithmic)
scale_pos_weight	Ratio of non-fraud / fraud	Fixed float

Based on the performance results further discussed in Section 5.1, XGBoost was selected as the best model. Before generating the LIME and SHAP visualizations, the encoded features were mapped back to their original representations to make the results from LIME and SHAP easier to interpret. For example, the sin and cos values from cyclical encoding done to the `Month` feature are not easily interpretable, so the values were mapped back to their original labels. After this, LIME and SHAP visualizations were generated, which are presented and discussed in the next chapter.

5 Results and Discussion

This chapter presents the results of the machine learning model training. Performance is evaluated based on precision, recall, F1-score, AUC-ROC, and AUC-PR. Confusion matrices and ROC curves are also compared. After the performance evaluation, the model interpretability with LIME and SHAP is evaluated. The chapter ends with a discussion of the results.

5.1 Model Performance

All of the performance metrics have been collected in Table 5.1. The final performance metrics were reported based on the model's performance on the unseen test set. The metrics are reported for the minority class (fraud). This is because, due to class imbalance, metrics for the majority class (legitimate) are very high and do not reflect the model's ability to detect fraud, which is the ultimate goal of the models.

Table 5.1: Performance metrics for each model.

Model	**Precision**	**Recall**	**F1-score**	**AUC-ROC**	**AUC-PR**
Log regression	0.15	**0.75**	0.24	0.79	0.15
Decision tree	0.16	0.55	0.24	0.77	0.20
Random forest	**0.25**	0.40	0.31	0.83	0.22
XGBoost	0.24	0.52	**0.33**	**0.86**	**0.27**

As shown in Table 5.1, XGBoost was the best-performing model based on F1-score, AUC-ROC, and AUC-PR. Random forest achieved the second-best performance,

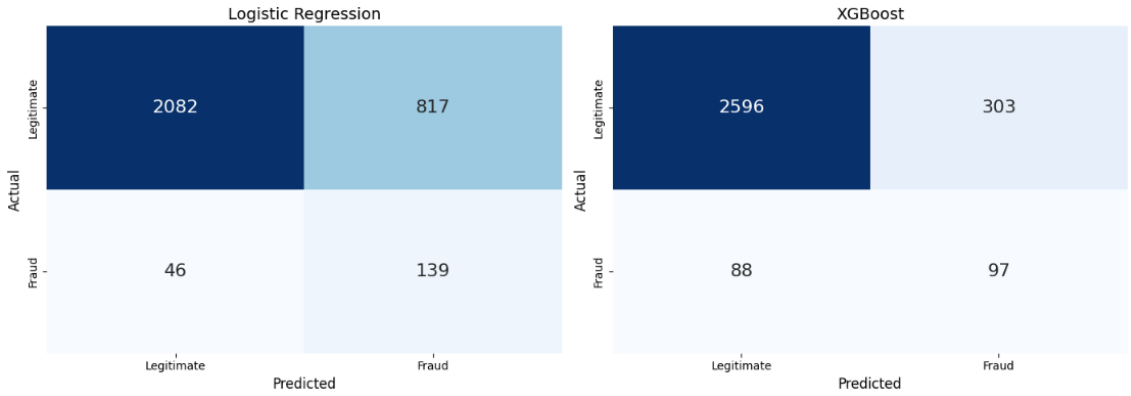

Figure 5.1: Confusion matrices for logistic regression and XGBoost models.

while the decision tree and logistic regression models had the lowest performance, achieving the same F1-score. While logistic regression achieved a slightly higher AUC-ROC score, the decision tree achieved a higher AUC-PR score.

The confusion matrices for logistic regression and XGBoost are shown in Figure 5.1. Logistic regression identified 139 fraudulent claims, compared to 97 identified by XGBoost. This explains the high recall score of logistic regression. However, it came at the cost of a drastic increase in false positives compared to XGBoost.

The combined ROC plot (Figure 5.2) illustrates the trade-off between true positives and false positives for the different models. The diagonal dotted line shows the baseline for a hypothetical model that randomly makes predictions. The XGBoost model dominates across almost all classification thresholds. The XGBoost achieved the highest AUC-ROC of 0.86, followed by the random forest with a score of 0.83. Logistic regression achieved a score of 0.79, while the decision tree had the lowest score at 0.77. The curve of the decision tree looks more jagged because the decision tree contains a limited number of leaf nodes and, consequently, limited discrete probability thresholds.

As was discussed in Section 3.3, the ROC curve can give overly optimistic results on highly imbalanced data. This is why AUC-PR was used as the main evaluation metric. During Optuna hyperparameter optimization, hyperparameters

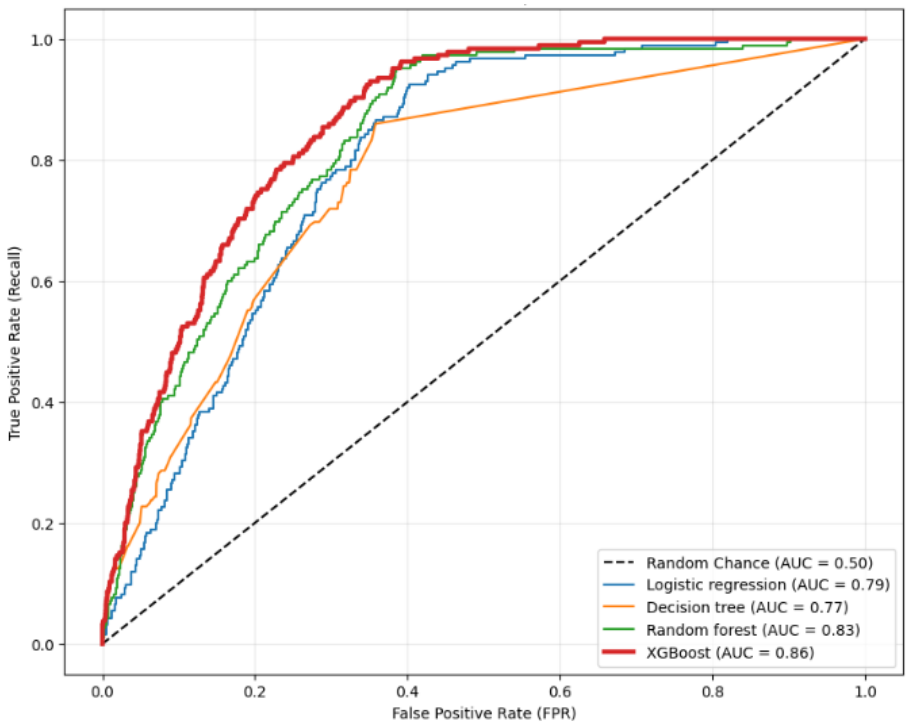

Figure 5.2: Comparison of the ROC curves.

were evaluated based on the AUC-PR score. PR curves for each model are presented in Figure 5.3. XGBoost also dominates the precision-recall across almost all classification thresholds. XGBoost had the highest AUC-PR at 0.27, followed by the random forest (0.22), the decision tree (0.20), and finally logistic regression (0.15).

5.2 Model Interpretability

Now we can evaluate the interpretability of the trained models. XGBoost achieved the highest performance, and while it has built-in feature importance scores, they only indicate the magnitude of a feature's impact, not direction. Furthermore, they cannot explain why a specific prediction was made for an individual instance. For this reason, LIME and SHAP were applied to the XGBoost model. As a comparison with an intrinsically interpretable model, the interpretability of the decision tree is briefly discussed before evaluating LIME and SHAP.

