



Wirtschaftsinformatik  
und Maschinelles Lernen  
Stiftung Universität Hildesheim  
Universitätsplatz 1  
31141 Hildesheim  
Prof. Dr. Dr. Lars Schmidt-Thieme  
Johannes Burchert

# **KOEX: Kollaboratives Machine Learning zur Erkennung von Fraud und Risiken in ERP-Systemen**

Teilvorhaben: Fraud- und Risiko-Erkennung in  
ERP-Systemen durch Semi-Supervised Machine  
Learning.

Sachbericht: Teil 1  
Förderkennzeichen: 16KIS1582  
Laufzeit des Vorhabens: 01.01.2022 - 30.06.2024

Das Teilvorhaben der Universität Hildesheim *Fraud- und Risiko-Erkennung in ERP-Systemen durch Semi-Supervised Machine Learning* unterteilt sich im Wesentlichen in fünf Arbeitspakete. Im ersten Schritt hat sich die Universität Hildesheim auf die Erforschung von Ansätzen zur unüberwachten Anomalie-Erkennung in ERP-Daten fokussiert. Die verwendeten Methoden sollen komplexe und heterogene Datenmuster analysieren und Anomalien zuverlässig identifizieren können, ohne auf umfassend gelabelte Datensätze angewiesen zu sein. Unüberwachte Verfahren eignen sich besonders gut für diese Art von Daten, da die in KOEX verwendeten ERP-Daten oft durch ihre Vielfalt und Sensitivität gekennzeichnet sind. Ziel ist es, betrügerische oder riskante Aktivitäten datengetrieben und effizient zu identifizieren. Dieses Arbeitspaket spielt eine zentrale Rolle im KOEX-Projekt, da die entwickelten Modelle die Grundlage für weitere Lernmethoden im Teilprojekt 4 und darüber hinaus bilden.

Im Rahmen des Arbeitspakets wurden verschiedene Methoden untersucht, darunter KNN (k-Nearest Neighbors), ECOD (Empirical Cumulative Distribution Functions for Outlier Detection), AnoGAN (Anomaly Detection using Generative Adversarial Networks) und Autoencoder. KNN nutzt lokale Nachbarschaften zur Erkennung von Anomalien und ist aufgrund seiner Flexibilität für ERP-Daten als simples Basismodell gut geeignet. ECOD basiert auf empirischen Verteilungsfunktionen und bietet eine effiziente und robuste Methode zur Erkennung von Anomalien. AnoGAN kombiniert Generative Adversarial Networks (GANs) mit Anomalie-Erkennung und bietet hohe Präzision in komplexen, hochdimensionalen Daten. Autoencoder modellieren nichtlineare Muster in ERP-Daten und erkennen Anomalien durch erhöhte Rekonstruktionsfehler. Diese Methoden sind besonders geeignet für die Analyse der oft stark heterogenen ERP-Daten.

Im zweiten Schritt untersuchte die Universität Hildesheim Ansätze für halb-überwachtes und überwachtes Lernen zur Verbesserung der Anomalieerkennung in ERP-Daten und der Klassifikation von potenziellen Betrugsfällen. Der Großteil der ERP-Daten im KOEX-Projekt liegt in tabellarischer Form vor. Für diese Datenmodalität sind Gradient Boosted Entscheidungsbaum-Modelle besonders geeignet. Hier wurden Modelle wie XGBoost und CatBoost eingesetzt, die auf Gradient Boosting für Entscheidungsbäumen basieren, um zwischen normalen und anomalen Datensätzen zu unterscheiden. Des Weiteren hat die Universität Hildesheim neuronale Netze und Transformer, die auf dem Attention Mechanism basieren, erforscht, um diese auf den ERP-Daten des KOEX-Projektes einzusetzen. TabNet ist eines dieser Modelle und nutzt eine dynamische Attention-Mechanik, um relevante Features zu gewichten, während TabTransformer auf der Transformer-Architektur basiert und komplexe Feature-Interaktionen gut modelliert.

Neben tabellarischen Daten wurden auch Zeitreihenmodelle untersucht. Modelle wie Logistische Regression und Recurrent Neural Networks (RNNs) ermöglichen die Erkennung von Anomalien durch Analyse von zeitlichen Mustern und eignen sich aufgrund ihrer geringen Komplexität als zuverlässige Basismodelle. Komplexere Modelle wie InceptionTime, basierend auf der Inception-Architektur, nutzen Convolutional Neural Networks (CNNs) zur Erfassung von kurzfristigen und langfristigen Mustern in ERP-Daten. Diese Zeitreihenmodelle sind in der Lage, Prozesse mit mehreren Änderungen in den ERP-Daten über einen längeren Zeitraum zu analysieren und temporale Merkmale zu extrahieren.

Für das dritte Arbeitspaket hat die Universität die Implementierung der Modelle vorgenommen. Hierbei ist eine sorgfältige Datenvorverarbeitung entscheidend. Die ERP-Daten enthalten viele ordinale und kategoriale Merkmale, die in maschinellen Lernverfahren problematisch sein können. Die Vorverarbeitung beginnt mit dem Auffüllen

fehlender Werte durch den Standardwert null. Datum- und Zeitinformationen werden kombiniert und in das `datetime`-Format umgewandelt, um eine chronologische Sortierung zu ermöglichen. Kategorische Spalten werden in numerische und kategorische Versionen unterteilt, und die numerischen Spalten werden standardisiert. Fehlende Werte werden ersetzt, und für die Modellierung wird eine Zielvariable **Fraud** extrahiert.

Für die Extraktion neuer Regeln aus den Modellen des maschinellen Lernens wurde der Regelkatalog der SIVIS GmbH untersucht, der spezifische Regeln zur Erkennung von Betrugsfällen beschreibt. Auf Basis dieser Regeln erstellte die Hochschule Karlsruhe gemeinsam mit Projektpartnern einen synthetischen Datensatz, um diese Betrugsfälle zu simulieren. *XGBoost* wurde zur Klassifizierung dieser Betrugsfälle verwendet und zeigte merkmalebasierte Vorhersagen, die nicht exakt den Regeln des SIVIS Regelkatalogs folgten. In Absprache mit den Projektpartnern der SIVIS GmbH, der Hochschule Karlsruhe und der prenode GmbH wurde daher ein negatives Forschungsergebnis für diesen synthetischen Datensatz festgestellt. Es könnte jedoch sein, dass ähnliche Ansätze auf echten Unternehmensdaten im Rahmen des verteiltem Lernen anwendbar wären, wenn geeignete Datengrundlagen vorhanden sind.

Die Evaluation im Teilprojekt 4 wurde hauptsächlich anhand eines synthetischen Datensatzes der Hochschule Karlsruhe, welcher im Rahmen des KOEX-Projektes erstellt wurde, und realer Unternehmensdaten von Endress+Hauser durchgeführt. Verschiedene Metriken wie Accuracy (Genauigkeit), AUC (Area Under the Curve) und F1-Score wurden herangezogen, um die Modellperformance zu bewerten. Die *Accuracy* (Genauigkeit) gibt den Anteil korrekt klassifizierter Instanzen an. Die *AUC* (Area Under the Curve) misst die Fähigkeit eines Modells, zwischen verschiedenen Klassen zu unterscheiden. Der *F1-Score* kombiniert Präzision und Recall und gibt ein ausgewogenes Maß für die Leistung des Modells. Diese verschiedenen Evaluationsmetriken sind für das KOEX-Projekt besonders relevant, da die Datensätze sehr unausgewogen sind, da die meisten der Instanzen unauffällige Geschäftsprozesse dokumentieren und nur ein geringer Teil tatsächlich Betrug sind. Das *ECOD*-Modell zeigte hervorragende Leistungen in der unüberwachten Anomalie-Erkennung mit dem besten AUC-ROC-Score und hoher Präzision bei niedrigen Anomaliezahlen. Besonders auffällig war seine Robustheit gegenüber unterschiedlichen Kontaminationsraten im Vergleich zu den anderen getesteten Modellen. In den überwachten Lernverfahren schnitten *XGBoost* und *CatBoost* in den tabellarischen Daten CDHDR und CDPOS am besten ab. Besonders *XGBoost* zeigte eine starke Präzision. Die Analyse der Merkmale bestätigte, dass leicht verständliche Attribute wie Buchungstypen und manuell erfasste Details den größten Einfluss auf die Vorhersage haben. Dies unterstützt die Praktikabilität des Modells für die Betrugserkennung.

In Absprache mit den anderen Projektpartnern wurde entschieden, das *ECOD*-Modell für das unüberwachte Lernen in der Demonstrator-Software des KOEX-Projekts aufgrund seiner guten Performanz und geringen Falsch-Positiv-Rate bei hohen Anomalie-Scores einzusetzen. Für das überwachte Lernen kamen insbesondere die Modelle *XGBoost* und *CatBoost* für die Integration in den Demonstrator in Betracht. Schlussendlich wurde *XGBoost* für den Demonstrator gewählt, da die prenode GmbH bereits über besondere Kompetenzen bei dem Einsatz dieser Architektur im Kontext des Verteilten Lernens verfügt.



