Use Case—Fraud Detection Using Machine Learning Techniques

Philipp Enzinger and Sangmeng Li

1 Introduction

The cost of fraudulent claims for insurers was estimated at around 13 billion euros in 2017 and is second only to tax fraud. Only fraud cases in the amount of 2.5 billion euros have been detected. That is less than one in five fraud cases. Though initiatives to counter insurance fraud exist in many European countries, some were started as early as 1989, detection rates remain low (insurance europe 2019). Fraud leads to competitive disadvantages, moral hazards, adverse selection, reputational damage, and other indirect disadvantages in addition to falsely paid claims. Therefore, insurance companies have tried to detect and prevent fraud manually and automated long before the evolution of machine learning (ML) techniques. They were just not very successful in doing so—right now only one in ten fraudulent claims is correctly detected by insurance companies in Germany. In addition to susceptibility to errors, manual fraud detection processes are complex in maintenance and cost-intensive, not least because forms of fraud

P. Enzinger (✉) · S. Li
ifb SE, Grünwald, Germany
e-mail: Philipp.Enzinger@ifb-group.com

S. Li
e-mail: Sangmeng.Li@ifb-group.com

V. Liermann and C. Stegmann (eds.), *The Digital Journey of Banking and Insurance, Volume II*, https://doi.org/10.1007/978-3-030-78829-2_3

are constantly changing. Added to this is the increasing digitalization of the insurance business and the resulting new data sources.

In times where insurers are constantly looking for new investment opportunities while facing enormous pressure to drive down costs to fulfill the required returns that are needed to stay profitable, the untapped potential of investing in automated fraud detection therefore sounds like a no-brainer. We are going to look into why that is the case, what are the challenges insurers are facing when setting up automated fraud detection systems with ML, and what a process can look like in the new model.

2 A Primer on Insurance Fraud

An insurance contract is a socialization of costs that could endanger the financial welfare of every individual but are bearable when allocated to a large pool of homogeneous individuals. In practice, one of the biggest challenges is the identification of the real costs of the damage the insured individual suffered and whether the circumstances under which the damage was incurred are covered by the insurance contract. These are the two most significant sources of insurance fraud and they basically come down to an information asymmetry problem. This is one of the biggest challenges in avoiding insurance fraud—decreasing the information asymmetry between the insurer and the insured. This is often called "costly state verification" and means that verifying the information the insured individual provides to the insurer comes at a cost. Machine learning is one tool to decrease these costs. Unfortunately, machine learning requires good data quality. But, luckily, insurers have spent the past decade or so **increasing the data availability** (Velauthapillai and Floß 2021), for example, for car insurance contracts. This means that they realized it makes sense to use weather data and geo data when evaluating whether a particular claim is fraudulent or not. But it is not only important to evaluate the data of the claim. You must also take into account the **data at the inception of the contract**. The data might already have been incomplete then, which results in a lack of coverage for the claim. This is a very common phenomenon in health insurance. Fraud can also differ significantly in the degree of criminal energy behind the fraudulent claim. So-called **soft frauds** are opportunistic in nature. The typical case is when your friend claims for the cell phone that you broke yourself using his liability insurance. In contrast, **hard frauds** are planned from the start, often beginning with the deliberate destruction of an insured object. Hard frauds might even become more significant when the insurance company gets a reputation for being lax in fraud detection. It will then fall victim to organized crime syndicates more often. This is known as adverse selection. Knowing this differentiation is important when setting up a fraud detection system because the structure of the data can vary significantly.

Combating fraud is key for insurance companies because high fraud rates pose in themselves a significant competitive disadvantage as the claims paid out are higher than the true insured risk, which results **in higher premiums** and/or lower margins. In addition to this, not being able to "know" the true insured risk also hinders companies in setting up good models as the estimator is always biased. This also poses a significant **model risk** as pricing and customer segmentation models take the cost of fraud falsely into account. Being known as an insurer with lax fraud guidelines can also lead to significant **reputational risks**. Aside from the adverse selection problem mentioned, this can also be alarming for regulators (AML, tax authorities…) and other stakeholders in the company—in particular, shareholders because they end up paying the cost and can opt to invest in a more efficient competitor. Finally, the **financial loss** itself should already be a high enough motivator to try to increase the fraud detection rate as far as possible.