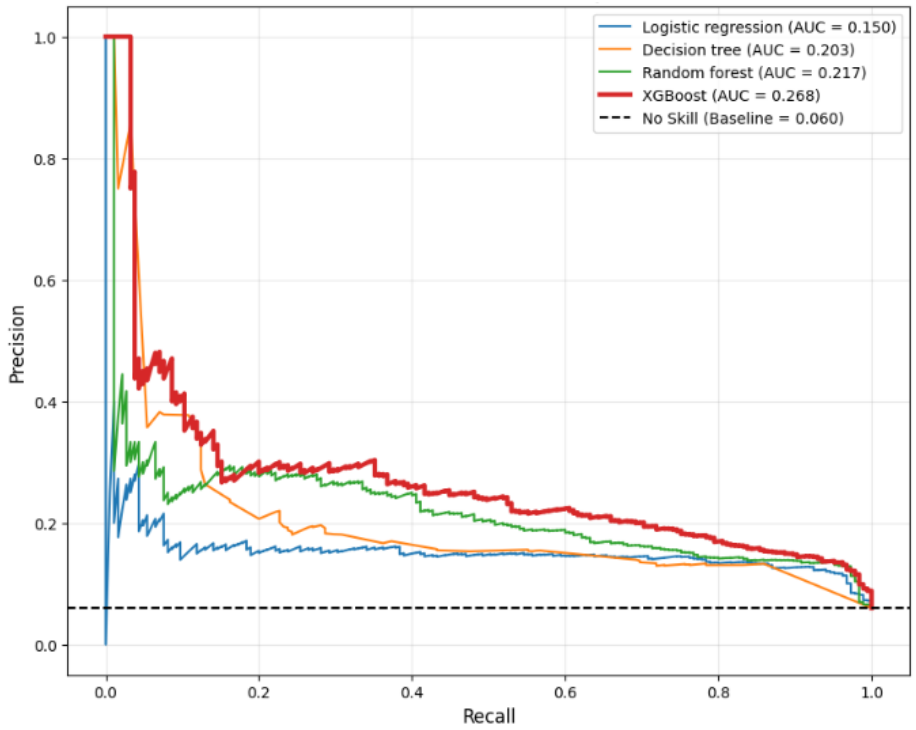

Figure 5.3: Comparison of the PR curves.

5.2.1 Decision Tree Interpretation

A decision tree is intrinsically interpretable, so no additional interpretation methods are needed. A visualization of the top three levels of the decision tree is shown in Figure 5.4. The `BasePolicy` feature with the value "Liability" ended up as the top node, indicating it can be used most efficiently to split the target feature into fraud and non-fraud. For a specific claim, if the condition in the node is true, the claim is routed to the left child node, and if it is not true, the claim is routed to the right. The orange nodes indicate fraud, while the blue nodes indicate a legitimate claim. The stronger the color, the stronger the prediction is. Each claim can be traced through the full decision tree, which shows why the model reached a specific prediction for that claim.

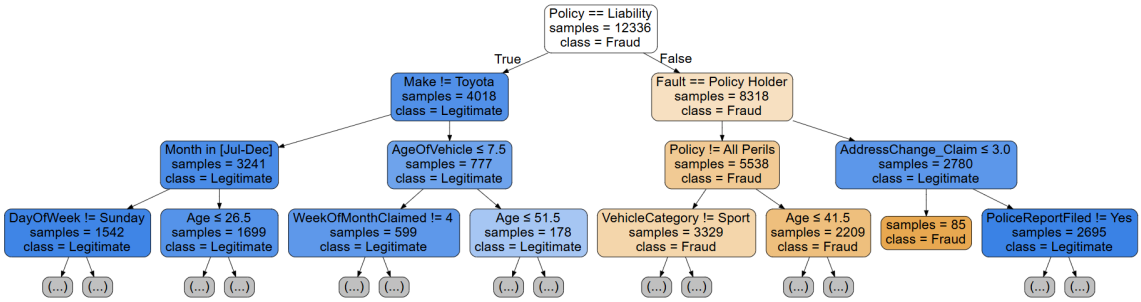

Figure 5.4: Top levels of the decision tree.

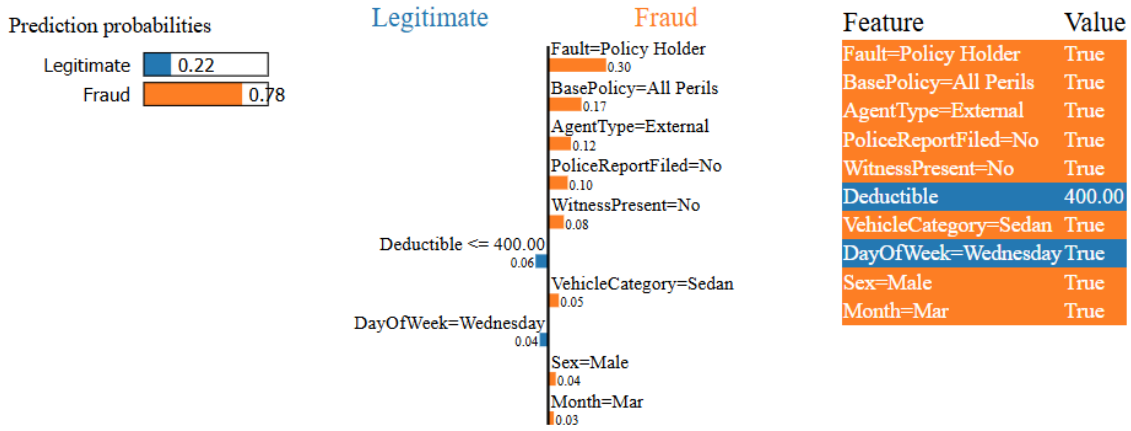

Figure 5.5: Visualization of LIME for an individual claim.

5.2.2 LIME Explanations

LIME was used to generate explanations for the predictions made by the XGBoost model. Figure 5.5 shows the LIME results for an individual true positive claim from the test set. The prediction probabilities for the claim are shown on the left side of the figure. In this case, the model gave it a 22% chance of being legitimate and an 78% chance of being fraudulent. The middle of the figure shows, in order, the most important features and how they influence the prediction probabilities. The right side of the figure shows the values of these features for the specific claim.

In this specific claim, the policyholder's fault in the accident is clearly the most predictive factor. This feature value most strongly pushes the prediction towards fraud. An all-perils base policy, an external agent type, and the absence of a police

report or witnesses also push the prediction toward fraud. On the other hand, the deductible being only 400 and the day of the accident being Wednesday pushes the prediction toward non-fraud.

5.2.3 SHAP Explanations

Figure 5.6 shows the results from the SHAP explanation generation for the XGBoost model. The figure lists 20 features in order of importance from top to bottom. For each feature, there are dots representing individual claims in the test set. Blue instances indicate a low value for the specific feature, and red instances indicate a high value for that feature. For binary features, a red data point indicates the feature is present (a value of 1), and a blue data point indicates it is absent (a value of 0). The position of the instances indicates how strongly they push the model output towards fraud or non-fraud. Data points on the left side push the prediction towards non-fraud, and the points on the right push it towards fraud.