Wirtschaftsinformatik  
und Maschinelles Lernen  
Stiftung Universität Hildesheim  
Universitätsplatz 1  
31141 Hildesheim  
Prof. Dr. Dr. Lars Schmidt-Thieme  
Johannes Burchert

# **KOEX: Kollaboratives Machine Learning zur Erkennung von Fraud und Risiken in ERP-Systemen**

Teilvorhaben: Fraud- und Risiko-Erkennung in  
ERP-Systemen durch Semi-Supervised Machine  
Learning.

Sachbericht: Teil 2  
Förderkennzeichen: 16KIS1582  
Laufzeit des Vorhabens: 01.01.2022 - 30.06.2024

# Inhaltsverzeichnis

1	Teilprojekt 1: Projektmanagement . . . . .	1
2	Teilprojekt 2: Projektvorbereitung und Systemkonzeption . . . . .	1
3	Teilprojekt 3: Datenmanagement . . . . .	1
4	Teilprojekt 4: Entwicklung On-Premise-Solution . . . . .	1
4.1	AP 4.1 Erforschung und Design von Modellen und Lernverfahren zum unüberwachten Lernen . . . . .	2
4.2	AP 4.2 Erforschung und Design eines integrierten halb-überwachten Verfahrens . . . . .	4
4.3	AP 4.3 Implementierung der Modelle und Lernverfahren aus AP 4.1 und AP 4.2 . . . . .	8
4.4	AP 4.4 Design und Anpassung von Methoden zur Extraktion neuer Regeln aus den gelernten Modellen in AP 4.3 . . . . .	10
4.5	AP 4.5 Experimentelle Evaluation der entwickelten Modelle auf synthetischen sowie echten Daten. . . . .	12
5	Teilprojekt 5: Entwicklung Cloud-Solution . . . . .	16
6	Teilprojekt 6: Entwicklung Demonstrator und Evaluation . . . . .	16
7	Zahlenmäßiger Nachweis . . . . .	16
8	Nutzen und Verwertbarkeit . . . . .	17
9	Bekannt gewordener Fortschritt bei anderen Stellen . . . . .	17

# 1 Teilprojekt 1: Projektmanagement

Im Rahmen des Projektmanagements hat die Universität Hildesheim das KOEX-Projekt durch die Betreuung und Planung von Prof. Dr. Dr. Lars Schmidt-Thieme unterstützt. Die Hauptarbeit lag bei der SIVIS GmbH, und wir verweisen an dieser Stelle auf den entsprechenden Sachbericht Teil 2 für nähere Informationen.

# 2 Teilprojekt 2: Projektvorbereitung und Systemkonzeption

Für Teilprojekt 2 wurden Fachkonferenzen im Bereich des maschinellen Lernens sondiert und auf relevante Publikationen untersucht. Hierbei lag der Fokus insbesondere auf A- und A\*-Konferenzen wie unter anderem ICLR, AAAI, ICML und NeurIPS. Zusätzlich hat die Universität Hildesheim an drei Präsenzworkshops in Karlsruhe unter der Leitung der SIVIS GmbH am 22.05.2022, 08.03. – 09.03.2023 und 09.10. – 10.10.2023 teilgenommen. Hierbei standen das verteilte maschinelle Lernen, Feature Engineering sowie die Planung des Vorgehens bei Industriepartnern und die Produktbeschreibung des Demonstrators, sowie die Erstellung von Personas und Anforderungsprofilen im Vordergrund. Des Weiteren hat die Universität Hildesheim in Kooperation mit der prenode GmbH einen Workshop zum Thema maschinelles Lernen vorbereitet und durchgeführt. Die restliche Betreuung des Teilprojekts 2 wurde von der SIVIS GmbH übernommen, und weitere Informationen sind dem entsprechenden Sachbericht Teil 2 zu entnehmen.

# 3 Teilprojekt 3: Datenmanagement

Bei der Erstellung der Datensätze hat die Universität Hildesheim eng mit der Hochschule Karlsruhe zusammengearbeitet. Die Hochschule Karlsruhe hat Skripte zur automatischen Simulation von Betrugsfällen entwickelt, die von der Universität weiterverarbeitet wurden. Hier wurde größtenteils eine beratende Rolle eingenommen, und die Details können dem Sachbericht Teil 2 der Hochschule Karlsruhe entnommen werden.

# 4 Teilprojekt 4: Entwicklung On-Premise-Solution

Im Rahmen des KOEX-Projekts steht das Teilprojekt 4 im Zentrum der Forschung und Entwicklung von maschinellen Lernverfahren zur Erkennung von Fraud und Risiken in ERP-Systemen. Ziel war es, eine On-Premise-Lösung zu entwickeln, die den spezifischen Anforderungen an Sicherheit und Datenschutz entspricht. Durch den lokalen Einsatz moderner Lernverfahren konnten potenzielle Fraud-Muster in ERP-Daten identifiziert und verarbeitet werden, ohne sensible Daten außerhalb der Unternehmensgrenzen zu teilen.

Das Teilprojekt hatte folgende zentrale Zielsetzungen:

- AP 4.1: Erforschung und Design von Modellen und Lernverfahren zum unüberwachten Lernen (Anomalie-Erkennung).
- AP 4.2: Erforschung und Design eines integrierten halb-überwachten Verfahrens (Anomalie-Erkennung + direkte Fraud-Erkennung).

- AP 4.3: Implementierung der Modelle und Lernverfahren aus AP 4.1 und AP 4.2.
- AP 4.4: Design und Anpassung von Methoden zur Extraktion neuer Regeln aus den gelernten Modellen in AP 4.3.
- AP 4.5: Experimentelle Evaluation der entwickelten Modelle auf synthetischen sowie echten Daten.

Bei der Arbeit an den Arbeitspaketen erfolgte eine enge Kooperation, insbesondere bei AP 4.3–4.5.

## 4.1 AP 4.1 Erforschung und Design von Modellen und Lernverfahren zum unüberwachten Lernen

Das Arbeitspaket 4.1 zielte darauf ab, Ansätze zur unüberwachten Anomalie-Erkennung in ERP-Daten zu erforschen. Der Fokus lag auf Methoden, die komplexe und heterogene Datenverteilungen analysieren und Abweichungen zuverlässig erkennen können, ohne auf umfassend gelabelte Datensätze angewiesen zu sein. Unüberwachte Verfahren sind besonders geeignet, da die in KOEX verwendeten ERP-Daten oft durch ihre Vielfalt und Sensitivität gekennzeichnet sind. Ziel war es, betrügerische oder riskante Aktivitäten datengetrieben und effizient zu identifizieren.

Dieses Arbeitspaket spielt eine zentrale Rolle im KOEX-Projekt, da die entwickelten Modelle die Grundlage für weitere Lernmethoden im Teilprojekt 4 und darüber hinaus bilden. Im Folgenden werden die erforschten Methoden detailliert beschrieben.

**KNN (k-Nearest Neighbors)** Der k-Nearest-Neighbors-Algorithmus [Peterson, 2009] ist eine flexible Methode, die durch die Analyse lokaler Nachbarschaften Anomalien identifiziert. Diese Methode ist besonders geeignet für ERP-Daten, da sie keine Annahmen über die zugrunde liegende Datenverteilung macht. Die Flexibilität, unterschiedliche Metriken zur Distanzberechnung zu nutzen, ermöglicht eine präzise Anpassung an verschiedene Datenstrukturen. Trotz ihrer Einfachheit bietet die Methode eine solide Basis zur Betrugserkennung, insbesondere in Kombination mit Approximate-Nearest-Neighbors-Optimierungen zur Reduktion der Rechenzeit.

1. **Distanzberechnung:** Für jeden Punkt wird die Distanz zu allen anderen Punkten im Datensatz berechnet, oft mit der euklidischen Distanz:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^n (x_{i,k} - x_{j,k})^2}.$$

2. **Nachbarschaftsbestimmung:** Die  $k$ -nächsten Nachbarn werden identifiziert. Der Anomalie-Score berechnet sich als Durchschnitt der Distanzen zu diesen Nachbarn:

$$\text{Score}(\mathbf{x}) = \frac{1}{k} \sum_{j=1}^k d(\mathbf{x}, \mathbf{x}_j).$$

3. **Klassifikation:** Ein Schwellenwert  $T$  entscheidet, ob ein Punkt eine Anomalie ist ( $\text{Score}(\mathbf{x}) > T$ ).

Im Rahmen dieses Arbeitspakets dient KNN als robustes Basismodell zum Vergleich mit komplexeren Methoden.