2.1 Current State of Fraud Detection

A study by McKinsey found the following flaws in existing fraud detection systems from an organizational point of view. First of all, fraud is still not a focus topic of top management, although McKinsey estimates potential savings exceeding three percent of claims expenditure. Additionally, the claims department is of limited importance in the organizational and strategic focus of insurance companies. Drilling down into the claims department, fraud detectors historically are the same employees as regular claim handlers—without any additional specializations (McKinsey & Company 2015).

The issue of bad fraud detection systems and high fraud rates became so prominent that regulatory and national institutions decided to act on it and created industry-wide programs around the world to tackle this problem. Some countries set up **dedicated investigative groups**. France has an investigative body for fraud detection that provides training and certificates for fraud officers. The UK set up an Insurance Fraud Bureau. Some European countries directly **cooperate with law enforcement** as police officers are often very well equipped to detect fraudulent activities, in particular when they stem from organized crime. Maybe the most important development in fraud detection is the use of Big Data and new technologies to improve fraud detection systems. Insurance companies are increasingly making use of external databases, e.g., tax authorities' data. National authorities are supporting this, for example, the UNESPA, the national insurance association of Spain, which provides **two common databases** of all motor and property insurance claims which can be used to train and improve fraud detection models. Similar networks with shared data across the industry exist in the UK, Slovenia, Italy, and other countries. Thinking this through to the end, a modern, digitalized insurance company must **employ machine learning** to best make use of the Big Data available. The largest Belgian insurance company did exactly that and created an integrated solution based on ML for its fraud detection model. Insurance Europe, the European insurance federation, recommends making use of Big Data and artificial intelligence in fraud detection and making it a board-level issue (insurance europe 2019).

3 Use Case: Fraud Detection with Machine Learning on Car Insurance Claims

3.1 Data Collection and Preparation

The analysis in the paper is based on the public dataset from Kaggle (Kaggle 2018) including 1000 insurance claim samples, where nearly a quarter of them are fraudulent. The data is slightly imbalanced and the minority class (fraud class) is the one that we are interested in identifying. As introduced in (Liermann et al., Mathematical Background of Machine Learning 2019), imbalanced data is a common problem in classification where the classes are distributed unequally in the dataset. This affects model training and causes underfitting, where the model can recognize patterns from the majority class but ignores the minority class. In this article, we will use the oversampling technique to adjust the class distribution by duplicating the minority class (Fig. 1).

39 attributes are included in the dataset where the attribute "fraud_reported" is binary and shows whether the claim is fraudulent or not. The attribute "fraud_reported" will be the "label," i.e., the information the model is aiming to predict. The other 38 attributes are called features. Some attributes contain numerical values, and some are categorical. Examples are given in the following figures. Note that the data attributes are all considered numeric, although they are discrete, since they are measurements and have a mathematical meaning (Tables 1 and 2).

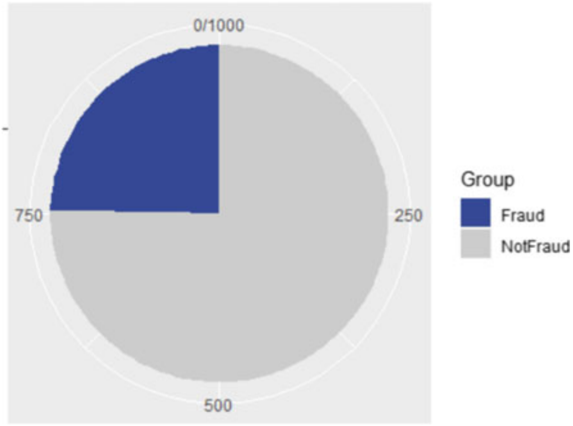


Fig. 1 Imbalanced dataset (© ifb SE)

Before we start to train the model, the data needs to be cleansed according to the following two steps:

Remove the identity attributes

Some attributes, such as identification number, are unique per claim and should not be used for training the model. In our dataset, we discovered two columns: policy_number and incident_location.

Remove the missing value

The attributes containing missing values should be removed. In more advanced cases, the missing value should be fulfilled by using average values or sampling methods. We only discovered one attribute X_c39, therefore it makes sense to ignore it.

In the more general case, we should take some further steps to clean the data, such as removing duplicate observations, removing attributes with very few unique values, and standardization for numerical values.

The following analysis is implemented by using the R and machine learning open-source platform H2O (H2O.ai 2019).