The top features were `Fault_Third Party`, `BasePolicy`, and `Month`. As an example, for the feature `Fault_Third Party`, red data points indicate claims in which the third party was at fault for the accident, and blue data points indicate claims that were the fault of the policyholder. As shown in the figure, third-party fault (red dots) typically strongly supports a non-fraud prediction, while policyholder fault (blue dots) typically pushes the prediction toward fraud. For the third feature, `Month`, we can see that months later in the year (red dots) often push the prediction toward non-fraud.

The plot displays all one-hot-encoded features as separate binary features. For example, the feature `VehicleCategory` is encoded into `VehicleCategory_Sport`, `VehicleCategory_Sedan` etc. This means the feature's importance is split across multiple one-hot-encoded features. To obtain a comprehensive view of global feature importance, the mean absolute SHAP values for all one-hot-encoded features were

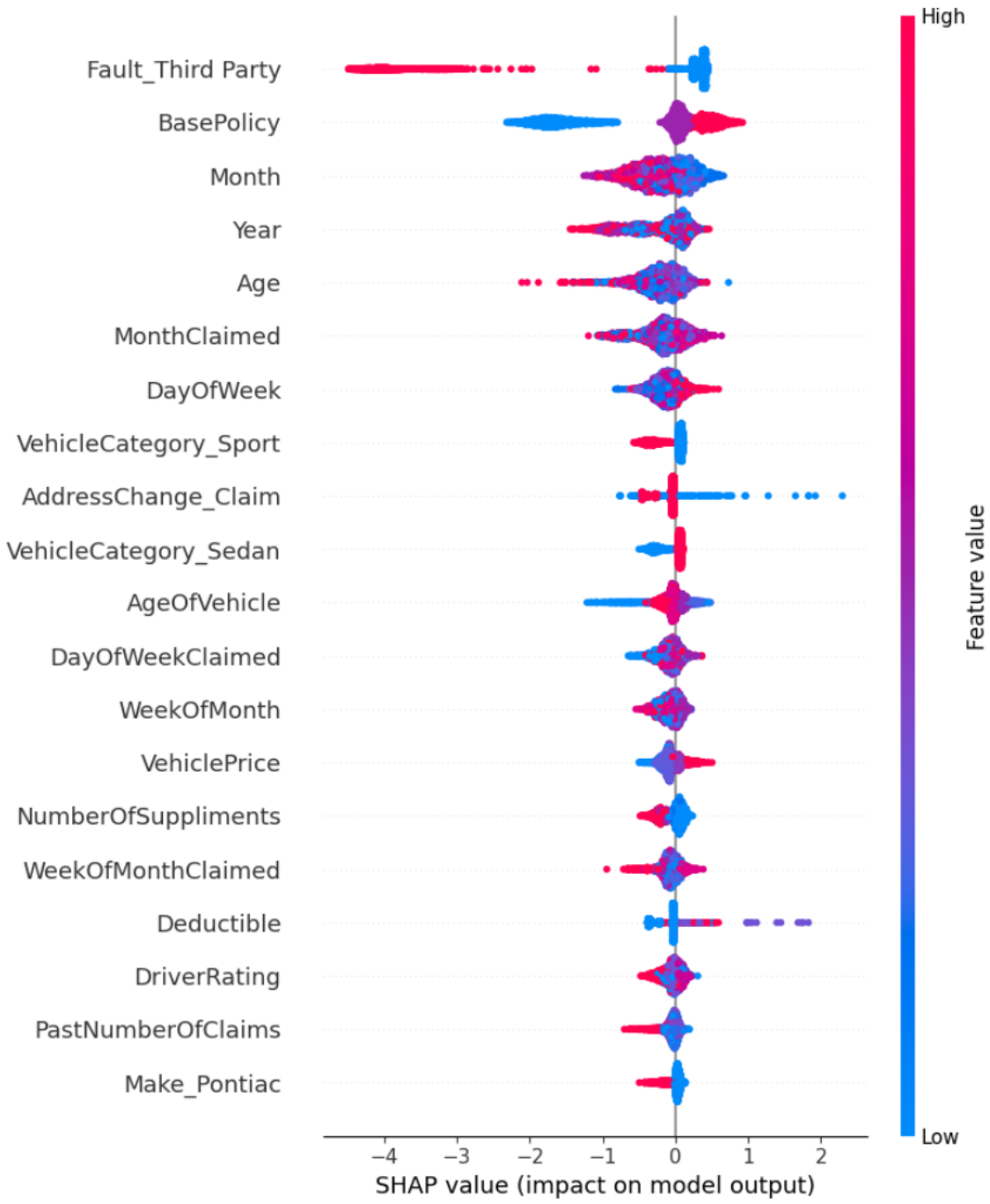

Figure 5.6: Visualization of global SHAP values.

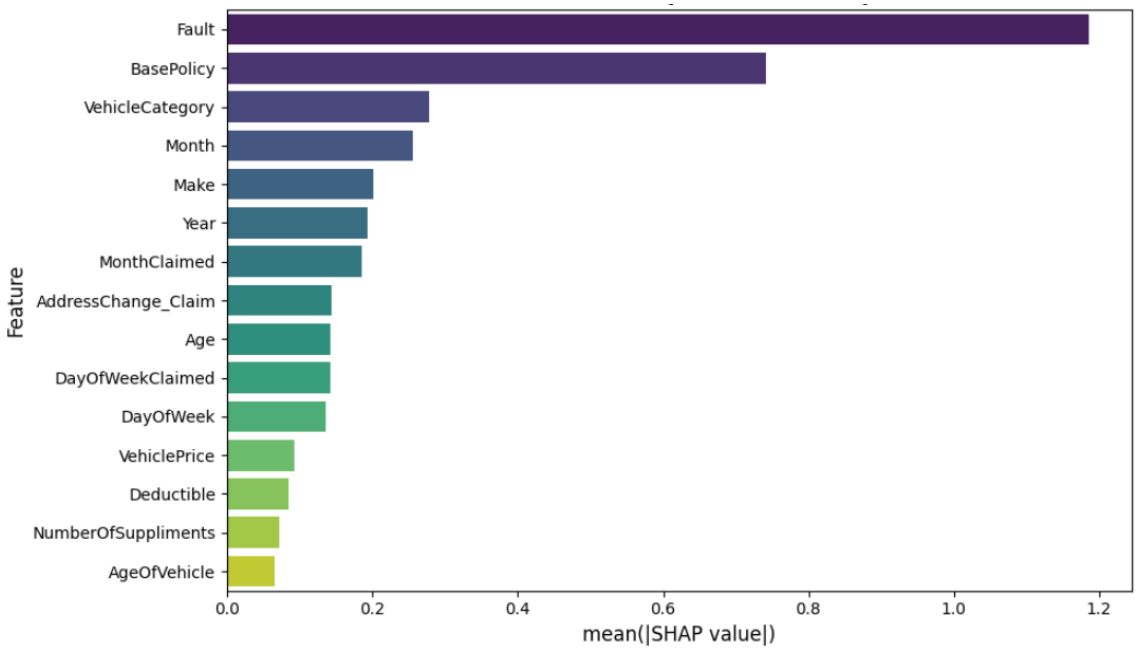

Figure 5.7: Global predictive contribution of the top 15 features.

computed and aggregated.

Figure 5.7 presents the aggregated global importance of the top 15 features. It highlights the key variables that drive the model's fraud-detection capabilities. In the figure, most features are similar to those in the SHAP summary plot. However, vehicle manufacturer now appears as the 5th most important feature, whereas it was not significant in the SHAP summary plot. This suggests that while no single manufacturer heavily influences the model, the overall category is still highly predictive.