## ECOD (Empirical Cumulative Distribution Functions for Outlier Detection)

ECOD [Li et al., 2022] ist ideal für die Betrugserkennung in heterogenen und hochdimensionalen ERP-Daten, da es auf empirischen Verteilungsfunktionen basiert und keine Annahmen über die Verteilung der Daten erfordert. Seine Recheneffizienz und Robustheit ermöglichen die Verarbeitung großer Datenmengen, wie sie in ERP-Systemen typischerweise vorliegen. Mit ECOD lassen sich Anomalien, die durch ungewöhnliche Aktivitäten oder untypische Verteilungen gekennzeichnet sind, frühzeitig erkennen und adressieren.

1. **ECDF-Berechnung:** Für jede Dimension wird die empirische kumulative Verteilungsfunktion berechnet:

$$\text{ECDF}_j(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(x_{i,j} \leq x).$$

2. **Score-Berechnung:** Der Anomalie-Score ergibt sich durch Aggregation der Abweichungen über alle Dimensionen:

$$\text{Score}(\mathbf{x}) = \sum_{j=1}^d S_j(x_j).$$

Des Weiteren ist das ECOD-Modell bereits in anderen Domänen erprobt worden und hat sich als sehr leistungsstark erwiesen. Damit ist es auch im Rahmen des KOEX-Projekts für die Adaption auf eine neue Datenmodalität sehr vielversprechend.

## AnoGAN (Anomaly Detection using Generative Adversarial Networks)

AnoGAN [Schlegl et al., 2019] basiert auf GANs und eignet sich für die Anomalieerkennung in hochdimensionalen Daten. Es kombiniert die Fähigkeit eines GAN, die Trainingsdatenverteilung zu modellieren, mit einer Methode zur Identifikation von Abweichungen:

1. **GAN-Training:** Das GAN besteht aus einem Generator ( $G$ ), der synthetische Daten erzeugt, und einem Diskriminator ( $D$ ), der echte von generierten Daten unterscheidet. 2. **Latenter Raum:** Um einen Testpunkt zu analysieren, wird der latente Vektor  $\mathbf{z}^*$  gesucht, der die beste Rekonstruktion liefert. 3. **Anomalie-Score:** Der Score kombiniert den Rekonstruktionsfehler:

$$R(\mathbf{x}_{\text{test}}) = \|\mathbf{x}_{\text{test}} - G(\mathbf{z}^*)\|$$

und den Diskriminator-Fehler:

$$D_{\text{loss}}(\mathbf{x}_{\text{test}}) = \|f_D(\mathbf{x}_{\text{test}}) - f_D(G(\mathbf{z}^*))\|,$$

mit einem Gewichtungsfaktor  $\lambda$ .

AnoGAN bietet durch seine Modellierung komplexer Datenverteilungen ein hohes Potenzial für präzisere Vorhersagen, ist jedoch rechenintensiv. Im Vergleich zu ECOD hat es damit die Kapazität, im KOEX-Projekt komplexere Muster zu lernen, hat jedoch gleichzeitig den Nachteil, dass es anfälliger für das Rauschen in den heterogenen ERP-Daten ist.

**Autoencoder** Autoencoder [An and Cho, 2015] bieten eine robuste Lösung zur Modellierung nichtlinearer Muster in ERP-Daten. Ihre Fähigkeit, hochdimensionale Eingabedaten zu komprimieren und zu rekonstruieren, macht sie besonders wertvoll für die Erkennung von Anomalien, die durch erhöhte Rekonstruktionsfehler gekennzeichnet sind. Diese

Eigenschaft ist entscheidend, um komplexe, versteckte Betrugsmuster in ERP-Daten aufzudecken und präventive Maßnahmen zu ermöglichen.

1. **Architektur:** Der Autoencoder besteht aus einem Encoder, der die Eingabedaten in eine latente Darstellung  $\mathbf{z}$  komprimiert, und einem Decoder, der die Daten rekonstruiert. 2. **Training:** Ziel ist die Minimierung des Rekonstruktionsfehlers:

$$\mathcal{L}_{\text{recon}} = \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2.$$

3. **Anomalie-Score:** Der Fehler eines Testpunkts bestimmt den Score:

$$\text{Score}(\mathbf{x}_{\text{test}}) = \|\mathbf{x}_{\text{test}} - \hat{\mathbf{x}}_{\text{test}}\|_2^2.$$

## 4.2 AP 4.2 Erforschung und Design eines integrierten halbüberwachten Verfahrens

Dieses Arbeitspaket erforscht und entwickelt Ansätze für halb-überwachtes und überwachtes Lernen, um die Anomalieerkennung in ERP-Daten zu verbessern. Im KOEX-Projekt liegen die ERP-Daten häufig in tabellarischer Form vor und dadurch sind Modelle wie Boosting-Verfahren besonders gut geeignet. Diese Datentypen umfassen typische Geschäftsprozesse wie Buchungen und Bestellungen, die als Anomalien entdeckt werden müssen. Daher stellt die Erforschung integrierter halb-überwachter Verfahren eine zentrale Herausforderung dar.

### Klassifikation von tabellarischen Daten

**XGBoost** XGBoost (Extreme Gradient Boosting) [Chen, 2015] ist ein leistungsstarkes, skalierbares maschinelles Lernverfahren, das Gradient Boosting auf Entscheidungsbäumen implementiert. Diese Boosting-Methoden sind insbesondere auf tabellarischen Daten sehr gut, wodurch sie hervorragend für das KOEX-Projekt geeignet sind. Sie wurden ursprünglich für Klassifikations- und Regressionsaufgaben entwickelt, können jedoch auch für Anomalieerkennung verwendet werden, indem das Problem als Klassifikations- oder Ranking-Problem formuliert wird.

#### Grundlagen von XGBoost:

XGBoost kombiniert schwache Modelle (z. B. Entscheidungsbäume) zu einem starken Ensemblemodell. Die Funktion  $f(x)$ , die die Vorhersagen optimiert, ist additiv:

$$f(x) = \sum_{t=1}^T h_t(x),$$

wobei  $h_t(x)$  die  $t$ -te Entscheidungsregel darstellt. Die Optimierung erfolgt durch Minimierung der Verlustfunktion  $\mathcal{L}$ :

$$\mathcal{L} = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{t=1}^T \Omega(h_t),$$

wobei  $l(y_i, \hat{y}_i)$  die Fehlerfunktion und  $\Omega(h_t)$  die Regularisierungsfunktion der Modellkomplexität ist.

#### Anomalieerkennung mit XGBoost:

**Überwacher Ansatz:** Gelabelte Daten werden verwendet, wobei normale Daten mit  $y = 0$  und Anomalien mit  $y = 1$  markiert werden. Das Modell wird trainiert, diese Labels vorherzusagen.

**Halb-überwacher Ansatz:** Ein Teil der Daten wird künstlich als potenzielle Anomalien markiert, und XGBoost lernt eine Trennfunktion zwischen normalen und abweichenden Daten.

Der Anomalie-Score ergibt sich aus dem vorhergesagten Wert  $\hat{y}$ :

$$\text{Score}(\mathbf{x}) = \hat{y}(\mathbf{x}),$$

wobei höhere Werte größere Anomalie-Wahrscheinlichkeiten anzeigen. XGBoost ist zusätzlich sehr gut für das verteilte Lernen unter der Leitung der prenode GmbH im Rahmen des KOEX-Projekts geeignet.

**CatBoost** CatBoost basiert ebenfalls auf Gradient Boosting über Entscheidungsbäume und bietet spezifische Optimierungen für kategoriale Daten:

- **Optimierung für kategoriale Daten:** Direkte Umwandlung ohne vollständiges One-Hot-Encoding.
- **Ordinales Target-Encoding:** Vermeidet Ziel-Leakage durch eine spezielle Encoding-Methode.
- **Gleichmäßige Schätzung von Gradienten:** Reduziert Vorhersageverzerrungen.

Gerade diese Eigenschaften machen es für das KOEX-Projekt sehr interessant, da die verwendeten ERP-Daten viele kategoriale Merkmale enthalten, die eine Herausforderung für das maschinelle Lernen darstellen.

#### **Anomalieerkennung mit CatBoost:**

Im Rahmen der Betrugserkennung wird das Problem als überwachte Klassifikationsaufgabe formuliert. Normale Daten werden mit  $y = 0$  und Betrugsfälle mit  $y = 1$  gekennzeichnet. CatBoost berechnet die Anomalie-Wahrscheinlichkeit:

$$\text{Score}(\mathbf{x}) = P(y = 1 | \mathbf{x}),$$

wobei höhere Werte auf eine größere Anomalie-Wahrscheinlichkeit hinweisen.

CatBoost bietet folgende Vorteile:

- Effiziente Verarbeitung von kategorialen Daten, wie sie in ERP-Daten (z. B. CDH-DR/CDPOS, BKPF/BSEG) häufig vorkommen.
- Klare Interpretierbarkeit der Feature-Bedeutung.
- Hohe Robustheit bei unausgewogenen Daten.

