# Report

## Einleitung und Motivation

Durch die Digitalisierung der Versicherungsbranche wurde eine neue Norm gesetzt, die es ermöglichte besser auf die Bedürfnisse der Kunden einzugehen. Die Versicherungen stellten sich damit einer bereits bekannten Herausforderung, dem Versicherungsbetrug. Jede zehnte Schadensmeldung ist betrugsverdächtig und kostet den deutschen Versicherungen rund fünf Milliarden Euro pro Jahr (GDV, 2020). Um Versicherungsbetrug zu erkennen, wird spezielle Betrugserkennungssoftware verwendet. Hierbei nutzen Versicherungsbetrüger zunehmend Anleitungen aus dem Internet, die es erleichtern die Betrugserkennungssoftware zu umgehen (GDV, 2020).

*Quelle: https://www.gdv.de/gdv/medien/medieninformationen/sorge-der-versicherer-corona-gibt-betruegern-auftrieb-61842*

Herkunft der Daten: Quelle: https://www.kaggle.com/datasets/girishvutukuri/insurance-fraud Die Daten stammen aus Kaggle und wurden von einem User zur Verfügung gestellt. Sie enthalten Informationen zu Betrugsfällen innerhalb eines Versicherungsunternehmens. Die genaue Datenerfassung, wurde vom User nicht beschrieben. Die Daten wurden auf GitHub abgespeichert, falls der User sich entscheidet die Daten zu löschen, sind diese weiterhin verfügbar.

Mit diesem Datensatz untersuchen wir, ob und welche Maßnahmen nötig sind, um den Prozess der Erkennung von Versicherungsbetrug zu verbessern.

### Forschungsfrage

Welche Faktoren sind für die Erkennung von Betrugsfällen relevant?

Wir haben folgende Hypothesen:

1. Unter 35 jährige Versicherungskunden betrügen öfters.-> Relevante Variablen: InsuredAge, ReportedFraud
2. Männer betrügen häufiger als Frauen.-> Relevante Variablen: InsuredGender, ReportedFraud
3. Schadensfälle ohne polizeilicher Dokumentation sind häufiger Betrugsfälle.-> Relevante Variablen: AuthoritiesContacted, ReportedFraud

### Data Dictionary

Beschreibung der relevanten Variablen

* InsuredAge: Das Alter des Kunden zum Zeitpunkt des Anspruchs. (Datentyp: Integer)
* InsuredGender: Das Geschlecht des Kunden. (Datentyp: String)
* AuthorotiesContacted: Information darüber, ob Behörden (Polizei, Krankenwagen, Feuerwehr oder andere) kontaktiert wurden. (Datentyp: String)
* ReportedFraud: Information darüber, ob der Anspruch als Betrug gemeldet wurde oder nicht. (Datentyp: String)

| Name | Type | Format |
| --- | --- | --- |
| AuthoritiesContacted | nominal | object |
| InsuredAge | numerical | int |
| InsuredGender | nominal | object |
| ReportedFraud | nominal | object |

## Setup

import pandas as pd  
import altair as alt  
import numpy as np  
import warnings  
  
warnings.simplefilter(action='ignore', category=FutureWarning)  
alt.data\_transformers.disable\_max\_rows()

DataTransformerRegistry.enable('default')

## Daten

### Importieren der Daten

ROOT= "https://raw.githubusercontent.com/christophersegatz/dst-project/main/data/insurance\_fraud/"  
DATA = "fraud.csv"  
  
df=pd.read\_csv(ROOT + DATA)

### Daten Anpassungen

# Nicht benötigte Tabellen aus dem Dataframe entfernen  
df.drop(["DateOfIncident","TypeOfCollission", "SeverityOfIncident", "IncidentState", "IncidentCity", "IncidentAddress", "IncidentTime",  
 "NumberOfVehicles", "PropertyDamage", "BodilyInjuries", "Witnesses", "PoliceReport", "AmountOfTotalClaim", "AmountOfInjuryClaim",  
 "AmountOfPropertyClaim", "AmountOfVehicleDamage", "InsuredZipCode", "InsuredOccupation", "InsuredHobbies",  
 "CapitalGains", "CapitalLoss", "Country", "InsurancePolicyNumber", "DateOfPolicyCoverage", "InsurancePolicyState", "Policy\_CombinedSingleLimit",  
 "Policy\_Deductible", "PolicyAnnualPremium", "UmbrellaLimit", "InsuredRelationship", "CustomerID", "TypeOfIncident",  
 "InsuredEducationLevel", "CustomerLoyaltyPeriod"], axis= 1, inplace=True)  
  