SHAP can also be used to generate and visualize local explanations for individual claims. Figure 5.8 shows the SHAP waterfall plot for one of the fraudulent test set claims that was correctly predicted by the model. The features are again listed in order from top to bottom of how much they affect the prediction. In this case, the policyholder's fault in the accident was the primary factor that affected the prediction. Positive numbers (red bars) push the prediction toward fraud, while negative numbers (blue bars) push it toward non-fraud. The baseline value $E[f(X)] = 0.111$ indicates

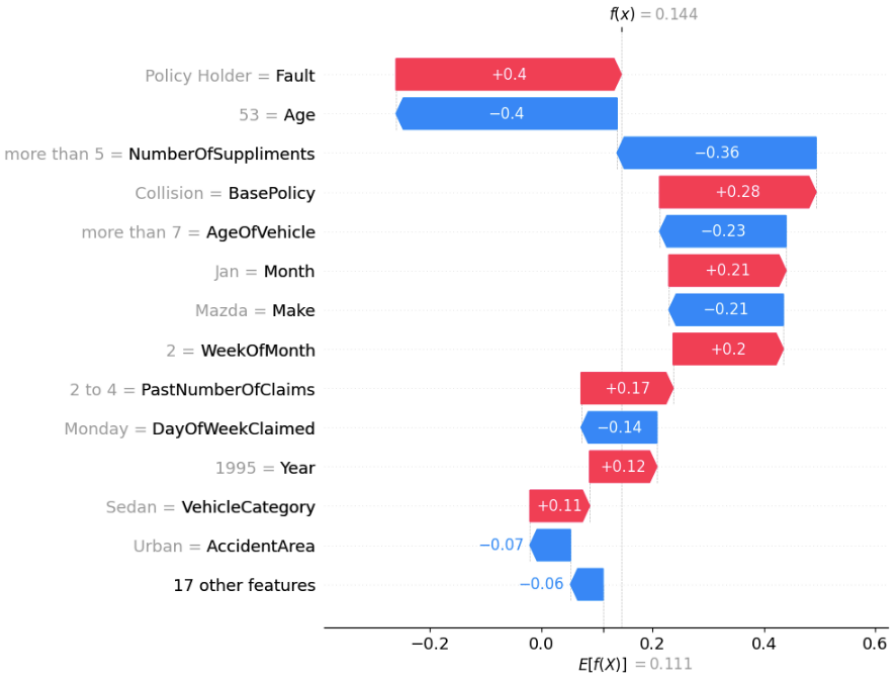

Figure 5.8: SHAP waterfall plot for a true positive claim.

the average prediction the model makes across the entire training dataset. Every waterfall plot starts from this point. If $f(x)$ exceeds the baseline, it is predicted to be more suspicious than the average claim.

In the second example in Figure 5.9, there is another example of a claim that the model correctly identified as fraudulent. In the exploratory data analysis (Section 4.3), it was observed that the fraud rate is significantly higher for people aged 20 or younger. But in this case, the driver's age of 20 actually makes the claim significantly less suspicious. In this combination of features, the model has concluded that the young age is actually not a sign of fraud but a signal that the claim might be legitimate.

Figure 5.10 shows a claim that the model incorrectly identified as fraudulent. We can see that even when most of the other features push the claim towards legitimate, the combination of the top two features of all-perils policy and policyholder's fault is enough to push the prediction heavily to the fraudulent side.

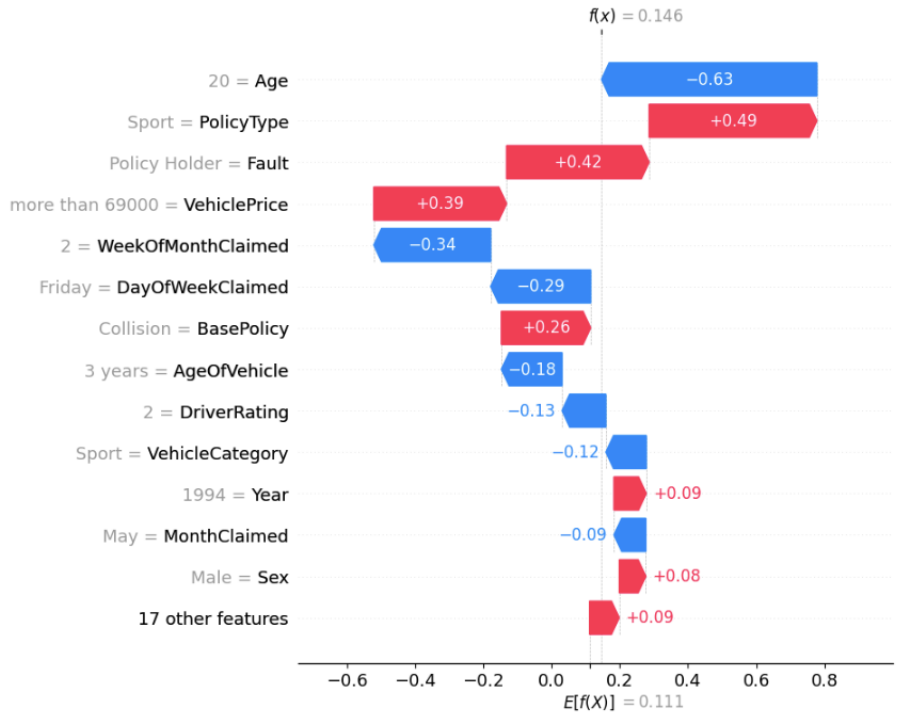

Figure 5.9: SHAP waterfall plot for another true positive claim.

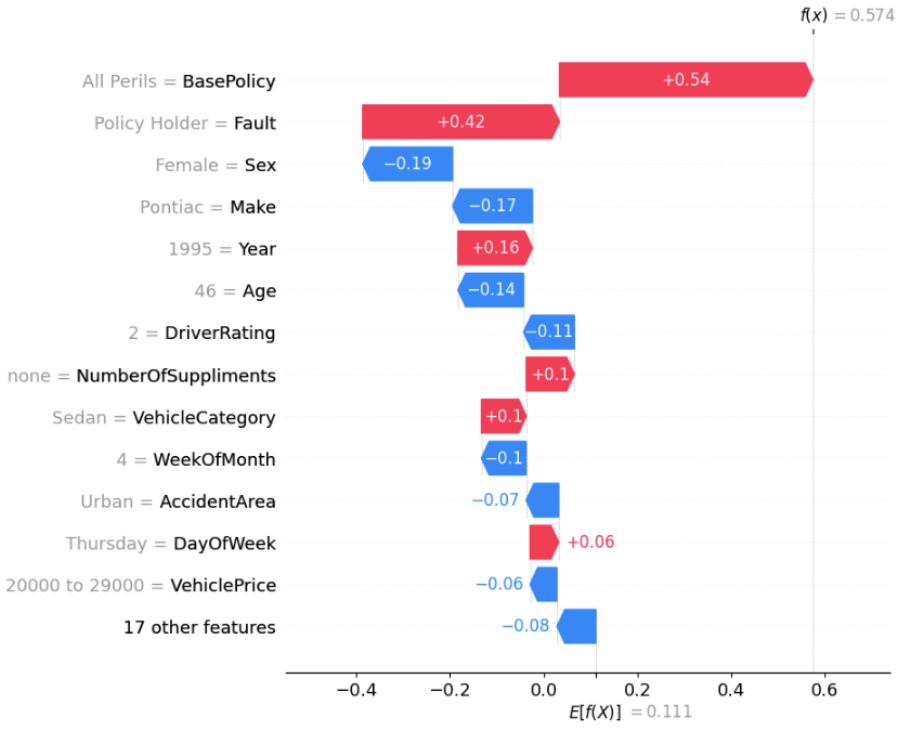

Figure 5.10: SHAP waterfall plot for a false positive claim.