3.2 Model Selection

As a supervised learning model, the algorithms usually learn on a labeled dataset and the label is the target that needs to be predicted. In our example, the data samples are already labeled as “Fraud” or “Regular Claim.” In contrast, labeling is not necessarily required for unsupervised learning models. In this case, the algorithms assume that most of the claims are not fraudulent and learn to inherit structure from the normal data. The anomalies/outliers,

Table 1 Numeric attributes (© ifb SE)

months_as_customer	age	policy_bind_date	policy_deductable	policy_annual_premium
umbrella_limit	capital.gains	capital.loss	incident_date	incident_hour_of_the_day
number_of_vehicles_involved	bodily_injuries	total_claim_amount	injury_claim	property_claim
vehicle_claim	auto_year			

Table 2 Categorical attributes (© ifb SE)

policy_number	policy_state	policy_csl	insured_zip	insured_sex
insured_education_level	insured_occupation	insured_hobbies	insured_relationship	incident_type
collision_type	incident_severity	authorities_contacted	incident_state	incident_city
incident_location	property_damage	witnesses	police_report_avaliable	auto_make
auto_model				

which are suspicious by differing significantly from the normal structure, are identified as anomalous. In the following sections, we will provide experiment results for both supervised and unsupervised learning models and compare them by listing the pros and cons in Table 5. In addition, we suggest an advanced ensemble model, which combines the supervised and unsupervised learning in section Fehler! Verweisquelle konnte nicht gefunden werden. As a result, we verify that the model performance is getting better, where AUC (area under the curve), precision, and recall are selected as evaluation indexes. For more details about model evaluation, we refer the reader to (Liermann et al., Mathematical Background of Machine Learning 2019).

3.2.1 Supervised Model

For the supervised model, we chose a random forest where the number of trees is set to ten and the maximum depth of trees is set to ten. The cross-validation performance result is given as follows, where 10 folds are used (Table 3).

We extract the variable importance of the above random forests. Recalling method article, variable importance shows the contribution of a feature for making accurate prediction on label. The more a model relies on a feature to make predictions, the more important it is for the model. In the following figure, the first ten significant features are illustrated (Fig. 2).

Reading from the Fig. 2: Variable importance, the features "incident_severity" and "insured_hobbies" have the most significant influence on the prediction of fraud. We are also able to verify this by illustrating "insured_hobbies" against "Fraud." As shown in Fig. 3: Insured_hobbies vs. label, the insured parties who like playing chess and watching movies have a much higher probability of making fraudulent claims. As a comparison, we illustrate the "insured_education_level" against "Fraud" in Fig. 4: Insured_education_level vs. label. It is not hard to see that education level has nearly no influence on the claim being fraudulent or not.

Table 3 Random forests: performance (© ifb SE)

AUC (area under the curve)	Precision	Recall
0.8439	0.6242	0.7950

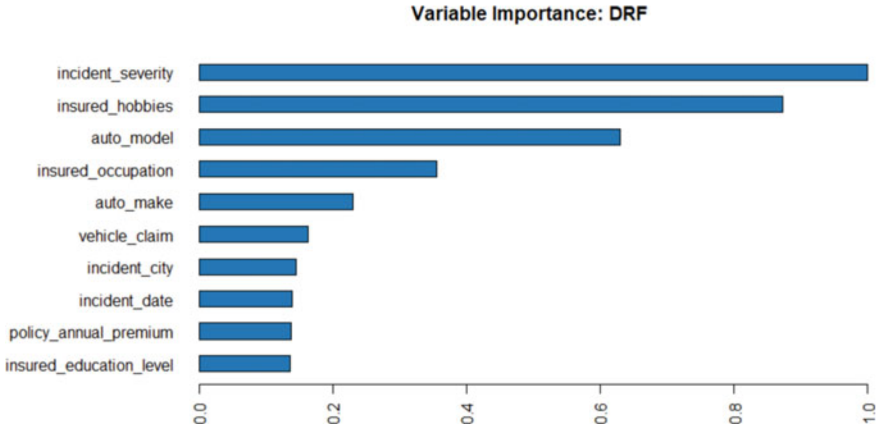


Fig. 2 Variable importance (© ifb SE)