# Entfernt alle Zeilen, die mindestens einen Null-Wert haben  
df.dropna()

|  | AuthoritiesContacted | InsuredAge | InsuredGender | ReportedFraud |
| --- | --- | --- | --- | --- |
| 0 | Police | 35 | MALE | N |
| 1 | Police | 36 | MALE | N |
| 2 | Other | 33 | MALE | N |
| 3 | Other | 36 | MALE | N |
| 4 | Fire | 29 | FEMALE | N |
| ... | ... | ... | ... | ... |
| 28831 | Police | 46 | MALE | N |
| 28832 | Fire | 44 | MALE | N |
| 28833 | Fire | 53 | MALE | N |
| 28834 | Ambulance | 53 | MALE | N |
| 28835 | Other | 36 | FEMALE | N |

# Als kategoriale Variablen abändern  
df = df.astype(  
 {"AuthoritiesContacted": "category","InsuredGender": "category",})

## Analyse

### Deskriptive Analyse

df.head()

|  | AuthoritiesContacted | InsuredAge | InsuredGender | ReportedFraud |
| --- | --- | --- | --- | --- |
| 0 | Police | 35 | MALE | N |
| 1 | Police | 36 | MALE | N |
| 2 | Other | 33 | MALE | N |
| 3 | Other | 36 | MALE | N |
| 4 | Fire | 29 | FEMALE | N |

df.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 28836 entries, 0 to 28835  
Data columns (total 4 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 AuthoritiesContacted 28836 non-null category  
 1 InsuredAge 28836 non-null int64   
 2 InsuredGender 28806 non-null category  
 3 ReportedFraud 28836 non-null object   
dtypes: category(2), int64(1), object(1)  
memory usage: 507.3+ KB

mean\_insured\_age = df['InsuredAge'].mean()  
mean\_insured\_age = round(mean\_insured\_age, 0)  
print("Durchschnittliches Alter der Versicherten: {:.0f}".format(mean\_insured\_age))  
  
# Boxplot-Diagramm  
box = alt.Chart(df).mark\_boxplot().encode(x=alt.X('InsuredAge:Q',axis=alt.Axis(title="")))  
  
hist = alt.Chart(df).mark\_bar().encode(alt.X("InsuredAge", bin=alt.Bin(maxbins=30), title="Alter"),  
 y=alt.Y('count()', axis=alt.Axis(title="Anzahl der Versicherten")),)  
  
hist | box

Durchschnittliches Alter der Versicherten: 39

alt.HConcatChart(...)

Erkenntnis aus den Graphen:

* Das maximale Alter der Versicherten beträgt 64 und das minimalste 19.
* Der dritte Quartil (Q3) des Alters liegt bei 44, was bedeutet, dass 75% der Versicherten jünger als 44 Jahre alt sind.
* Der erste Quartil (Q1) des Alters liegt bei 33, was bedeutet, dass 25% der Versicherten jünger als 33 Jahre alt sind.
* Der Median des Alters beträgt 38, was bedeutet, dass 50% der Versicherten jünger und 50% älter als 38 Jahre alt sind.

# Fehlende werte entfernen  
df\_gender = df.dropna(subset=['InsuredGender'])  
  
bar\_gender = alt.Chart(df\_gender).mark\_bar(size=40).encode(  
 alt.X("InsuredGender:N", axis=alt.Axis(title=""), sort=['F', 'M']),  
 alt.Y("count()", axis=alt.Axis(title="")), color=alt.Color('InsuredGender:N', legend=alt.Legend(title="Geschlecht"))  
)  
text\_gender = alt.Chart(df\_gender).mark\_text(  
 align='center', baseline='middle', fontWeight='bold', dy=-10  
).encode(x=alt.X('InsuredGender:N', axis=alt.Axis(title="")), y=alt.Y('count()', axis=alt.Axis(title='Anzahl Versicherter')),  
text=alt.Text('count()'),)  
gender\_layer = bar\_gender+text\_gender  
gender\_layer.configure\_axis(labelAngle=0).properties(width=400, height=200, title="Anzahl Versicherter nach Geschlecht"  
).configure\_legend(orient='right', strokeWidth=1, padding=10, strokeColor='grey')

alt.LayerChart(...)