5.3 Discussion

The best-performing model was XGBoost, which was consistent with strong performance in previous literature on fraud detection [40] [41] [27]. Random forest also performed well, which is consistent with previous fraud detection literature [42] [43] [23]. The XGBoost's sequential tree building most likely helped it outperform the independent tree building of the random forest. Logistic regression and decision tree failed to outperform random forest and XGBoost. The poor performance of decision trees was also noticed in previous insurance fraud detection literature [35]. During the exploratory data analysis (Section 4.3), it was observed that the fraud rate is highest with the least expensive and most expensive vehicles. Logistic regression cannot capture this type of non-linear relationship, which might explain the poorer performance. The decision tree was most likely better at detecting these non-linear patterns, which is why it beat logistic regression based on the AUC-PR score.

While it has been argued that it is possible to achieve as good performance with transparent models like logistic regression and decision tree [72], it was not the case in this research setup. The random forest and XGBoost models were able to overcome the limitations of logistic regression and the decision tree. Black-box models have also achieved high performance in previous fraud detection research [40] [41] [27]. This means there is a trade-off between interpretability and performance when selecting a fraud-detection model.

However, even with XGBoost, the performance had room for improvement. The plateauing performance indicates that the dataset may lack the necessary signal to more effectively separate fraudulent claims from legitimate ones. To increase the performance, more features would probably be needed. Features such as claim size could help distinguish between different types of fraud, including soft and hard fraud. Unstructured data sources, such as textual claim descriptions and images, could also significantly improve the performance.

Despite the high number of false positives, in an operational context, the model could still serve as a preliminary filter system. It could help reduce incoming claims to a smaller subset of highly suspicious claims. It is not necessary to perfectly separate the two classes in insurance fraud detection. The goal is to identify the most suspicious claims for manual investigation to focus on. The claims can be ordered by the model's confidence in their fraudulence. The claims handlers could then focus their efforts on the most suspicious cases. This is how fraud detection systems typically work in insurance companies [21].

In the performance evaluation (Figure 5.1), recall and precision were reported for the specific threshold that maximizes the F1-score. The threshold doesn't necessarily have to be this, and in an operational context, the balance between precision and recall can be modified. The PR curves (Figure 5.3) demonstrate this tradeoff. The chosen threshold should be based on the cost of flagging a legitimate customer versus the cost of missing a fraudulent claim. There are many factors that influence the choice of the threshold. These can be, for example, the cost of manually processing a claim and the reputational cost of flagging a legitimate customer [24].

From the perspective of explainability of the predictions, the decision tree model was interpretable, but it did not reach the performance of the more advanced models. This means that it's not the most optimal method for a practical system where fraud detection performance is crucial. Because the highest performing model, XGBoost, was not intrinsically interpretable, SHAP and LIME were used to generate explanations.

Both methods identified similar features to be among the top predictors, such as the base policy and the fault of the accident. However, SHAP did not find the presence of a witness or a police report to be important, whereas LIME often identified them as among the top features. As was seen in the exploratory data analysis (Section 4.3), the features `WitnessPresent` and `PoliceReportFiled` are

very rarely true (0.6% and 2.8%). LIME gives more weight to these features because changing their value from false to true significantly affects the prediction model. This illustrates the strength of SHAP compared to LIME. SHAP accounts for the global distribution of the data, whereas LIME's local surrogate model misinterprets XGBoost's response to changes in these feature values. The value of `WitnessReport` is almost always false, so the absence of witnesses should not significantly contribute to the prediction of fraud, contrary to what LIME suggests. Previous research has also indicated that LIME results can be very similar across different instances and, as a result, less useful to practitioners compared to SHAP [69].

SHAP can also be used to analyze the cause of the large number of false positives. When observing, for example, the claim in Figure 5.10, it becomes clear that the XGBoost model assigns significant weight to the all-perils policy and the policyholder's fault in the accident. This is also true of many other false positive claims. The model relies too heavily on a couple of the most important features when making predictions. This provides further evidence that the dataset lacks the necessary signal to achieve better results. The model itself should not be lacking in capabilities, since XGBoost has achieved impressive results in previous research in fraud detection [40] [41] [27].

This research does not prove that the explanations are actually useful to the claims handlers. As discussed in Chapter 2, the explanations can also lead experts to over-trust the predictions made by the models [71]. However, this research shows that post-hoc explanation tools can be used to make the black-box machine learning models more interpretable. Further research is needed to determine the practical utility of these tools.

Explainable AI methods presented in this study are one potential way to satisfy the requirements of regulations such as the GDPR and the Artificial Intelligence Act. GDPR states that individuals have the right to get explanations for algorithmic

decisions made about them [63]. Further research is needed to determine whether the post-hoc explanations meet the regulatory requirements.

From the explanations, it can be noted that the models used features such as age and sex to make predictions. According to the GDPR, using these features could count as discrimination [63]. Explanation tools can be used to detect whether decisions are made using discriminatory features. Even without these tools, features like age and sex can be easily detected and removed from the system. However, the data may still contain other, more difficult-to-detect features that correlate with discriminatory features [64]. Explanatory tools can possibly help to spot these more hidden correlations.

Post-hoc explanation tools were found to be susceptible to intentional manipulation in previous research [70]. It has even been argued that post-hoc explainers should be abandoned in high-risk domains like finance because they provide approximations rather than truthful computations [72]. Even if the post-hoc explanation tools, by themselves, would not meet regulatory requirements, they could still be used as part of the solution. However, they should not be used as the sole decision-making mechanism.

Keeping these limitations in mind, the post-hoc explanation tools have been demonstrated to be useful in practice in various fields like medicine [67], finance [68], and also specifically in fraud detection [69] [31]. This research gives further evidence that incorporating XAI methods into machine learning fraud detection systems can be highly beneficial.

6 Conclusion

This chapter concludes the thesis. It consists of a summary of the findings, answers to research questions, limitations of the study, and proposals for future research directions.

6.1 Summary of Findings

This thesis compared logistic regression, decision tree, random forest, and XGBoost machine learning models for supervised vehicle insurance fraud detection. XGBoost was found to be the best-performing model based on F1-score, AUC-ROC, and AUC-PR. Random forest had the second-highest performance, beating both the decision tree and logistic regression.

The objective was also to make the models interpretable. The decision tree was shown to be intrinsically interpretable, but it failed to beat ensemble-based methods. This means it is not the optimal choice for a practical fraud detection system. XGBoost outperformed the other models, but it is not interpretable without using additional tools. That is why LIME and SHAP were applied to get explanations for its predictions. LIME was effectively used to obtain the local explanations for single predictions, while SHAP was used to obtain both local and global explanations.

6.2 Answers to Research Questions

Research questions of this thesis were first presented in Chapter 1. This section provides answers to the research questions derived from the research conducted in this thesis.

RQ1: How do specific machine learning models (logistic regression, decision tree, random forest, and XGBoost) compare in their predictive performance when detecting vehicle insurance fraud?