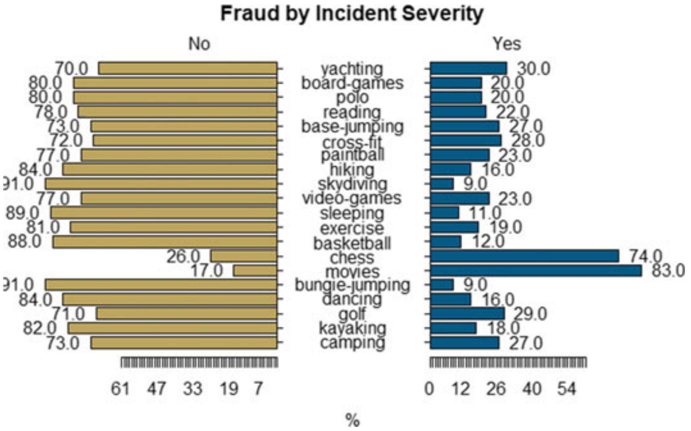


Fig. 3 Insured_hobbies vs. label (© ifb SE)

3.2.2 Unsupervised Model

Similarly to (Liermann et al., Batch processing—Pattern recognition 2019), we use autoencoders, which are classic artificial deep networks and capable of performing unsupervised learning tasks. They are targeted by learning an efficient representation/reconstruction of the input data within the network structure. The reconstruction error is used to quantify the probability of an anomalous sample, in other words, a large reconstruction error value indicates an anomaly. For more details about autoencoders, we refer the reader to (Liermann et al., Batch processing—Pattern recognition 2019).

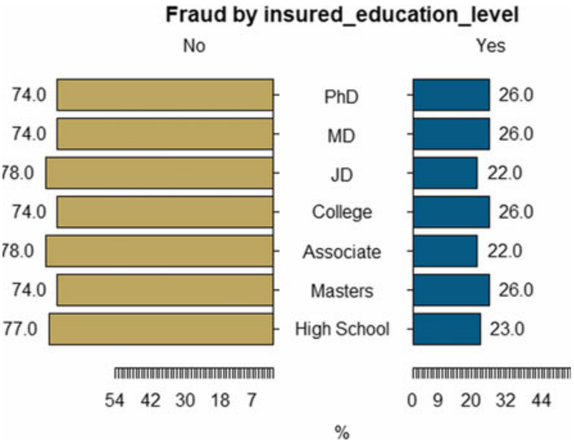


Fig. 4 Insured_education_level vs. label (© ifb SE)

Table 4 Autoencoder: performance (© ifb SE)

AUC (area under the curve)	Precision	Recall
0.6508	0.4424	0.7449

We chose an autoencoder with three hidden layers and 20, 30, 20 hidden nodes, separately, where the Tanh function is set as an activation function. The ten-cross-validation result is given as follows (Table 4).

We illustrate the reconstruction error against label in Fig. 5: Reconstruction error vs. label. It is not hard to see a sharp separation between normal and fraudulent samples at a reconstruction error of around 0.065.

In comparison to supervised learning models, autoencoders target identifying the anomaly according to the reconstruction error. Some normal samples (the points on the top right-hand side of Fig. 5) are suspicious as an anomaly, but incorrectly identified as fraudulent samples (false positive). This leads to lower precision. We summarize the pros and cons of both learning models in the following table (Table 5).

3.2.3 Ensemble Model

In the last part of this section, we provide an algorithm which combines both supervised and unsupervised learning approaches. The intuition is illustrated in the following figure, where the reconstruction error of the unsupervised method is taken as an additional feature for random forests (Fig. 6).

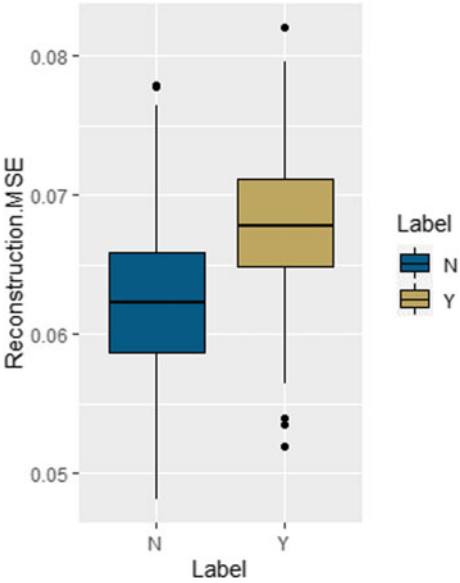


Fig. 5 Reconstruction error vs. label (© ifb SE)

Table 5 Pros and cons of supervised and unsupervised learning (© ifb SE)

Supervised learning	Unsupervised learning
• Labeling necessary	– Labeling not necessary
– Good explainability (variable importance)	• Black box
– Efficient detection of existing patterns in the past	• Risk of high false negative (anomaly ≠ fraud)
• Not sensible for new fraud	– Able to recognize new anomaly/fraud pattern

By adding this additional feature, we can improve the model learning performance by extracting a further finer correlation between anomaly and fraud. For example, which anomalous samples have a higher probability of being fraudulent and which are just normally anomalous. The ten-cross-validation result is presented in Table 6.