Die Anzahl der weiblichen (15644) ist höher als die, der männlichen Versicherten (13162).

# Barchart  
bar\_auth = alt.Chart(df).mark\_bar().encode(alt.X('AuthoritiesContacted:N', axis=alt.Axis(title=''),),  
 alt.Y('count()', axis=alt.Axis(title='')),alt.Color('AuthoritiesContacted:N', legend=alt.Legend(title='Kontaktierte Behörden')),  
)  
text\_auth = alt.Chart(df\_gender).mark\_text(align='center', baseline='middle',fontWeight='bold',dy=-10  
).encode(x=alt.X('AuthoritiesContacted:N', axis=alt.Axis(title=''),),y=alt.Y('count()', axis=alt.Axis(title='Anzahl')), text=alt.Text('count()'))  
grouped\_auth=bar\_auth+text\_auth  
  
grouped\_auth.properties(title='Relative Häufigkeit von kontaktierten Behörden', height=200, width=400  
).configure\_legend(orient='right',strokeWidth=1,padding=10,strokeColor='grey').configure\_axis(labelAngle=0)

alt.LayerChart(...)

Bei den meisten Fällen wurde die Polizei (8313) kontaktiert. Ebenso gab es viele Fälle mit Kontakt zur Feuerwehr (6513) und Krankenwagen(5726). Bei 5564 Fällen wurden andere kontakiert und bei 2690 Fällen gar keiner.

bar\_fraud = alt.Chart(df).mark\_bar(size=40).encode(alt.X('ReportedFraud:N', axis=alt.Axis(title=''),),  
 alt.Y('count()', axis=alt.Axis(title='')), alt.Color('ReportedFraud:N', legend=alt.Legend(title='Gemeldeter Betrug')),)  
  
text\_fraud = alt.Chart(df\_gender).mark\_text(align='center',baseline='middle',fontWeight='bold',dy=-10  
).encode(x=alt.X('ReportedFraud:N', axis=alt.Axis(title=''),),y=alt.Y('count()', axis=alt.Axis(title='Anzahl')),text=alt.Text('count()'))  
group\_fraud = bar\_fraud + text\_fraud  
  
group\_fraud.properties(title='Relative Häufigkeit von gemeldeten Betrugsfällen',height=200,width=400  
).configure\_legend(orient='top-right',strokeWidth=1,padding=10,strokeColor='grey').configure\_axis(labelAngle=0)

alt.LayerChart(...)

Von 28806 Fällen sind 7780 als Betrug gemeldet worden. 21026 Fälle wurden nicht als Betrug gemeldet.

### Explorative Analyse

H1: Unter 35 jährige Versicherungskunden betrügen öfters. Diese Hypothese wurde gewählt, um zu untersuchen, ob die Altersgruppe der Versicherungskunden einen Einfluss auf die Betrugsrate hat.

# Erstelle eine neue Spalte is\_over\_35, die die Versicherten ab 35 Jahre kennzeichnet  
df['is\_over\_35'] = df['InsuredAge'] >= 35  
pd.crosstab(index=df['is\_over\_35'], columns=df['ReportedFraud'],normalize='index').round(4) \* 100

| ReportedFraud | N | Y |
| --- | --- | --- |
| is\_over\_35 |  |  |
| False | 72.4 | 27.6 |
| True | 73.3 | 26.7 |

Die Betrugsrate bei Kunden unter 35 Jahren ist nur um 0.9 Prozentpunkte höher als bei Kunden über 35 Jahren. Daher ist die Hypothese nur bedingt bestätigt.

H2: Männer betrügen häufiger als Frauen. Diese Hypothese wurde gewählt, da Statistiken zeigen das Männer häufiger Straftaten begehen als Frauen. Quelle: Statistisches Bundesamt. (29. November, 2022). Anzahl der rechtskräftig verurteilten Personen in Deutschland nach Geschlecht von 2011 bis 2021 [Graph]. In Statista. Zugriff am 01. Februar 2023, von https://de.statista.com/statistik/daten/studie/1068769/umfrage/rechtskraeftig-verurteilte-personen-in-deutschland-nach-geschlecht/ Es soll daher untersucht werden ob mehr Männer einen Versicherungsbetrug versuchen.

pd.crosstab(index=df['InsuredGender'],columns=df['ReportedFraud'],normalize='index').round(4) \* 100

| ReportedFraud | N | Y |
| --- | --- | --- |
| InsuredGender |  |  |
| FEMALE | 74.03 | 25.97 |
| MALE | 71.75 | 28.25 |

Bei den Männern betrügen 28.25% - bei den Frauen 25.97%. Die Hypothese ist somit bestätigt.

H3: Schadensfälle ohne polizeilicher Dokumentation sind häufiger Betrugsfälle. Diese Hypothese wurde gewählt da davon ausgegangen wird das bei einem Betrugsversuch die Polizei nicht kontaktiert wird um kein zusätzliches Beweismaterial zu hinterlassen.

pd.crosstab(index=df['AuthoritiesContacted'], columns=df['ReportedFraud'],normalize='index').round(4) \* 100

| ReportedFraud | N | Y |
| --- | --- | --- |
| AuthoritiesContacted |  |  |
| Ambulance | 69.28 | 30.72 |
| Fire | 70.60 | 29.40 |
| None | 90.12 | 9.88 |
| Other | 67.58 | 32.42 |
| Police | 75.54 | 24.46 |

Wurde keine Authority kontaktiert ist die Betrugsrtate mit 9.88% am geringsten. Wurde die Polizei kontaktiert ist die Betrugsrate mit 24.46% am 2. höchsten.

## Visualisierungen

#### H1: Unter 35 jährige Versicherungskunden betrügen öfters.

df['is\_over\_35'] = df['InsuredAge'] >= 35 # Erstelle neuer Spalte is\_over\_35, die die Versicherten ab 35 Jahre kennzeichnet  
df['is\_over\_35'] = df['is\_over\_35'].astype(str) #Umwandlung in Stringtyp  
df['is\_over\_35'] = df['is\_over\_35'].replace({"True": "Über 35", "False": "Unter 35"}) #True, False mit Über 35 und Unter 35 ersetzen  
  
bar = alt.Chart(df).mark\_bar(size=50).encode(x=alt.X("is\_over\_35:N", axis=alt.Axis(title="Altersgruppe"),),  
 y=alt.Y("count()", scale=alt.Scale(domain=[0, 21000]),axis=alt.Axis(title="Anzahl der Meldungen")),  
 color=alt.Color("ReportedFraud:N", legend=alt.Legend(title="Gemeldeter Betrug")),)  
  
text\_age = alt.Chart(df).mark\_text(align='center',baseline='middle',fontWeight='bold',dy=-10 #Text für Werte an Balken  
).encode(x=alt.X('is\_over\_35:N', axis=alt.Axis(title="")),y=alt.Y('count()', axis=alt.Axis(title='')),text=alt.Text('count()'),)  
  
grouped\_bar=bar+text\_age #Kombinieren von bar+text\_age in grouped\_bar  
#Anpassungen am chart  
grouped\_bar.properties(width=400,height=200,title="Anzahl gemeldeter Betrugsfälle nach Altersgruppe"  
).configure\_axis(labelAngle=0).configure\_legend(orient='top-left',strokeWidth=1,padding=10,strokeColor='grey').configure\_view(strokeWidth=2)

alt.LayerChart(...)

Für diese Hypothese ist ein gestapeltes Balkendiagramm am besten geeignet, da es sich hierbei um 2 Kategoriale Variablen “InsuredAge” bzw. “is\_over\_35” und “ReportedFraud” handelt, im Bezug zur einer Messgröße, was in diesem Fall die Anzahl der Fälle darstellt. Andere Charttypen wären hier nicht sehr geeignet, da diese die Verteilung der Daten zwischen den Altersgruppen nicht gut erkennbar darstellen.

Bei der Erstellung der Visualisierung wurde eine neue Spalte “is\_over\_35” erstellt, die die Versicherten ab 35 Jahren kennzeichnet. Diese Spalte wurde aus der Spalte “InsuredAge” abgeleitet, indem überprüft wurde, ob sie größer oder gleich 35 ist. Die Spalte “is\_over\_35” wurde dann in einen String umgewandelt und in die Bezeichnungen “Über 35” oder “Unter 35” umbenannt, um die X-Achsenbeschriftung zu erleichtern.

Damit wurde anschließend mit Altair ein gruppiertes Balkendiagramm erstellt. Um eine bessere Übersicht über die Daten zu bieten, wurden die Werte auf den Balken dargestellt. Anschließend wurden Einstellungen vorgenommen, die unteranderem die Legende, Schriftgröße und -farbe uvm. anpassen.

#### H2: Männer betrügen häufiger als Frauen.

# Prozentzahlen berechnen  
grouped = df.groupby(['InsuredGender', 'ReportedFraud']).size().reset\_index(name='counts')  
total\_counts = grouped.groupby('InsuredGender').sum().reset\_index()  
grouped = pd.merge(grouped, total\_counts, on='InsuredGender', how='left')#left join grouped + total\_counts  
grouped['Prozent'] = (grouped['counts\_x'] / grouped['counts\_y']) \* 100 #Auf Prozent berechnen  
grouped['Prozent'] = grouped['Prozent'].round(0) #Auf Ganzzahl abrunden   
  
chart = alt.Chart(grouped).mark\_bar(size=50).encode(x=alt.X('InsuredGender:N'), #barchart erstellen  
 y=alt.Y('Prozent:Q'),color=alt.Color('ReportedFraud:N', legend=alt.Legend(title="Gemeldete Betrugsfälle")))  
  
text = alt.Chart(grouped).mark\_text(align='center',baseline='bottom',fontWeight='bold',dy=+15 #Text für Prozentzahlen  
).encode(x=alt.X('InsuredGender:N', axis=alt.Axis(title="Geschlecht")),y=alt.Y('Prozent:Q'),text=alt.Text('Prozent:Q'))  
  
layered\_gender = chart + text #charts kombinieren  
# Anpassungen am Chart durchführen  
layered\_gender.configure\_axis(labelAngle=0).properties(height=200,width=400,title="Betrugsfälle nach Geschlecht"  
).configure\_legend(orient='right',strokeWidth=1,padding=10,strokeColor='grey').configure\_view(strokeWidth=2)

alt.LayerChart(...)

Die Entscheidung für diese Visualisierung ist wie bei H1 schon erwähnt, die einzige sinnvolle Möglichkeit, um die Verteilung der Daten gut darzustellen. Hierbei wurden die Variablen “ReportedFraud” und “InsuredGender” gewählt. Hierbei wird verglichen, wie viel Prozent von Männern bzw. Frauen, betrogen oder nicht betrogen haben.

Mit der group-by Methode wird die Anzahl der gemeldeten Betrugsfälle nach Geschlecht und gemeldetem Betrug berechnet. Daraufhin wird die Gesamtzahl aller gemeldeten Betrugsfälle nach Geschlecht berechnet. Anschließend werden die beiden miteinander verbunden, um die prozentuale Häufigkeit der gemeldeten Betrugsfälle zu berechnen. Dafür wurden die Daten von grouped nochmal umgewandelt und anschließend abgerundet auf, sodass diese ohne Nachkommastellen angezeigt werden.

Es wird ein gestapeltes Balkendiagramm wie in H1 verwendet, mit den Prozentangaben innerhalb der Balken. Auch hier werden dieselben Einstellungen durchgeführt wie bei H1.

#### H3: Schadensfälle ohne polizeilicher Dokumentation sind häufiger Betrugsfälle.

grouped = df.groupby(['AuthoritiesContacted', 'ReportedFraud']).size().reset\_index(name='counts') #gruppieren in neue tabelle  
total\_counts = grouped.groupby('AuthoritiesContacted').sum().reset\_index() #gruppieren in tabelle total\_counts   
grouped = pd.merge(grouped, total\_counts, on='AuthoritiesContacted', how='left') #mergen mit left join  
grouped['Prozent'] = (grouped['counts\_x'] / grouped['counts\_y']) \* 100 #in prozent umwandeln  
grouped['Prozent'] = grouped['Prozent'].round(0) #auf Ganzzahl abrunden  
  
  
chart = alt.Chart(grouped).mark\_bar(size=50).encode(x=alt.X('AuthoritiesContacted:N'), #bar chart erstellen  
 y=alt.Y('Prozent:Q'),color=alt.Color('ReportedFraud:N', legend=alt.Legend(title="Gemeldete Betrugsfälle")))  
  