XGBoost achieved the highest performance based on F1-score, AUC-ROC, and AUC-PR. Random forest was the second-best-performing model. The decision tree and logistic regression failed to outperform the black-box models.

RQ2: How can the imbalanced nature of insurance fraud data be addressed when using machine learning based methods

Sampling-based methods, especially SMOTE and its variants, and class-sensitive learning are effective for handling imbalanced data. Based on the categorical nature of the dataset and the aim to incorporate post-hoc explanation tools, cost-sensitive learning was found to be the most suitable method in the context of this thesis.

RQ3: Can explainable artificial intelligence methods be used to interpret the predictions of fraud detection machine learning models?

LIME and SHAP were successfully applied to the XGBoost model to get explanations for the predictions. SHAP was used to generate both local and global explanations, while LIME was used to generate local explanations. SHAP was found to be a more reliable and robust method compared to LIME.

6.3 Limitations

The biggest limitation in this study was the dataset used. The overall quality of the dataset is unclear, and the details of how it was acquired are unavailable. This means that the quality of the labels cannot be evaluated, and there could be a significant number of undetected fraud in the dataset, potentially affecting the results. The dataset was also quite old, as the claims were from 1994-1996. Fraud tactics might have evolved since then. The quality of datasets is a more fundamental problem of insurance fraud research, since there are very few publicly available fraud datasets. There is no motivation for insurance companies to share their private data for scientific research.

Supervised learning was used, meaning that only the already detected fraud was considered during training. The system would not be able to detect fraud types that have not been previously flagged in the data. This is a fundamental problem in supervised-learning-based fraud detection, and combining unsupervised methods would make the fraud detection system more robust.

Another limitation with the methodology is that the problem was modeled as a binary classification task. Fraud with varying quality and severity, such as soft and hard fraud, is combined into a single positive class. The characteristics and investigative approaches of these fraud types may differ significantly, suggesting that the problem could rather be modeled as a multi-class classification problem.

A limitation in the incorporation of XAI tools is the lack of a robust way to evaluate the quality of predictions from the post-hoc explanation tools. The research was conducted independently from the insurance industry, so it was not possible to study whether claims handlers would actually benefit from the explanations provided by SHAP and LIME.

6.4 Future Work

This research used only supervised learning, which can detect known fraud patterns. Unsupervised learning methods could be incorporated into the system to detect unknown fraud patterns, thereby potentially improving overall performance. The supervised and unsupervised systems could be either run in parallel, or the unsupervised model's output could be used as a feature in supervised model training.

Additional sources of data could also be incorporated into the prediction models. This could include unstructured data sources, such as textual descriptions and images that are often attached to claims. This might have to be done in collaboration with insurance companies because of the lack of publicly available datasets. This study considered only vehicle insurance, so it could be studied how the methods presented here apply to other insurance types, assuming suitable datasets are available.

Next steps could also include applying and testing the fraud detection system proposed in this thesis in a real business context at an insurance company. This way, it could be evaluated whether the claims adjusters benefit from the predictions and explanations provided by the system.

References

[1] "Allianz Global Insurance Report 2025: Rising demand for protection | Allianz", Allianz.com, Accessed: Jan. 22, 2026. [Online]. Available: https://www.allianz.com/en/economic_research/insights/publications/specials_fmo/250527-global-insurance-report.html.

[2] "Annual report 2023-2024: Fraud", Insurance Europe, Accessed: Jan. 18, 2026. [Online]. Available: http://www.insuranceeurope.eu/.

[3] "Fraud Stats", Coalition Against Insurance Fraud, Accessed: Feb. 13, 2026. [Online]. Available: https://insurancefraud.org/fraud-stats/.

[4] "Fraudulent insurance claims continue to top £1 billion | ABI", Accessed: Mar. 12, 2026. [Online]. Available: https://www.abi.org.uk/news/news-articles/2025/11/fraudulent-insurance-claims-continue-to-top-1-billion/.

[5] "Vakuutuspetos on rikos ja rikoksella on seuraamuksia – yhtiöiden tutkinnassa viime vuonna 2 500 epäselvää vahinkoilmoitusta", Finanssiala, Accessed: Nov. 27, 2025. [Online]. Available: https://www.finanssiala.fi/uutiset/vakuutuspetos-on-rikos-ja-rikoksella-on-seuraamuksia-yhtioiden-tutkinnassa-viime-vuonna-2-500-epaselvaa-vahinkoilmoitusta/.

[6] "Investigating Insurance Fraud", Federal Bureau of Investigation, Accessed: Mar. 28, 2026. [Online]. Available: https://www.fbi.gov/news/stories/investigating-insurance-fraud.

[7] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*, Apr. 27, 2016. Accessed: Feb. 12, 2026. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/oj.

[8] *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*, Jun. 13, 2024. Accessed: Mar. 15, 2026. [Online]. Available: http://data.europa.eu/eli/reg/2024/1689/oj.

[9] "Vehicle Insurance Claim Fraud Detection", Accessed: Dec. 8, 2025. [Online]. Available: https://www.kaggle.com/datasets/shivamb/vehicle-claim-fraud-detection.

[10] R. A. Derrig, "Insurance Fraud", *Journal of Risk and Insurance*, vol. 69, no. 3, pp. 271–287, 2002. DOI: 10.1111/1539-6975.00026.

[11] S. Tennyson, "Moral, Social, and Economic Dimensions of Insurance Claims Fraud", *Social Research: An International Quarterly*, vol. 75, pp. 1181–1204, Dec. 1, 2008. DOI: 10.1353/sor.2008.0020.

[12] "Insurance Research Council Finds That Fraud and Buildup Add Up to $7.7 Billion in Excess Payments for Auto Injury Claims | IRC", Accessed: Mar. 12, 2026. [Online]. Available: https://insurance-research.org/news/insurance-research-council-finds-fraud-and-buildup-add-77-billion-excess-payments-auto-injury.

[13] S. Viaene and G. Dedene, "Insurance Fraud: Issues and Challenges", *The Geneva Papers on Risk and Insurance - Issues and Practice*, vol. 29, no. 2, pp. 313–333, Apr. 1, 2004. DOI: 10.1111/j.1468-0440.2004.00290.x.

[14] L. Šubelj, Š. Furlan, and M. Bajec, "An expert system for detecting automobile insurance fraud using social network analysis", *Expert Systems with Applications*, vol. 38, no. 1, pp. 1039–1052, Jan. 1, 2011. DOI: 10.1016/j.eswa.2010.07.143.

[15] C. Phua, V. Lee, K. Smith, and R. Gayler, "A Comprehensive Survey of Data Mining-based Fraud Detection Research", *Computers in Human Behavior*, vol. 28, no. 3, pp. 1002–1013, May 2012. DOI: 10.1016/j.chb.2012.01.002.

[16] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey", *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, Jun. 1, 2016. DOI: 10.1016/j.jnca.2016.04.007.

[17] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey", *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, Jul. 2009. DOI: 10.1145/1541880.1541882.

[18] G. Pang, C. Shen, L. Cao, and A. van den Hengel, "Deep Learning for Anomaly Detection: A Review", *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, Mar. 31, 2022. DOI: 10.1145/3439950.

[19] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature", *Decision Support Systems*, On Quantitative Methods for Detection of Financial Fraud, vol. 50, no. 3, pp. 559–569, Feb. 1, 2011. DOI: 10.1016/j.dss.2010.08.006.