Recalling Tables 3, 4, the validation performance is significantly better in comparison to each single approach. The AUC (area under the curve) increases from 84 to 88.5%, which shows that the ensemble models can learn more sufficiently. In addition, we verify this by illustrating the variable importance of ensemble models. Not surprisingly, the reconstruction error (Reconstruction.MSE) has taken Rank 4, which indicates that the influence of anomaly on the prediction of fraud should not be ignored (Fig. 7).

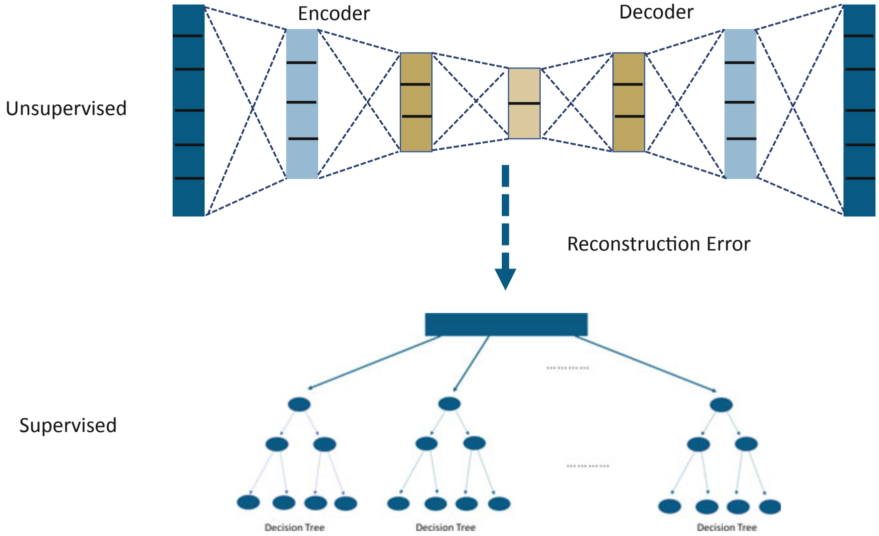


Fig. 6 Ensemble model (© ifb SE)

Table 6 Combined model: performance (© ifb SE)

AUC (area under the curve)	Precision	Recall
0.8858	0.6915	0.8787

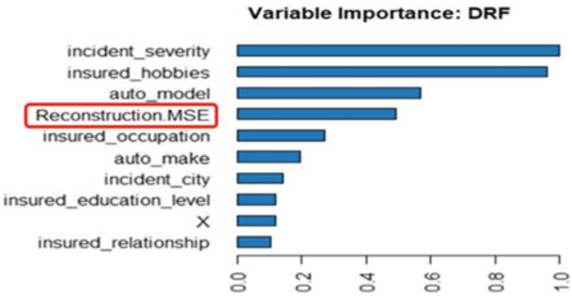


Fig. 7 Variable importance (© ifb SE)

3.3 Model Calibration

When carrying out the model calibration, the business owner first must think about the goals of the new model. In our approach like in many statistical models, there is a natural inverse relationship between the rate of fraud cases the model can predict correctly (recall) and the efficiency with which the prediction can take place (precision). Therefore, it is wise to think about these numbers in the very early stage of a fraud detection project, also because the calibration can have a drastic effect on the process design in claims handling and the model must be accepted by the claims department. A more efficient model needs more care in checking the stability of the unpredicted fraud cases (random sampling), while a model focused more on the fraud detection rate needs more process steps to weed out the false positives (Fig. 8).

The calibration of the model also has an effect on the business case, which should be made before the project is even budgeted. A model focused on the fraud detection rate will have more savings from avoiding paying wrongful claims while a more efficient process frees up more employees for other tasks. The model is calibrated by changing the threshold from which a particular claim is classified as a fraud case. This threshold is set on the fraud probability, which is an interim result in the classification model. This will be elaborated more in 4.Process Design. The model calibration can then be read and is set like this.

"A claim is classified as a fraud case if the estimated fraud probability exceeds 95%" means the threshold is set at 95%

"The model correctly predicts 93% of all fraud cases" means recall of 93%

"For every 4 true fraud cases a claims handler will have to look into 1 falsely predicted claim" means the efficiency (precision) of the model is 80%. For

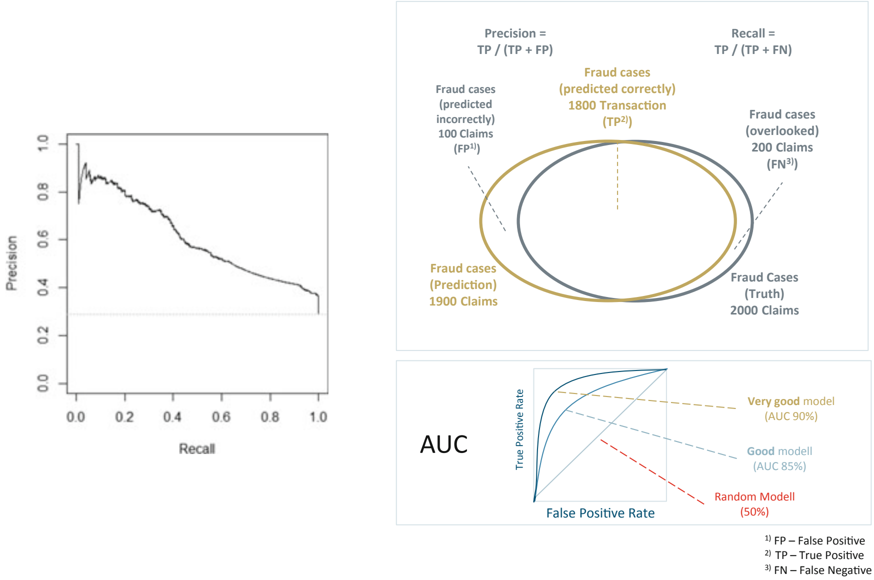


Fig. 8 Calibration statistics (© ifb SE)

more details on recall, precision, and model calibration in machine learning models see (Hartung and Führer 2021).

4 Process Design

A key insight into adopting machine learning can be drawn from experience in the medical field: when doctors have the right amount of data and sufficient time, they are currently still able to keep up with success rates of AI-based solutions. HBR cites a study in which 99% of treatments suggested by the doctors were also suggested by the AI solution. But the assumptions from lab tests are usually not fulfilled in reality. Doctors are not always able to request all diagnostics and do not have the time to analyze everything in detail. Therefore, in reality, success rates are often lower. This shows the potential for fraud detection: a synergetic approach to AI selection and expert approval. This will also lead to a much more satisfying approach in claims handling as the responsible parties will mostly look at true frauds. The study also indicates that AI-based solutions are on the verge of beating humans at diagnostics as they are just able to consume much more information. In this case, the AI solution merely suggested treatments based on papers the doctors had not read yet. The same is possible for fraud detection: the engine will be able to take into account all available information and if implemented correctly even tells the fraud expert how it came to the conclusion of a fraud indication. Now we will show how to set up an AI-based process.

The fraud detection process usually designed follows the process steps detection, investigation, prevention. The task of setting up an efficient machine learning-based fraud detection process can be divided into two pillars: (I) An ML classification engine that categorizes new claims into risk buckets based on fraud patterns in the past (II) A process that searches for new fraud patterns in the "no fraud bucket" and identifies false positives. In addition to this, a feedback loop is required to feed the results into the model training process (Fig. 9).

Let us first look into the fraud classification engine. This engine will be based on the methodology described in Part III. The resulting estimate of the machine learning algorithm provides a probability for each claim that, given the history of detected frauds, it belongs to a particular fraud cluster. Using this, claims will be categorized into risk buckets. The number of buckets depends on client-specific needs and affects the granularity of the process. The minimum requirement is a two-bucket process with a fraud and no-fraud bucket for claims. This can easily be extended to a traffic light (three buckets)