text = alt.Chart(grouped).mark\_text(align='center',baseline='middle',fontWeight='bold',dy=+10 #Text für Prozentzahlen in den Balken  
).encode(x=alt.X('AuthoritiesContacted:N', axis=alt.Axis(title="Kontaktierte Behörden")),  
 y=alt.Y('Prozent:Q', axis=alt.Axis(title='Prozent')),text=alt.Text('Prozent:Q'),)  
  
layered\_chart = chart + text #Kombinieren von chart und dem text  
# Anpassungen am chart  
layered\_chart.configure\_axis(labelAngle=0).properties(height=300,width=600,title="Betrugsfälle nach kontaktierten Behörden"  
).configure\_legend(orient='right',strokeWidth=1,padding=10,strokeColor='grey').configure\_view(strokeWidth=2)

alt.LayerChart(...)

Wie in H2 werden die Daten hier auch mit der groupby Methode gruppiert und in die Spalte “counts” gezählt. Die beiden erstellten Tabellen werden nun in die Tabelle grouped zusammengefügt durch den gemeinsamen Schlüssel “AuthoritiesContacted” mit einem left join. Dadurch entsteht die neue Tabelle grouped.

Daraufhin werden die Prozentzahlberechnung durchgeführt. Die Gesamtzahl der gemeldeten Fälle für jede kontaktierte Behörde wird berechnet und die Spalte “counts” wird auf die Gesamtzahl bezogen, um die Prozentzahl der gemeldeten Fälle für jede Gruppe zu berechnen. Anschließend werden auch hier die Zahlen auf die nächste Ganzzahl gerundet.

Die Visualisierung ist erneut ein gestapeltestes Balkendiagramm und die einzige sinnvolle Visualisierungsart für diesen Fall.

Auch hier werden dieselben Einstellungen und Designelemente von den vorherigen Charts verwendet.

## Ergebnis und Empfehlung

In dieser Analyse haben wir untersucht, welche Relevanz die Faktoren Alter, Geschlecht und Kontaktaufnahme zu Behörden für die Erkennung von Versicherungsbetrug bei Verkehrsunfällen haben.

Nach den aktuellen Ergebnissen, stellt sich heraus, dass es nicht ein passendes Profil für einen Betrüger gibt. Das Alter hat nur einen geringen Einfluss auf die Häufigkeit von Betrugsfällen. Jüngere Versicherungskunden unter 35 Jahren sind nicht unbedingt häufiger Betrüger als ältere. Damit ist dies kein signifikanter Faktor für die Erkennung von Betrugsfällen.

Die Betrugsrate bei Männern liegt 2,28% höher als bei Frauen, ist jedoch nicht aussagekräftig genug um dies als Faktor für die Betrugsfallerkennung einzubeziehen.

Bei der Betrachtung der Kontaktaufnahme mit Behörden gab es allerdings eine überraschende Erkenntnis. Fälle, bei denen die Polizei verständigt wurde weisen eine viel höhere Betrugsrate auf als bei keiner Kontaktaufnahme. Daher kann diese Betrachtung ein relevanter Faktor zur Erkennung von Betrugsfällen sein.

Jedoch müssen einige Punkte berücksichtigt werden. Die Stichprobe ist möglicherweise nicht repräsentativ, da sie aus einem kleinen Teil des Gesamtdatensatzes ausgewählt wurde. Ebenso könnte es sein, dass Faktoren, die wir untersucht haben, lediglich ein Indiz für die Erkennung von Betrugsfällen darstellen. Ebenso kann es sein, dass einige relevante Faktoren für die Betrugsfallerkennung im Datensatz fehlen.

Daher sollte umso mehr geprüft werden, in welchen weiteren Korrelationen Betrugsfälle häufiger vorkommen. Damit sind weitere Forschungen zur Erkennung von Betrugsfällen wichtig, um verbesserte Methoden zur Betrugsfallerkennung zu entwickeln. Außerdem sollte die Aufmerksamkeit auf die Kontaktaufnahme mit verschiedenen Behörden gelenkt werden, um zu verstehen, warum dieser Faktor einen Einfluss auf die Erkennung von Betrugsfällen hat.

Dafür müssen Mitarbeiter mit aktuellen Ergebnissen aus Datenanalysen geschult werden. Ebenso sollten Betrugserkennungssoftwares regelmäßig ein Update mit aktuellen Daten erhalten.

Die Empfehlung bleibt hiermit, die Schulung der Mitarbeiter und die kontinuierliche Verbesserung der Betrugserkennungssoftware.