[20] M. Tümmler and R. Quick, "How to detect fraud in an audit: A systematic review of experimental literature", *Management Review Quarterly*, Jan. 7, 2025. DOI: 10.1007/s11301-024-00480-7.

[21] R. J. Bolton and D. J. Hand, "Statistical Fraud Detection: A Review", *Statistical Science*, vol. 17, no. 3, pp. 235–255, Aug. 2002. DOI: 10.1214/ss/1042727940.

[22] J. Li, K.-Y. Huang, J. Jin, and J. Shi, "A survey on statistical methods for health care fraud detection", *Health care management science*, vol. 11, pp. 275–87, Oct. 1, 2008. DOI: 10.1007/s10729-007-9045-4.

[23] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective", *Expert Systems with Applications*, vol. 41, no. 10, pp. 4915–4928, Aug. 2014. DOI: 10.1016/j.eswa.2014.02.026.

[24] K. Crocker and S. Tennyson, "Insurance Fraud and Optimal Claims Settlement Strategies", *Journal of Law and Economics*, vol. 45, pp. 469–507, Feb. 1, 2002. DOI: 10.1086/340394.

[25] F. T. Liu, K. Ting, and Z.-H. Zhou, "Isolation Forest", Jan. 19, 2009, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[26] E. Stripling, B. Baesens, B. Chizi, and S. vanden Broucke, "Isolation-based conditional anomaly detection on mixed-attribute data to uncover workers' compensation fraud", *Decision Support Systems*, vol. 111, pp. 13–26, Jul. 1, 2018. DOI: 10.1016/j.dss.2018.04.001.

[27] J. Debener, V. Heinke, and J. Kriebel, "Detecting insurance fraud using supervised and unsupervised machine learning", *Journal of Risk and Insurance*, vol. 90, no. 3, pp. 743–768, 2023. DOI: 10.1111/jori.12427.

[28] C. Gomes, Z. Jin, and H. Yang, "Insurance fraud detection with unsupervised deep learning", *Journal of Risk and Insurance*, vol. 88, no. 3, pp. 591–624, 2021. DOI: 10.1111/jori.12359.

[29] K. Nian, H. Zhang, A. Tayal, T. Coleman, and Y. Li, "Auto insurance fraud detection using unsupervised spectral ranking for anomaly", *The Journal of Finance and Data Science*, vol. 2, no. 1, pp. 58–75, Mar. 1, 2016. DOI: 10.1016/j.jfds.2016.03.001.

[30] P. L. Brockett, X. Xia, and R. A. Derrig, "Using Kohonen's Self-Organizing Feature Map to Uncover Automobile Bodily Injury Claims Fraud", *The Journal of Risk and Insurance*, vol. 65, no. 2, pp. 245–274, 1998. DOI: 10.2307/253535.

[31] H. De Meulemeester, F. De Smet, J. van Dorst, and E. D. D. Moor, "Explainable unsupervised anomaly detection for healthcare insurance data", *BMC Medical Informatics and Decision Making*, vol. 25, pp. 1–11, 2025. DOI: 10.1186/s12911-024-02823-6.

[32] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "ADBench: Anomaly detection benchmark", in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc., Nov. 28, 2022, pp. 32 142–32 159. DOI: 10.52202/068431-2329.

[33] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection", *Information Sciences*, vol. 557, pp. 317–331, May 1, 2021. DOI: 10.1016/j.ins.2019.05.042.

[34] Ghosh and Reilly, "Credit card fraud detection with a neural-network", in *1994 Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*, vol. 3, Jan. 1994, pp. 621–630. DOI: 10.1109/HICSS.1994.323314.

[35] S. Viaene, R. A. Derrig, B. Baesens, and G. Dedene, "A Comparison of State-of-the-Art Classification Techniques for Expert Automobile Insurance Claim Fraud Detection", *Journal of Risk and Insurance*, vol. 69, no. 3, pp. 373–421, 2002. DOI: 10.1111/1539-6975.00023.

[36] F. Aslam, A. I. Hunjra, Z. Ftiti, W. Louhichi, and T. Shams, "Insurance fraud detection: Evidence from artificial intelligence and machine learning", *Research in International Business and Finance*, vol. 62, p. 101 744, Dec. 1, 2022. DOI: 10.1016/j.ribaf.2022.101744.

[37] A. K. I. Hassan and A. Abraham, "Modeling Insurance Fraud Detection Using Imbalanced Data Classification", in *Advances in Nature and Biologically Inspired Computing*, N. Pillay, A. P. Engelbrecht, A. Abraham, M. C. du Plessis, V. Snášel, and A. K. Muda, Eds., Springer International Publishing, 2016, pp. 117–127. DOI: 10.1007/978-3-319-27400-3_11.

[38] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks", *Journal of Big Data*, vol. 7, no. 1, pp. 1–41, Apr. 1, 2020. DOI: 10.1186/s40537-020-00305-w.

[39] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?", in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc., Nov. 28, 2022, pp. 507–520. DOI: 10.48550/arXiv.2207.08815.

[40] A. Q. Abdulghani, O. N. UCAN, and K. M. A. Alheeti, "Credit Card Fraud Detection Using XGBoost Algorithm", in *2021 14th International Conference on Developments in eSystems Engineering (DeSE)*, Dec. 2021, pp. 487–492. DOI: 10.1109/DeSE54285.2021.9719580.

[41] N. Averro, H. Murfi, and G. Ardaneswari, "The Imbalance Data Handling of XGBoost in Insurance Fraud Detection:" in *Proceedings of the 12th International Conference on Data Science, Technology and Applications*, SCITEPRESS - Science and Technology Publications, 2023, pp. 460–467. DOI: 10.5220/0012126900003541.

[42] E. Nabrawi and A. Alanazi, "Fraud Detection in Healthcare Insurance Claims Using Machine Learning", *Risks*, vol. 11, no. 9, Sep. 5, 2023. DOI: 10.3390/risks11090160.

[43] B. Itri, Y. Mohamed, Q. Mohammed, and B. Omar, "Performance comparative study of machine learning algorithms for automobile insurance fraud detection", in *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, Oct. 2019, pp. 1–4. DOI: 10.1109/ICDS47004.2019.8942277.

[44] Z. Wang, X. Chen, Y. Wu, L. Jiang, S. Lin, and G. Qiu, "A robust and interpretable ensemble machine learning model for predicting healthcare insurance fraud", *Scientific Reports*, vol. 15, no. 1, p. 218, Jan. 2, 2025. DOI: 10.1038/s41598-024-82062-x.

[45] M. Artís, M. Ayuso, and M. Guillén, "Detection of Automobile Insurance Fraud With Discrete Choice Models and Misclassified Claims", *Journal of Risk and Insurance*, vol. 69, no. 3, pp. 325–340, 2002. DOI: 10.1111/1539-6975.00022.

[46] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data", in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas Nevada USA: ACM, Aug. 24, 2008, pp. 213–220. DOI: 10.1145/1401890.1401920.

[47] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, Jul. 2012. DOI: 10.1109/TSMCC.2011.2161285.

[48] C. Elkan, "The Foundations of Cost-Sensitive Learning", *Proceedings of the Seventeenth International Conference on Artificial Intelligence: 4-10 August 2001; Seattle*, vol. 1, May 12, 2001.

[49] H. He and E. A. Garcia, "Learning from Imbalanced Data", *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009. DOI: 10.1109/TKDE.2008.239.