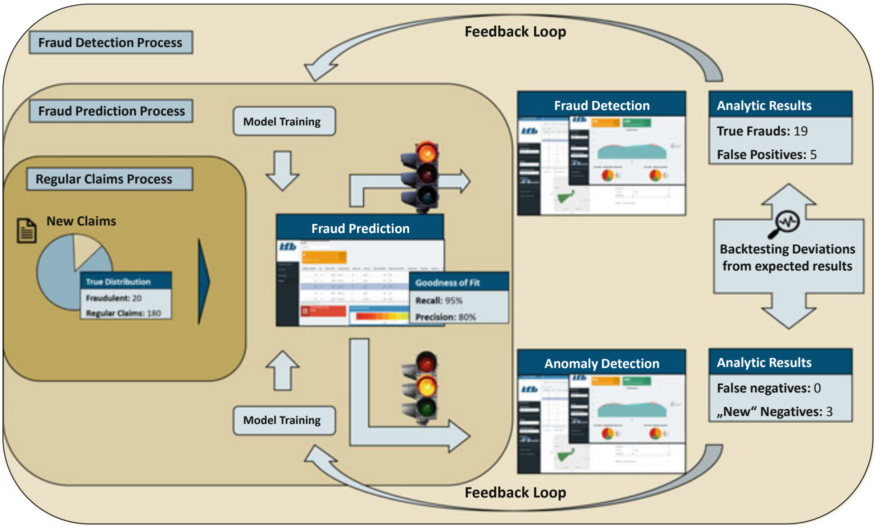


Fig. 9 Target operating model fraud (© ifb SE)

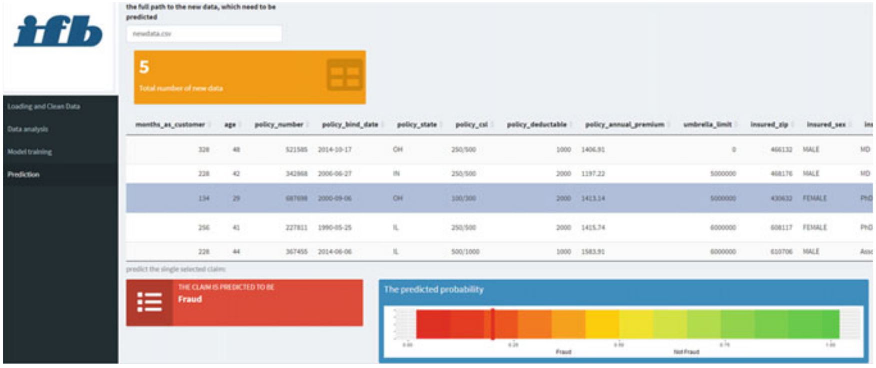
approach. In both cases, the classification is based on thresholds/boundaries for the estimated fraud probabilities. A traffic light approach could use the following boundaries for the classification: Green light: Fraud probability < 70%; Yellow light: Fraud probability 70–90%; Red light: Fraud probability > 90%. The last category should be calibrated such that it remains feasible to check every claim in that bucket. In particular, the threshold for the highest risk bucket will determine the quality of the classification. The lower the threshold, the higher the share of true fraud detected by the algorithm. This is indicated by the derived recall measure—when recall is at 95% this means that 95% of all true frauds will be in the highest risk bucket. The threshold also has a second implication, which is precision. The precision of the process will increase with the level of the threshold, which means that the higher the fraud probability threshold is set, the higher the share of true frauds in the fraud classification bucket, but also the higher the number of missed fraud cases in the lower risk buckets. Therefore, insurance companies need to take into account the share of true frauds they want to have in the highest risk bucket when redesigning the process. In addition, the middle-risk bucket(s) can be set up with a leaner process or a random sampling approach to increase the overall recall and retain a high precision in the top bucket. It might be that recall of the highest risk bucket is 90%, but the overall recall of the two combined risk categories is 98%. Identifying these eight additional percentage points will be much tougher though due to the lower precision and the resulting (far) lower proportion of true frauds in the analyzed claims. The process design is therefore key when deciding on a multi-bucket approach.

Now let us look into the second pillar. When setting up processes as mentioned above, two things need to be clear and cannot be forgotten. Firstly, the algorithm described in section three only relies on past observation. It will inherently be biased toward the past. Therefore, it is crucial to always analyze your lower risk buckets for new fraud patterns. Secondly, the minute an insurer goes live with an ML-based classification system, the algorithm will only be trained with the true frauds detected through the clustering. This will to some degree be countered by the analysis of the lower risk categories mentioned above. Nevertheless, it is recommended to include a sample in the process that is analyzed in an expert-based process. Feeding this data into the algorithm increases the quality of the classification and combats the risk of historical bias.

5 Implementing an AI-Based Fraud Detection System

Most insurance companies will have a claims management system in place. Also. data will probably be extracted to carry out fraud analysis. To implement an ML-based KI classification, an open-source solution is strongly recommended. Insurance companies tend to have more experience with R, so the displayed implementation shows a setup based on an R architecture. Nevertheless, the same can be done using Python. For the target architecture, an interface of the claims management system to R should be established. This can easily be established using microservices or other open-source solutions to keep costs low. The models should be pretrained in the implementation of the architecture and updated regularly, but not too often to keep the process efficient. Once a month is a recommended frequency, but this depends on the inflow of new claims. ifb is using the R package shiny as a user interface (UI). The UI usually has two components: one for the data science team to train and calibrate the model and one for the claims handling department. The second dashboard will be the main supporting pillar for the fraud detection team.

A dashboard should also offer other supportive information to help the claims handler choose the best next step, e.g., which information to ask for, such as which features were most important in classifying the particular claim as fraudulent (Fig. 10).



the full path to the new data, which need to be predicted

newdata.csv

5
Total number of new data

months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	insured_sex	ins
328	48	521585	2014-10-17	OH	250/500	1000	1406.91	0	466132	MALE	MD
228	42	342868	2006-06-27	IN	250/500	2000	1197.22	5000000	468176	MALE	MD
134	29	687698	2000-09-06	OH	100/300	2000	1413.14	5000000	430632	FEMALE	PhD
256	41	227811	1990-05-25	IL	250/500	2000	1415.74	6000000	608117	FEMALE	PhD
228	44	367455	2014-06-06	IL	500/1000	1000	1583.91	6000000	610706	MALE	Ass

predict the single selected claim:

THE CLAIM IS PREDICTED TO BE
Fraud

The predicted probability

Fig. 10 Example setup of fraud detection dashboard (© ifb SE)

One challenge in implementation can be the maintenance of an open-source architecture. As it is not standard software, there is no vendor with maintenance responsibility. On the other hand, there are no license fees and millions of users worldwide are carrying out maintenance by developing new content and ensuring the compatibility of different packages with one another. The second big challenge is data cleaning. For the algorithm to work well, features need to be defined in a consistent format. Also including new data sources needs to be considered in PoCs in the implementation phase (Hartung and Führer 2021).

6 Summary

All in all, it is clear that machine learning is the future of fraud detection for insurance companies. The transition point is just a matter of data availability, budgets, and tackling the resistance of, for example, the claims department to changing current processes. When it comes to the implementation, data cleaning and introducing new external data sources into the process are key to get the results in terms of recall and precision that are needed to set up an efficient process. Finally, it is important to start small but think big. Start the project with PoCs and prototypes, but when the decision for implementation of the productive process is made, think about whether to use a common data store for the ML architecture. Every AI solution will benefit from large data availability, so do not restrict yourself with short-sighted architecture decisions. Open-source microservices can help a lot with setting this up without encountering the past pitfalls of failed central data platform projects.

Literature

Hartung, Sören and Manuela Führer. 2021. "AI for Impairment Accounting." In *The Digital Journey of Banking and Insurance, Volume I—Disruption and DNA*, edited by Volker Liermann and Claus Stegmann. New York: Palgrave Macmillan.

H2O.ai. 2019. *h2o.ai Overview*, January 29. Accessed September 29, 2020. http://docs.h2o.ai/h2o/latest-stable/h2o-docs/index.html.

insurance europe. 2019. *Insurance Fraud: Not a Victimless Crime.* Brussels: Insurance Europe aisbl, November.

Kaggle. 2018. *Auto Insurance Claims* Data, August 20. Accessed September 27, 2020. https://www.kaggle.com/buntyshah/auto-insurance-claims-data.

Liermann, Volker, Sangmeng Li, and Norbert Schaudinnus. 2019. "Batch Processing—Pattern Recognition." In *The Impact of Digital Transformation and Fintech on the Finance Professional*, edited by Volker Liermann and Claus Stegmann. New York: Palgrave Macmillan.

Liermann, Volker, Sangmeng Li, and Norbert Schaudinnus. 2019. "Mathematical Background of Machine Learning." In *The Impact of Digital Transformation and Fintech on the Finance Professional*, edited by Volker Liermann and Claus Stegmann. New York: Palgrave Macmillan.

McKinsey & Company. 2015. *Claims Management: Taking a Determined Stand Against Insurance Fraud*. Germany: Munich.

Velauthapillai, Jeyakrishna and Johannes Floß. 2021. "Special Data for Insurance Companies." In *The Digital Journey of Banking and Insurance, Volume III—Data Storage, Processing, and Analysis*, edited by Volker Liermann and Claus Stegmann. New York: Palgrave Macmillan.