Herausforderungen bestehen jedoch bei der Abhängigkeit von gelabelten Daten und der Leistung bei hochdimensionalen oder verrauschten Daten ohne geeignete Vorverarbeitung.

**TabNet** TabNet [Arik and Pfister, 2021] kombiniert Entscheidungsbäume und neuronale Netzwerke, um tabellarische Daten effizient zu modellieren. Es verwendet eine dynamische *Attention*-Mechanik, die relevante Eingabefeatures selektiv gewichtet und verarbeitet. Auch wenn boosted Entscheidungsbaum-Modelle in der aktuellen Forschung die besten Ergebnisse für tabellarische Daten liefern, sind neuronale Netzwerke in anderen Bereichen des maschinellen Lernens in der Regel besser. Deshalb werden diese auch für die spezifischen Daten des KOEX-Projekts evaluiert, um zu untersuchen, ob sich dieser Trend in der allgemeinen Literatur bestätigt.

**Funktionsweise:**

TabNet nutzt adaptives Feature-Selection in mehreren Schritten:

$$z_t = \text{Attention}_t(x_{t-1}, a_{t-1}) \cdot x_{t-1},$$

wobei  $x_{t-1}$  die Eingabe und  $a_{t-1}$  die Aufmerksamkeit des vorherigen Schritts ist. Diese iterativen Schritte fokussieren auf relevante Features und verbessern die Generalisierung.

Ein Sparsity-Mechanismus reguliert die Modellkomplexität, indem nur eine Teilmenge der Features berücksichtigt wird. Dies reduziert die Rechenlast und unterstützt die Generalisierung.

Die selbstlernenden Mechanismen machen das System robust gegenüber verrauschten und hochdimensionalen Daten, während die interpretierbare Architektur durch die nachvollziehbare Gewichtung einzelner Features Transparenz und Verständlichkeit bietet.

**TabTransformer** TabTransformer [Huang et al., 2020] basiert auf der Transformer-Architektur und ist speziell auf tabellarische Daten zugeschnitten. Es integriert numerische und kategoriale Daten, wobei letztere durch Embeddings dargestellt werden.

**Funktionsweise:**

Der Self-Attention-Mechanismus modelliert Feature-Interaktionen:

$$H^{(l+1)} = \text{TransformerLayer}(H^{(l)}, W),$$

wobei  $H^{(l)}$  die Eingabe nach der  $l$ -ten Schicht ist. Dieser Mechanismus erfasst globale Beziehungen zwischen Features und ist besonders leistungsstark für komplexe Interaktionen.

Im Vergleich zu TabNet, das sich auf Feature-Sparsity konzentriert und interpretierbar ist, wodurch es besonders für große Datensätze geeignet ist, zeichnet sich TabTransformer durch die Modellierung tiefer Interaktionen aus und ist bei komplexen Beziehungen zwischen Variablen häufig überlegen. Beide Modelle eignen sich hervorragend für die Verarbeitung tabellarischer ERP-Daten, jedoch bietet TabNet eine bessere Interpretierbarkeit, während TabTransformer bei hochkomplexen Feature-Interaktionen überlegene Ergebnisse liefern kann.

## Klassifikation von Zeitreihen

Im Rahmen des KOEX-Projekts hat die Universität Hildesheim außerdem Zeitreihenmodelle analysiert. Diese Modelle sind in der Lage, aus Sequenzen wie den Prozessen in den verwendeten ERP-Daten zusätzlich temporale Merkmale zu lernen. Für die korrekte Klassifikation von Betrugsfällen ist dies eine wichtige Ergänzung, da auch zeitliche Verhältnisse ein wichtiger Indikator für verdächtiges Verhalten sein können.

**Logistische Regression** Die logistische Regression [LaValley, 2008] ist ein Verfahren des überwachten Lernens, das insbesondere für binäre Klassifikationsprobleme, wie im Fall von KOEX 'Betrug' und 'Nicht-Betrug', eingesetzt wird. Ziel ist es, eine Wahrscheinlichkeitsfunktion zu modellieren, die die Wahrscheinlichkeit eines Ereignisses oder einer Klasse basierend auf Eingabedaten vorhersagt. Im Gegensatz zur linearen Regression, die kontinuierliche Werte liefert, gibt die logistische Regression Wahrscheinlichkeiten im Bereich  $[0, 1]$  aus.

Die Berechnung basiert auf der Sigmoid-Funktion, welche eine lineare Kombination der Eingabedaten in eine Wahrscheinlichkeit transformiert. Die Grundformel lautet:

$$P(y = 1|X) = \sigma(\mathbf{w}^\top \mathbf{x} + b)$$

wobei  $\sigma(z) = \frac{1}{1+e^{-z}}$  die Sigmoid-Funktion ist. Hierbei bezeichnet  $\mathbf{w}$  den Gewichtsvektor,  $b$  den Bias-Term und  $\mathbf{x}$  die Eingabedaten.

Die Parameter  $\mathbf{w}$  und  $b$  werden so optimiert, dass die log-likelihood-Funktion maximiert wird:

$$L(\mathbf{w}, b) = \sum_{i=1}^n [y^{(i)} \log(P(y^{(i)} = 1|\mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - P(y^{(i)} = 1|\mathbf{x}^{(i)}))].$$

Dies geschieht üblicherweise mittels Gradientenabstieg. Nach der Modellanpassung liefert die logistische Regression Vorhersagen in Form von Wahrscheinlichkeiten. Zur Klassifikation wird ein Schwellenwert, häufig 0.5, verwendet:

$$\hat{y} = \begin{cases} 1 & \text{wenn } P(y = 1|X) > 0.5, \\ 0 & \text{sonst.} \end{cases}$$

Die logistische Regression ist aufgrund ihrer Einfachheit, Effizienz und Interpretierbarkeit ein weit verbreitetes Klassifikationsverfahren.

**Recurrent Neural Networks** RNNs [Medsker et al., 2001] sind neuronale Netzwerke, die speziell für die Verarbeitung sequenzieller Daten entwickelt wurden. Im Gegensatz zu Feedforward-Netzwerken berücksichtigen RNNs vorherige Eingaben, um zeitliche Abhängigkeiten zu modellieren. Dies ist besonders nützlich in den Trace-Daten des KOEX-Projekts, da hier die Reihenfolge der Prozessschritte bekannt ist.

Ein RNN speichert bei jedem Zeitschritt  $t$  einen verborgenen Zustand  $\mathbf{h}_t$ , der von der aktuellen Eingabe  $\mathbf{x}_t$  und dem vorherigen Zustand  $\mathbf{h}_{t-1}$  abhängt:

$$\mathbf{h}_t = f(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + \mathbf{b}_h),$$

wobei  $\mathbf{W}_h$  und  $\mathbf{W}_x$  Gewichtsmatrizen,  $\mathbf{b}_h$  ein Bias-Term und  $f$  eine Aktivierungsfunktion wie Tanh oder Sigmoid sind. Die Ausgabe des Netzwerks  $\mathbf{y}_t$  wird aus  $\mathbf{h}_t$  berechnet:

$$\mathbf{y}_t = g(\mathbf{W}_y \mathbf{h}_t + \mathbf{b}_y).$$

Das Training erfolgt durch Backpropagation Through Time (BPTT), eine Erweiterung des klassischen Backpropagation-Algorithmus. RNNs haben jedoch Schwierigkeiten mit langfristigen Abhängigkeiten, da Gradienten während des Trainings exponentiell abnehmen oder anwachsen können (vanishing/exploding gradient problem). Erweiterte Architekturen wie Long Short-Term Memory (LSTM) und Gated Recurrent Units (GRU) adressieren diese Probleme.

**InceptionTime** InceptionTime [Ismail Fawaz et al., 2020] ist ein Modell für die Zeitreihenklassifikation, das auf der Inception-Architektur aus der Bildverarbeitung basiert. Es kombiniert die Fähigkeit von Convolutional Neural Networks (CNNs), lokale Merkmale zu extrahieren, mit mehreren Filtergrößen, um sowohl kurz- als auch langfristige Muster zu erfassen.

Eine Eingabezeitreihe  $\mathbf{x} = (x_1, x_2, \dots, x_T)$  wird durch parallele Convolutional-Schichten mit unterschiedlichen Kernelgrößen verarbeitet. Jede Schicht extrahiert Merkmale auf verschiedenen Skalen. Die Ergebnisse werden zu einer repräsentativen Merkmalsdarstellung zusammengeführt:

$$\mathbf{h}_t = \text{concat}(\mathbf{h}_t^{(1)}, \mathbf{h}_t^{(2)}, \dots, \mathbf{h}_t^{(N)}),$$

wobei  $\mathbf{h}_t^{(i)}$  das Ergebnis der  $i$ -ten Convolutional-Schicht ist, und  $\text{concat}$  die Zusammenführung der Ergebnisse bezeichnet.

Eine globale Durchschnittspooling-Schicht reduziert die Längenabhängigkeit der Zeitreihe, während das finale Ergebnis durch ein vollständig verbundenes Netzwerk erzeugt wird. Das Training erfolgt typischerweise durch Gradientenabstieg, wobei als Verlustfunktion die Kreuzentropie für Klassifikationsaufgaben verwendet wird.

Dank seiner flexiblen Architektur erfasst InceptionTime sowohl lokale als auch globale Merkmale und bietet eine robuste Lösung für die Klassifikation von Betrugsfällen.

### 4.3 AP 4.3 Implementierung der Modelle und Lernverfahren aus AP 4.1 und AP 4.2

Für die Implementierung der Modelle und Lernverfahren ist insbesondere die Datenvorverarbeitung essentiell, da die ERP-Daten viele ordinale und kategoriale Merkmale enthalten, wobei insbesondere letztere aufgrund ihrer Heterogenität schwierig im maschinellen Lernen anzuwenden sind. Die Erprobung der Modelle und der entsprechenden Vorverarbeitung wurde insbesondere auf dem synthetischen Datensatz der Hochschule Karlsruhe durchgeführt. Details zur Erstellung dieses Datensatzes entnehmen Sie bitte dem Sachbericht Teil 2 der Hochschule Karlsruhe.

#### Datenvorverarbeitung

**Änderungsbelege** Die Vorverarbeitung der Änderungsbelege in den Tabellen CDH-DR/CDPOS beginnt mit dem Auffüllen fehlender Werte durch den Standardwert null. Um Datum- und Zeitinformationen effizient zu nutzen, werden die Spalten UDATE und UTIME kombiniert, um eine neue Spalte namens DATETIME zu erstellen. Diese wird in das `datetime`-Format umgewandelt und für eine chronologische Sortierung des Datensatzes verwendet. Nach der Sortierung wird der Index zurückgesetzt und die temporären Spalten für Datum und Zeit entfernt, da sie keine weiteren Zwecke erfüllen.

Ein zentraler Bestandteil der Vorverarbeitung ist die Behandlung von Spalten, die sowohl numerische als auch kategoriale Werte enthalten, wie OBJECTID, TABKEY, VALUE\_OLD und VALUE\_NEW. Jede dieser Spalten wird in zwei neue Spalten aufgeteilt: eine numerische Version, die nur Zahlen enthält, und eine kategoriale Version, die Nicht-Zahlen speichert. Fehlende Werte in den numerischen Spalten werden durch null ersetzt, während die kategorialen Spalten mit leeren Zeichenketten aufgefüllt werden. Nach der Transformation werden die ursprünglichen Spalten gelöscht, um Redundanzen zu vermeiden.

Zur Vorbereitung der numerischen Daten werden diese entsprechend ihres Typs konvertiert, entweder in `float` oder `int`. Bestimmte numerische Spalten, die als Identifikatoren dienen, werden in einem separaten `DataFrame` gespeichert, der diese Werte aus den restlichen Daten entfernt, um sie für spezifische Analysen verfügbar zu machen.

Für überwachtes Lernen wird die Zielvariable `Fraud` extrahiert. Falls der Prozess unüberwacht ist, werden diese Zielvariablen zusammen mit anderen irrelevanten Spalten wie `Prozess` entfernt. Die verbleibenden Spalten werden anschließend für die Modellentwicklung verwendet.

Die kategorialen Spalten, wie etwa `MANDANT`, `OBJECTCLAS`, `USERNAME` oder `VALUE_OLD_cat`, werden in Strings konvertiert, und ihre Indizes werden gespeichert, um sie später gezielt referenzieren zu können. Schließlich werden die numerischen Spalten mit einem `StandardScaler` standardisiert, um sie für maschinelle Lernverfahren geeignet zu machen. Nach der Skalierung werden eventuell entstandene fehlende Werte durch `null` ersetzt, um eine vollständige Datenstruktur zu gewährleisten. Das Ergebnis der Vorverarbeitung wird in den Features `X` gespeichert, während die Label in der Variable `y` abgelegt werden, falls sie vorhanden sind.

**Accounting-Datensatz** Die Vorverarbeitung der Tabellen `BKPF/BSEG` beginnt mit der Erstellung einer neuen Spalte, `DATETIME`, durch die Kombination der Spalten `CPUTM` (Uhrzeit) und `CPUDT` (Datum). Diese Kombination wird in ein `datetime`-Format umgewandelt, um eine korrekte Zeitreihenanalyse zu ermöglichen. Anschließend wird das Dataset nach dieser neuen `DATETIME`-Spalte aufsteigend sortiert, und der Index wird zurückgesetzt, um eine geordnete Struktur zu gewährleisten. Nach der Sortierung werden die ursprünglichen Spalten `CPUTM` und `CPUDT` sowie die temporäre `DATETIME`-Spalte entfernt, da sie nun nicht mehr benötigt werden.

Ein weiterer Schritt im Preprocessing ist die Trennung von Spalten, die sowohl numerische als auch kategoriale Werte enthalten. In diesem Fall wird für jede betroffene Spalte eine numerische (`<column_name>_num`) und eine kategoriale (`<column_name>_cat`) Spalte erstellt. Die numerischen Werte werden in die neue numerische Spalte übertragen, während textuelle Werte in der neuen kategorialen Spalte gespeichert werden. Fehlen in den Spalten numerische oder kategoriale Werte, so werden diese durch `Null` (für numerische Werte) bzw. einen leeren String (für kategoriale Werte) ersetzt. Die ursprüngliche Spalte wird danach gelöscht, da sie durch die neuen Spalten ersetzt wurde.

Zusätzlich werden bestimmte Spalten aus dem Dataset entfernt, die entweder Datumswerte oder Währungsinformationen enthalten oder die überwiegend nur Nullwerte aufweisen. Zu diesen Spalten gehören unter anderem Datumsfelder wie `AUGDT`, `BUDAT`, `FDTAG` und `UPDDT` sowie Währungsfelder wie `WAERS`, `HWAER` und `HWAE2`. Ebenfalls gelöscht werden Spalten, die als irrelevant oder leer erachtet werden, wie `BKTX`, `TXBHW` oder `DMBTR`. Dies hilft, das Dataset zu bereinigen und die Datenmenge zu reduzieren, sodass nur relevante Informationen verbleiben.

Abschließend erfolgt eine Trennung der Spalten in zwei Gruppen: die kategorialen Spalten und die numerischen Spalten. Kategoriale Spalten sind solche, die Textwerte enthalten, während numerische Spalten aus Zahlen bestehen. Diese Trennung ist wichtig für viele Analyse- oder Modellierungstechniken, da verschiedene Algorithmen unterschiedliche Arten von Daten benötigen. Insgesamt dient dieses Preprocessing der Strukturierung und Bereinigung der Daten, um sie für weitere Analysen oder maschinelles Lernen vorzubereiten.

## Implementierung der Modelle

**Unüberwachte Methoden** Für die Implementierung der unüberwachten Methoden wurde das Benchmark-Framework [Han et al., 2022] *ADBench: Anomaly Detection Benchmark* verwendet. Hierbei wurden die Modellkomponenten, wie unter anderem die Anzahl der Schichten für AnoGAN und den Autoencoder, optimiert. Des Weiteren wurden die Hyperparameter angepasst. Von besonderem Interesse für den synthetischen Datensatz, der im Rahmen des KOEX-Projektes erstellt wurde, ist der Hyperparameter der Kontaminationsrate. Dieser gibt an, wie viele Anomalien in den Daten erwartet werden, und bestimmt den Schwellenwert.

**Überwachte Methoden** Die Implementierung der überwachten Methoden erfolgte durch die Anpassung der jeweiligen Veröffentlichungen. Das InceptionTime-Modell für den Zeitreihen-Use-Case wurde reimplementiert. Auch für diese Modelle wurden die Hyperparameter wie Lernrate, Regularisierungsstärke und die Tiefe des Netzwerks optimiert. Aufgrund der hohen Imbalance zwischen den *Betrugs-* und *Nicht-Betrugs-*Klassen wurde ein Upsampling für die Minderheitsklasse sowie eine gewichtete Zielfunktion verwendet, um die Performance zu verbessern.

### 4.4 AP 4.4 Design und Anpassung von Methoden zur Extraktion neuer Regeln aus den gelernten Modellen in AP 4.3

Für das AP 4.4 zur Extraktion neuer Regeln wurde als erster Schritt der Regelkatalog von der SIVIS GmbH untersucht. Dieser ist in Abbildung 1 dargestellt. Es gibt sowohl eine Beschreibung als auch die entsprechenden Tabellenänderungen, die im ERP-System vorgenommen werden müssen. Zum Beispiel erfordert der Betrugsfall des Anbieters FlipFlop mindestens sieben Änderungen im System. Dabei wird unter anderem die Bankverbindung auf die des Täters geändert und nach Abschluss der Transaktion wieder rückgängig gemacht, um die Spuren zu verwischen. Auf Basis dieser Regeln hat die Hochschule Karlsruhe in Absprache mit den anderen Projektpartnern, unter anderem der Universität Hildesheim, den synthetischen Datensatz erstellt, in dem die abgebildeten Betrugsfälle simuliert wurden.

Im nächsten Schritt wurde XGBoost verwendet, um die Betrugsfälle in diesem synthetischen Datensatz zu klassifizieren. Der resultierende Entscheidungsbaum ist in Abbildung 2 dargestellt. Hierbei wurde festgestellt, dass das XGBoost-Modell sinnvolle Merkmale des Datensatzes untersucht, wie in Abbildung 6 dargestellt. Diese Merkmale folgen jedoch nicht dem Schema, das in der SIVIS GmbH verwendet wird. Dadurch sind die resultierenden Entscheidungsbäume aus der Sicht des maschinellen Lernens zwar interessant, jedoch nicht in der Praxis geeignet, um neue Betrugsfälle zu kategorisieren. In Absprache mit den Projektpartnern der SIVIS GmbH, Hochschule Karlsruhe und prenode GmbH wurde somit ein negatives Forschungsergebnis für diesen synthetischen Datensatz festgestellt. Es ist möglich, dass dieser Ansatz im Rahmen von verteiltem Lernen auf echten Unternehmensdatensätzen anwendbar ist. Allerdings war die benötigte Datengrundlage zur Zeit der KOEX-Projektdauer noch nicht verfügbar.

<b>Vendor Flipflop</b>	Account number of a supplier is changed shortly before payment and changed back again after payment (= diverted payment) ME21N → ME29N → MIGO → MIRO → XK02 → F-53 → XK02
<b>CpD I</b>	Use of a one-time supplier (Conto per miscellaneous), although the supplier of the order has already been created in the master data ME21N → ME29N → MIGO → MIRO → F-53
<b>CpD II</b>	Multiple use of a one-time supplier (Conto per miscellaneous) for the same supplier without creating master data for the supplier ME21N → ME29N → MIGO → MIRO → F-53
<b>Similar vendors</b>	Create a supplier in the master data that is similar to a regular supplier except for the bank data, regular ordering from normal supplier and occasional ordering and payment of the fake supplier. ME21N → ME29N → MIGO → MIRO → F-53
<b>Different payee I</b>	Option "Different payee" activated in the supplier's master data, no bank data specified in the master data, upon receipt of the invoice the payee can be entered (and thus changed) manually each time ME21N → ME29N → MIGO → MIRO → FB03 → F-53
<b>Different payee II</b>	Same as Different payee I, except that the supplier has bank data in the master data. It is still possible to select a different payee each time an invoice is received by using this option. ME21N → ME29N → MIGO → MIRO → FB03 → F-53
<b>often changed purchase orders</b>	Order prices and quantities in the system changed several times after creation or after release of the order ME21N → ME22N → ME22N → ME22N → ME29N → ME22N → ME22N → MIGO → MIRO → F-53

Abbildung 1: Regelkatalog für Betrugsfälle der Sivis GmbH.

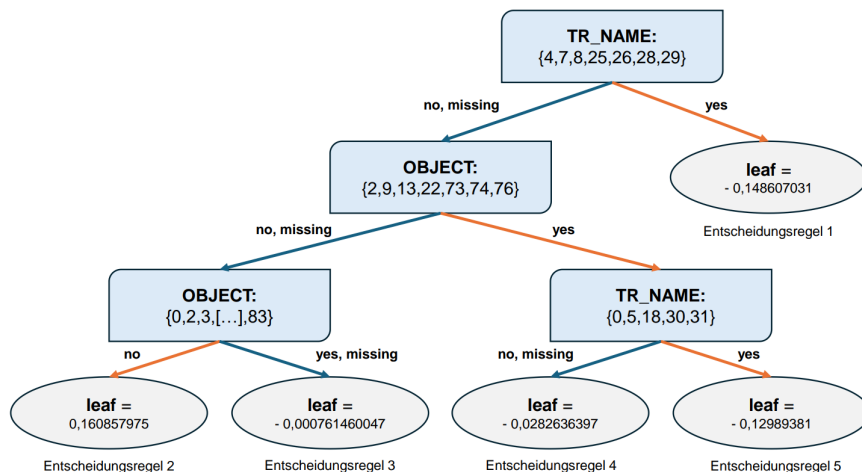


Abbildung 2: Gelernter Entscheidungsbaum des XGBoost Modelles.

	ECOD	AutoEncoder	CBLOF	AnoGAN	KNN
AUC-ROC	0.6611	<b>0.6756</b>	<u>0.672</u>	0.4795	0.454
precision@10	<b>1.000</b>	0.400	0.200	0.000	<u>0.600</u>
precision@25	<b>1.000</b>	0.160	0.120	0.040	<u>0.800</u>
precision@50	<b>0.920</b>	0.120	0.120	0.100	<u>0.400</u>
precision@100	<b>0.860</b>	0.130	0.080	0.070	<u>0.240</u>
precision@500	<u>0.440</u>	<b>0.552</b>	0.416	0.068	<u>0.094</u>
precision@1000	<u>0.279</u>	<b>0.289</b>	0.271	0.069	0.072

Tabelle 1: Ergebnisse des Unüberwachten Lernens.

## 4.5 AP 4.5 Experimentelle Evaluation der entwickelten Modelle auf synthetischen sowie echten Daten.

Die Evaluation des Teilprojekts 4 wurde vorwiegend auf dem synthetischen Datensatz der Hochschule Karlsruhe und den realen Unternehmensdaten von Endress+Hauser vorgenommen. Im Folgenden wurden die Modelle auf der Basis verschiedener Metriken evaluiert.

Die *Accuracy* (Genauigkeit) misst den Anteil korrekt klassifizierter Instanzen:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Die *AUC* (Area Under the Curve) ist die Fläche unter der ROC-Kurve und beschreibt die Fähigkeit, zwischen Klassen zu unterscheiden:

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}) d(\text{FPR})$$

Hierbei sind:

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN}$$

Der *F1-Score* ist das harmonische Mittel von Präzision und Recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Mit:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

Hierbei sind die AUC und der F1-Score von Interesse, da die Datensätze unausbalancierte Klassenverteilungen aufweisen. Diese liegt für den BSEG/BKPF-Datensatz bei 91,79% *Nicht-Betrugs*-Instanzen.

**Evaluation - Unüberwachtes Lernen** Die Ergebnisse für das unüberwachte Lernen zur Identifizierung von Anomalien zeigen sehr gute Ergebnisse für das eingesetzte *ECOD*-Modell in Tabelle 2. Es weist den zweitbesten AUC-ROC-Score auf, und die Precision für niedrige Anzahlen von Instanzen ist deutlich besser als bei vergleichbaren Modellen. Die *precision@n* gibt an, wie viele der Instanzen in den obersten *n* mit dem höchsten Anomalie-Score tatsächlich eine Anomalie sind. Die Marktforschung und Erstellung von

model	ECOD	AnoGAN	KNN	
ROC	0,6441	0,6600	0,5068	
precision @ rank n	1,0000	0,2000	1,0000	
precision @ rank 5	1,0000	0,2000	1,0000	
precision @ rank 10	1,0000	0,4000	0,8000	
precision @ rank 15	1,0000	0,6000	0,6667	
precision @ rank 20	1,0000	0,7000	0,8000	
precision @ rank 50	1,0000	0,6800	0,6800	
precision @ rank 100	0,9000	0,5200	0,3878	
precision @ rank 250	0,5120	0,3280	0,1727	
precision @ rank 500	0,3120	0,2580	0,1122	fraudRatio =
precision @ rank 670	0,2552	0,2418	0,0851	0,02
model	ECOD	AnoGAN	KNN	
ROC	0,6441	0,4698	0,5068	
precision @ rank n	1,0000	0,2000	1,0000	
precision @ rank 5	1,0000	0,2000	1,0000	
precision @ rank 10	1,0000	0,4000	0,8000	
precision @ rank 15	1,0000	0,6000	0,6667	
precision @ rank 20	1,0000	0,4500	0,8000	
precision @ rank 50	1,0000	0,3000	0,6800	
precision @ rank 100	0,9000	0,2200	0,3878	
precision @ rank 250	0,5120	0,1360	0,1727	
precision @ rank 500	0,3120	0,1120	0,1122	fraudRatio =
precision @ rank 670	0,2552	0,1000	0,0851	0,001

Abbildung 3: Untersuchung des Kontaminations-Hyperparameters für eine Betrugsrate von 0.02 und 0.001.

Personas in Kooperation mit den KOEX Projektpartner hat ergeben, dass die Anwender der automatischen Betrugserkennung einen hohen Wert auf die Vermeidung von *False Positives* legen. Diese Eigenschaft ist insbesondere bei diesem Modell gegeben.

Des Weiteren ist es robust gegenüber der Kontaminationsrate, welche ein Hyperparameter ist, der vom Modell selbst nicht gelernt werden kann. Dieser muss vorab eingestellt werden und ist in den meisten Unternehmen nicht bekannt, wodurch eine Abweichung zum realen Wert entstehen kann. Der Einfluss der Wahl dieses Parameters ist in Abbildung 3 bis Abbildung 5 dargestellt. Dieses Experiment erfolgte auf den Änderungsbelegen mit einer realen Betrugsrate von 0,072. Während sowohl *ECOD* als auch das einfache *KNN*-Modell keine Veränderung zeigen, wenn dieser Parameter geändert wird, zeigt das *AnoGAN*-Modell schlechtere Ergebnisse bei unpassender Hyperparameter-Wahl. Aufgrund dieser kombinierten Eigenschaften wurde in Absprache mit allen Projektpartnern entschieden, *ECOD* in den Demonstrator zu integrieren.

**Evaluation - Überwachtes Lernen** Für das überwachte Lernen sind insbesondere die Accuracy und der AUC-Score relevant. Hier sind die gradient-boosted Entscheidungsbäume *XGBoost* und *CatBoost* die besten Modelle für die Änderungsbelegtabellen CDHDR und CDPOS. Während die Genauigkeit für alle Modelle vergleichbar ist, wurde ein deutlicher Unterschied im AUC-Score festgestellt, was bedeutet, dass die Attention-basierten Architekturen in diesem Fall für den *Recall* schlechtere Ergebnisse liefern. Diese Erkenntnis deckt sich auch mit einer Studie von [Grinsztajn et al., 2022], die untersucht, warum Entscheidungsbaum-Modelle für tabellarische Daten immer noch besser abschneiden als komplexe Neuronale Netze im Vergleich zu anderen Forschungsfeldern des maschinellen Lernens. In Absprache mit allen Projektpartnern und insbesondere prenode GmbH wurde *XGBoost* als Modell für den Demonstrator ausgewählt, aufgrund seiner guten Performance und der einfachen Integration in das System für das verteilte Lernen.

Des Weiteren wurden die wichtigsten Merkmale des Datensatzes bei der Anwendung

model	ECOD	AnoGAN	KNN	
ROC	0,6441	0,4403	0,5068	
precision @ rank n	1,0000	0,2000	1,0000	
precision @ rank 5	1,0000	0,2000	1,0000	
precision @ rank 10	1,0000	0,3000	0,8000	
precision @ rank 15	1,0000	0,2667	0,6667	
precision @ rank 20	1,0000	0,2500	0,8000	
precision @ rank 50	1,0000	0,2000	0,6800	
precision @ rank 100	0,9000	0,2100	0,3878	
precision @ rank 250	0,5120	0,1120	0,1727	
precision @ rank 500	0,3120	0,0720	0,1122	fraudRatio =
precision @ rank 670	0,2552	0,0672	0,0851	0,01
model	ECOD	AnoGAN	KNN	
ROC	0,6441	0,3156	0,5068	
precision @ rank n	1,0000	0,0000	1,0000	
precision @ rank 5	1,0000	0,0000	1,0000	
precision @ rank 10	1,0000	0,0000	0,8000	
precision @ rank 15	1,0000	0,0000	0,6667	
precision @ rank 20	1,0000	0,0000	0,8000	
precision @ rank 50	1,0000	0,0000	0,6800	
precision @ rank 100	0,9000	0,0000	0,3878	
precision @ rank 250	0,5120	0,0280	0,1727	
precision @ rank 500	0,3120	0,0480	0,1122	fraudRatio =
precision @ rank 670	0,2552	0,0448	0,0851	0,05

Abbildung 4: Untersuchung des Kontaminations-Hyperparameters für eine Betrugsrate von 0.01 und 0.05.

model	ECOD	AnoGAN	KNN	
ROC	0,6441	0,5802	0,5068	
precision @ rank n	1,0000	0,2000	1,0000	
precision @ rank 5	1,0000	0,2000	1,0000	
precision @ rank 10	1,0000	0,3333	0,8000	
precision @ rank 15	1,0000	0,6000	0,6667	
precision @ rank 20	1,0000	0,6316	0,8000	
precision @ rank 50	1,0000	0,6600	0,6800	
precision @ rank 100	0,9000	0,4200	0,3878	
precision @ rank 250	0,5120	0,2560	0,1727	
precision @ rank 500	0,3120	0,1760	0,1122	fraudRatio =
precision @ rank 670	0,2552	0,1493	0,0851	0,07
model	ECOD	AnoGAN	KNN	
ROC	0,6441	0,3967	0,5068	
precision @ rank n	1,0000	0,4000	1,0000	
precision @ rank 5	1,0000	0,4000	1,0000	
precision @ rank 10	1,0000	0,2000	0,8000	
precision @ rank 15	1,0000	0,1333	0,6667	
precision @ rank 20	1,0000	0,1000	0,8000	
precision @ rank 50	1,0000	0,0400	0,6800	
precision @ rank 100	0,9000	0,0400	0,3878	
precision @ rank 250	0,5120	0,0400	0,1727	
precision @ rank 500	0,3120	0,0360	0,1122	fraudRatio =
precision @ rank 670	0,2552	0,0418	0,0851	0,1

Abbildung 5: Untersuchung des Kontaminations-Hyperparameters für eine Betrugsrate von 0.07 und 0.1.

	Random	KNN	XGBoost	CatBoost	TabNet	TabTran
Loss	3.2497	<b>1.0602</b>	<b>0.1640</b>	0.1847	<b>0.2242</b>	0.3748
Acc	0.8653	0.9524	0.9478	<b>0.9498</b>	0.9451	0.933
AUC	0.4901	0.7886	<b>0.8922</b>	0.8431	0.6986	0.5

Tabelle 2: Ergebnisse des Überwachten Lernens für Tabellarische Daten auf Änderungsbelegen.

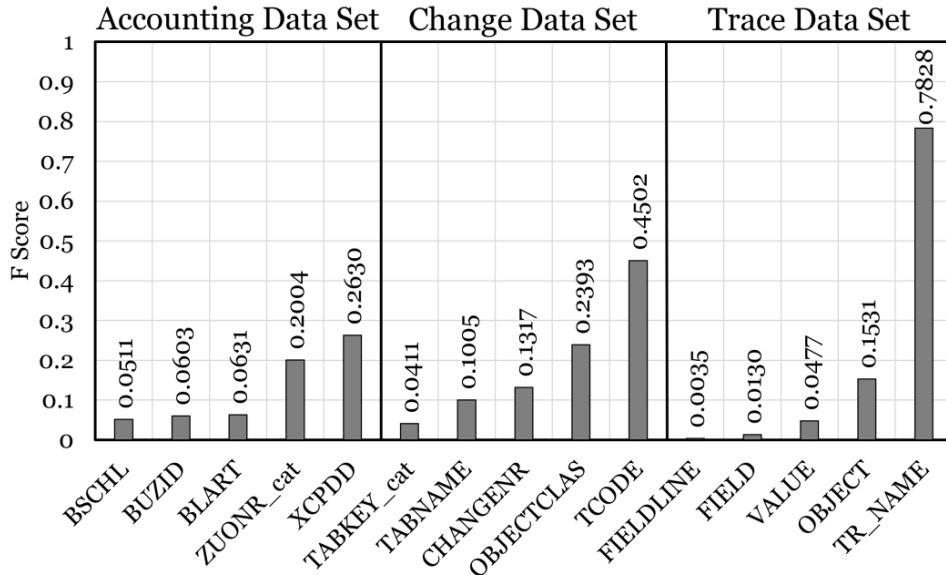


Abbildung 6: Gewichtung der Merkmale bei der Anwendung von XGBoost [Schnepf et al., 2023]

von *XGBoost* untersucht und in einer Studie der Hochschule Karlsruhe im Rahmen des KOEX-Projekts in [Schnepf et al., 2023] veröffentlicht. Die Analyse der Attributbedeutung zeigte, dass insbesondere leicht verständliche Attribute, wie Verweise auf andere Dokumente oder manuell erfasste Details, einen starken Einfluss auf die Vorhersage haben. Auch Attribute, die Buchungstypen (debitorisch/kreditorisch), Belegarten, betroffene Bereiche oder den Ort einer Änderung (z. B. Tabelle, Feld) spezifizieren, erwiesen sich als besonders relevant – dabei handelt es sich meist um intuitiv nachvollziehbare, nicht abstrakte Werte, was ein weiterer Indikator für die Güte des *XGBoost*-Modells im Kontext der Betrugserkennung ist.

Zusätzlich wurden im Accounting-Datensatz BSEG/BKPF mehrere Instanzen zeitlich aneinandergesetzt, um diese als Zeitreihe darzustellen. Damit ist es möglich, Modelle aus der Zeitreihenklassifikation anzuwenden. Die Ergebnisse sind Tabelle 3 zu entnehmen. In diesem Fall schneiden die Attention-basierten Modelle des *Transformers* und das designierte Zeitreihen-Modell *InceptionTime* besser ab als das Entscheidungsbaum-Modell *XGBoost*. Des Weiteren hat eine einfache *RNN*-Architektur deutliche Vorteile gegenüber einem tiefen Neuronalen Netz (*DNN*) und der *Logistischen Regression*, die nicht in der Lage sind, die zeitlichen Abhängigkeiten abzubilden. Des Weiteren haben wir das *InceptionTime*-Modell weiter optimiert und analysiert, um die Genauigkeit für die KOEX-Datensätze zu verbessern. Die Ergebnisse dieser Ablation sind in Tabelle 4 dargestellt.

	LogReg	XGBoost	DNN	RNN	Trans	Inception
Accuracy	0.6773	0.9264	0.6754	0.9396	<b>0.9551</b>	<u>0.9516</u>

Tabelle 3: Ergebnisse des Überwachten Lernens für Zeitreihen aus dem Accounting-Datensatz.

Ablation	Ohne Upsampling		Mit Upsampling	
	Accuracy	F1-Score	Accuracy	F1-Score
Normal	0.9366	0.2693	0.9516	0.0518
Focal Loss, $\gamma = 1$ , $\alpha = 0.25$	0.9551	0	0.9551	0
Bottleneck-Layer zu Identität geändert	0.9397	0.2522	0.8086	0.2495
Kernelgrößen = [3,5,7]	0.9544	0.0344	0.9119	0.3092
Residual-Verbindung entfernt	<b>0.9406</b>	<b>0.3146</b>	0.8658	0.2659
Aktivierungsfunktion ELU $\rightarrow$ LeakyReLU	0.9359	0.3137	0.9425	0.2413
Batch-Norm entfernt	0.9424	0.2136	0.9454	0.2044
Anzahl der Filter auf 64 geändert	0.9497	0.1081	0.9526	0.0567
Dritter Inception-Block entfernt	0.9512	0.0204	0.8942	0.1822
Maxpool durch Avgpool ersetzt	0.9410	0.275	0.9298	0.1039
Gewichtungszerrfall auf $10^{-4}$ angepasst	0.9551	0	0.908	0.2054

Tabelle 4: Ablation-Studienergebnisse für Genauigkeit und F1-Score (Ohne Upsampling vs. Mit Upsampling),

## 5 Teilprojekt 5: Entwicklung Cloud-Solution

Im Rahmen der Entwicklung der Cloud-Solution hat die Universität Hildesheim bei der Entwicklung sowie Optimierung der Modelle des maschinellen Lernens zur Arbeit beigetragen. Detaillierte Ausführungen zu diesem Teilprojekt finden sich im Sachbericht Teil 2 der prenode GmbH.

## 6 Teilprojekt 6: Entwicklung Demonstrator und Evaluation

Die Universität Hildesheim hat die SIVIS GmbH im Rahmen der Präsenzworkshops sowie durch weiteren Austausch bei der Erstellung von Mock-ups des Demonstrators unterstützt. Die Hauptentwicklung des Front-Ends liegt in der Verantwortung der SIVIS GmbH, und wir verweisen hiermit auf den Sachbericht Teil II der SIVIS GmbH. Des Weiteren wurden Experimente zur Evaluation von verschiedenen Ansätzen im Bereich des maschinellen Lernens erprobt. Hier lag der Fokus der Hochschule Karlsruhe auf einer erweiterten Auswahl von Datensätzen, jedoch ausschließlich für die Entscheidungsbaummodelle. Die SIVIS GmbH fokussierte sich auf das final im Demonstrator genutzte Datenmodell und den Algorithmus bei der Evaluation.

## 7 Zahlenmäßiger Nachweis

Die finanziellen Ausgaben der Universität im Rahmen des KOEX-Projektes sind Tabelle 5 zu entnehmen. Es wurden insgesamt 181.002,49 € für die Finanzierung eines Doktoranden

<b>Ausgaben</b>	entstandene Ausgaben	Ausgaben laut Zuwendungsbescheid
<b>812 Beschäftigte E12-E15</b>	181.002,49 €	179.544,29 €
<b>0846 Dienstreisen</b>	1.145,55 €	8.710,00 €
<b>Gesamtausgaben:</b>	182.148,04 €	188.254,29 €

Tabelle 5: Übersicht über die Ausgaben der Universität Hildesheim im Rahmen des KOEX-Projektes.

sowie einer wissenschaftlichen Hilfskraft ausgegeben.

Des Weiteren fielen 1.145,55 € für Dienstreisen an. Diese betrafen die drei Präsenzworkshops in Karlsruhe, an denen alle Projektpartner teilgenommen haben. Insgesamt hat die Universität Hildesheim 182.148,04 € gegenüber den 188.254,29 € im Zuwendungsbescheid genehmigten Ausgaben verwendet. Damit ergibt sich eine Differenz von 6.106,25 € an ungenutzten Mitteln. Diese Einsparungen ergaben sich insbesondere bei den Dienstreisen durch die Nutzung privater Unterkünfte, die frühzeitige Planung und Buchungen sowie durch ausstehende Publikationen und damit verbundene Konferenzbesuche.

## 8 Nutzen und Verwertbarkeit

Im Rahmen des KOEX-Projektes hat die Universität Hildesheim tiefe Einblicke in die Analyse von ERP-Daten gewonnen. Insbesondere die starke Performance von Gradient Boosted Decision Trees im Vergleich zu Modellen, die auf dem Attention-Mechanismus basieren, ist sehr interessant, da diese nur im Bereich von tabellarischen Daten der Fall ist und nicht allgemeingültig für das Feld des maschinellen Lernens. In diesem Bereich werden weitere Nachforschungen betrieben, und aktuell ist ein Paper zu den Erkenntnissen in Arbeit. Auch bei der Erstellung von synthetischen Datensätzen wurden neue Kompetenzen gewonnen und die Universität plant die automatische simulation von Betrugsfällen mit LLM zu untersuchen. Auch für die Lehre an der Universität Hildesheim konnte das KOEX Projekt verwendet werden um die Erkenntnisse im Bereich der Anwendungen von maschinellem Lernen in den Bereichen von tabellarischen Daten, Zeitreihen und Fraud-Detection zu stärken sowie diese in zwei Masterarbeiten auszugeben. Des Weiteren haben Verhandlungen begonnen, weitere Partner für die Betrugserkennungs-Software zu integrieren und die Modelle auf den Daten dieser Branche zu evaluieren.

## 9 Bekannt gewordener Fortschritt bei anderen Stellen

Ergebnisse aus den Bereichen „Betrugserkennung“, „Anomalieerkennung“, „Zeitreihenklassifikation“, „Klassifikation von Tabellendaten“ und „Big Data Analyse“ werden regelmäßig auf Konferenzen und Workshops vorgestellt. Nach aktuellem Wissensstand existieren keine konkurrierenden Ergebnisse zu den im KOEX-Projekt erreichten Zielen. Das in den Zwischenberichten 2022 und 2023 untersuchte DeepScan-Projekt der Universität Würzburg wurde weiterhin untersucht, und keine Konkurrenz zum KOEX-Projekt festgestellt.

# Literaturverzeichnis

- [An and Cho, 2015] An, J. and Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18.
- [Arik and Pfister, 2021] Arik, S. Ö. and Pfister, T. (2021). Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687.
- [Chen, 2015] Chen, T. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4).
- [Grinsztajn et al., 2022] Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520.
- [Han et al., 2022] Han, S., Hu, X., Huang, H., Jiang, M., and Zhao, Y. (2022). ADBench: Anomaly detection benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- [Huang et al., 2020] Huang, X., Khetan, A., Cvitkovic, M., and Karnin, Z. (2020). Tab-transformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*.
- [Ismail Fawaz et al., 2020] Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F. (2020). Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962.
- [LaValley, 2008] LaValley, M. P. (2008). Logistic regression. *Circulation*, 117(18):2395–2399.
- [Li et al., 2022] Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C., and Chen, G. H. (2022). Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12181–12193.
- [Medsker et al., 2001] Medsker, L. R., Jain, L., et al. (2001). Recurrent neural networks. *Design and Applications*, 5(64-67):2.
- [Peterson, 2009] Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883.
- [Schlegl et al., 2019] Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G., and Schmidt-Erfurth, U. (2019). f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54:30–44.

[Schnepf et al., 2023] Schnepf, J., Scheuermann, B., and Vetter, P. (2023). Analyzing data sets for ml-driven fraud detection in sap systems. In *2023 IEEE International Conference on Big Data (BigData)*, pages 3314–3324. IEEE.