[50] G. Batista, R. Prati, and M.-C. Monard, "A Study of the Behavior of Several Methods for Balancing machine Learning Training Data", *SIGKDD Explorations*, vol. 6, pp. 20–29, Jun. 1, 2004. DOI: 10.1145/1007730.1007735.

[51] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 1, 2002. DOI: 10.1613/jair.953.

[52] R. Bounab, K. Zarour, B. Guelib, and N. Khlifa, "Enhancing Medicare Fraud Detection Through Machine Learning: Addressing Class Imbalance With SMOTE-ENN", *IEEE Access*, vol. 12, pp. 54 382–54 396, Jan. 1, 2024. DOI: 10.1109/access.2024.3385781.

[53] Z. Khan, D. Olivia, and S. Shetty, "A Machine Learning and Explainable Artificial Intelligence Approach for Insurance Fraud Classification", Jan. 1, 2025. DOI: 10.4114/intartif.vol28iss75pp140-169.

[54] Y. Pristyanto, A. F. Nugraha, A. Dahlan, L. A. Wirasakti, A. Ahmad Zein, and I. Pratama, "Multiclass Imbalanced Handling using ADASYN Oversampling and Stacking Algorithm", in *2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Jan. 2022, pp. 1–5. DOI: 10.1109/IMCOM53663.2022.9721632.

[55] C. Chen and L. Breiman, "Using Random Forest to Learn Imbalanced Data", *University of California, Berkeley*, Jan. 1, 2004. Accessed: Mar. 28, 2026. [Online]. Available: http://xtf.lib.berkeley.edu/reports/SDTRWebData/accessPages/666.html.

[56] G. M. Weiss, K. McCarthy, and B. Zabar, "Cost-Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs?", in *Proceedings of the 2007 International Conference on Data Mining,*

DMIN 2007, June 25-28, 2007, Las Vegas, Nevada, USA, CSREA Press, 2007, pp. 35–41.

[57] M. Mukherjee and M. Khushi, "SMOTE-ENC: A Novel SMOTE-Based Method to Generate Synthetic Data for Nominal and Continuous Features", *Applied System Innovation*, vol. 4, no. 1, p. 18, Mar. 2021. DOI: 10.3390/asi4010018.

[58] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data", *BMC Bioinformatics*, vol. 14, no. 1, p. 106, Mar. 22, 2013. DOI: 10.1186/1471-2105-14-106.

[59] U. Zafar and F. Wu, "Methodological challenges in explainable AI for fraud detection: A systematic literature review", *Artificial Intelligence Review*, Feb. 17, 2026. DOI: 10.1007/s10462-026-11516-7.

[60] D. Cemernek, S. Siddiqi, and R. Kern, "Effects of Class Imbalance Countermeasures on Interpretability", *IEEE Access*, vol. 12, pp. 45 342–45 358, 2024. DOI: 10.1109/ACCESS.2024.3381536.

[61] A. Barredo Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI", *Information Fusion*, vol. 58, pp. 82–115, Jun. 1, 2020. DOI: 10.1016/j.inffus.2019.12.012.

[62] Z. Lipton, "The Mythos of Model Interpretability", *Communications of the ACM*, vol. 61, Oct. 6, 2016. DOI: 10.1145/3233231.

[63] B. Goodman and S. Flaxman, "European Union regulations on algorithmic decision-making and a "right to explanation"", *AI Magazine*, vol. 38, no. 3, pp. 50–57, Sep. 2017. DOI: 10.1609/aimag.v38i3.2741.

[64] M. Leese, "The new profiling: Algorithms, black boxes, and the failure of anti-discriminatory safeguards in the European Union", *Security Dialogue*, vol. 45, no. 5, pp. 494–511, Oct. 1, 2014. DOI: 10.1177/0967010614544204.

[65] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Aug. 13, 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

[66] S. Tonekaboni, S. Joshi, M. D. McCradden, and A. Goldenberg, "What Clinicians Want: Contextualizing Explainable Machine Learning for Clinical End Use", in *Proceedings of the 4th Machine Learning for Healthcare Conference*, PMLR, Oct. 28, 2019, pp. 359–380. DOI: 10.48550/arXiv.1905.05134.

[67] S. M. Lundberg et al., "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery", *Nature biomedical engineering*, vol. 2, no. 10, pp. 749–760, Oct. 2018. DOI: 10.1038/s41551-018-0304-0.

[68] P. Bracke, A. Datta, C. Jung, and S. Sen, "Machine learning explainability in finance: An application to default risk analysis", *Bank of England working papers*, no. 816, Aug. 9, 2019. DOI: 10.2139/ssrn.3435104.

[69] S. Jesus et al., "How can I choose an explainer? An Application-grounded Evaluation of Post-hoc Explanations", in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Mar. 3, 2021, pp. 805–815. DOI: 10.1145/3442188.3445941.

[70] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods", *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186, Feb. 7, 2020. DOI: 10.1145/3375627.3375830.

[71] H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, and J. Wortman Vaughan, "Interpreting Interpretability: Understanding Data Scientists' Use of Interpretability Tools for Machine Learning", in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20, Association

for Computing Machinery, Apr. 23, 2020, pp. 1–14. DOI: 10.1145/3313831.3376219.

[72] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead", *Nature Machine Intelligence*, vol. 1, pp. 206–215, May 1, 2019. DOI: 10.1038/s42256-019-0048-x.

[73] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework", Jul. 25, 2019. DOI: 10.48550/arXiv.1907.10902.

[74] D. Powers and Ailab, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation", *J. Mach. Learn. Technol*, vol. 2, pp. 2229–3981, Jan. 1, 2011. DOI: 10.9735/2229-3981.

[75] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves", in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Association for Computing Machinery, Jun. 25, 2006, pp. 233–240. DOI: 10.1145/1143844.1143874.

[76] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees", *Nature Machine Intelligence*, vol. 2, no. 1, pp. 56–67, Jan. 2020. DOI: 10.1038/s42256-019-0138-9.

[77] T. Hastie, R. Tibshirani, and J. Friedman, "Linear Methods for Classification", in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, T. Hastie, R. Tibshirani, and J. Friedman, Eds., Springer, 2009, pp. 101–137. DOI: 10.1007/978-0-387-84858-7_4.

[78] T. Hastie, R. Tibshirani, and J. Friedman, "Additive Models, Trees, and Related Methods", in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, T. Hastie, R. Tibshirani, and J. Friedman, Eds., Springer, 2009, pp. 295–336. DOI: 10.1007/978-0-387-84858-7_9.

[79] L. Breiman, "Random Forests", *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 1, 2001. DOI: 10.1023/A:1010933404324.

[80] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, Association for Computing Machinery, Aug. 13, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[81] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions", in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Curran Associates Inc., Dec. 4, 2017, pp. 4768–4777. DOI: 10.48550/arXiv.1705.07874.

[82] L. S. Shapley, "A Value for n-Person Games", in *Contributions to the Theory of Games (AM-28), Volume II*, Princeton University Press, Dec. 31, 1953, pp. 307–318. DOI: 10.1515/9781400881970-018.

[83] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, vol. 12, Jan. 2, 2012. DOI: 10.48550/arXiv.1201.0490.

[84] P. Chapman, "CRISP-DM 1.0: Step-by-step data mining guide", SPSS Inc., 2000. Accessed: Feb. 26, 2026. [Online]. Available: https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf.