DEUTSCHE BUNDESBANK EUROSYSTEM

AnoMATE: Mixed-type Tabular Embeddings for Anomaly Detection

Technical Report 2026-03

Deutsche Bundesbank, Research Data and Service Centre

Timur Sattarov
Robin Baudisch
Georg Steinbuß

Research Data and Service Centre

Abstract

Large-scale microdata have become foundational for central banks and financial supervisors. However, detecting anomalies in these datasets remains challenging due to their high volume, complex feature relationships, and mixed-type variables. Traditional categorical transformations, such as one-hot encoding, often fail in financial contexts where features exhibit high cardinality.

In this work, we propose a unified autoencoder-based framework that handles heterogeneous feature types while preserving their statistical properties. By utilizing entity embeddings for categorical encoding, our model significantly reduces training time and model size while improving detection performance compared to conventional techniques. We empirically validate our framework on three open-source datasets and one proprietary (AnaCredit) financial dataset.

Furthermore, the model provides explainability by decomposing global anomaly scores into individual reconstruction errors and expected feature values, allowing practitioners to identify the drivers of detected irregularities. To facilitate the exploration of these results, the framework includes an integrated dashboard designed for visualizing and analyzing anomalies in real-time. The entire system is implemented as a comprehensive Python package, covering the full pipeline from preprocessing to post-hoc analysis. The proposed framework acts as a practical utility for supervisors, bridging the gap between raw financial data and actionable regulatory insights.

Usage of AI: This work was developed with the assistance of generative AI (GenAI) tools. GenAI was used for brainstorming ideas, generating outlines, drafting sections, editing for clarity, identifying relevant literature, and coding. All GenAI-generated content was reviewed and, where necessary, revised by the authors, who remain responsible for the accuracy, originality, and integrity of the final work.

Keywords: anomaly detection, mixed-type tabular data, autoencoder neural networks, categorical embeddings

Citation: Sattarov, T., Baudisch, R., and Steinbuß, G. (2026). AnoMATE: Mixed-type Tabular Embeddings for Anomaly Detection, Technical Report 2026-03. Deutsche Bundesbank, Research Data and Service Centre.[1)]

1 We thank the members of the department Data and Statistics at the Deutsche Bundesbank for their valuable review and remarks

Contents

1 Introduction

Large-scale tabular microdata, such as AnaCredit [2], EMIR [3], SFTR [4] or MiFID II [5], have become fundamental in modern central banking, facilitating robust micro- and macroprudential supervision. Ensuring high data quality is essential for the reliability of downstream analyses in monetary policy, financial stability, and systemic risk assessment. In this context, robust anomaly detection is a key instrument for central banks, serving a twofold purpose:

- **Detection of Reporting Errors:** Automated anomaly detection enhances data quality and integrity by identifying reporting inaccuracies within massive datasets. Identifying anomalous attributes in each record enables granular explainability, making clear which variables are responsible for the anomaly and in what sense the observation violates regulatory expectations.
- **Downstream Analytical Applications:** Beyond error correction, systematic exploration of true economic anomalies improves the reliability of econometric and statistical models. Identifying such anomalies helps reveal emerging macroeconomic trends, structural shifts, or systemic risks that may warrant policy intervention.

However, designing such anomaly detection systems is challenging because tabular microdata are characterized by two main sources of complexity: (i) massive observation volumes, often exceeding millions of records, and (ii) high-dimensional, *mixed-type* feature spaces. Crucially, these features combine numerical, boolean, and temporal variables with categorical data that often exhibits high cardinality. This mixed-type nature is not a side aspect but a defining characteristic of modern supervisory data. For instance, according to the AnaCredit Regulation [29], such dataset comprises, among others, the following types of features:

- categorical features (e.g. 'Type of instrument', 'Currency'),
- numerical features (e.g. 'Outstanding nominal amount'),
- boolean features (e.g. 'Default status of the instrument').
- temporal features (e.g. 'Inception date').

Many classical anomaly detection algorithms are designed primarily for purely numerical data. As a result, these methods struggle to model the joint distribution of heterogeneous attributes in a principled way, especially when the data are both *large* and *mixed-type*. The core challenge is to develop scalable anomaly detection systems that natively integrate mixed-type features, rather than treating them as secondary preprocessing steps. Autoencoders are a flexible class of neural network models that have become widely used in the anomaly detection literature, including applications in finance [36, 35, 32, 33]. Their suitability for supervisory microdata stems from three key properties:

1. **High-dimensional scalability:** Neural networks efficiently process millions of records and high-dimensional spaces. This allows autoencoders to exploit the full depth of modern supervisory datasets without the performance bottlenecks of traditional methods.
2. **Unified latent representations:** Autoencoders map mixed-type data into a shared latent space, learning complex inter-dependencies simultaneously. This shared space is further optim-

2 Analytical Credit Dataset, Regulation (EU) 2016/867
3 European Market Infrastructure Regulation, Regulation (EU) No 648/2012
4 Securities Financing Transactions Regulation, Regulation (EU) 2015/2365
5 Markets in Financial Instruments Directive, Directive 2014/65/EU

ized via specialized output heads and loss functions to ensure accurate reconstruction across all feature types.
3. **Interpretable anomaly scoring:** The reconstruction error provides both a natural anomaly score and a direct path to interpretability. By comparing inputs to reconstructions, one can pinpoint which specific features contribute to an anomaly and determine their expected values.

Together, these properties make autoencoders ideal candidate for the supervisory context, where models must simultaneously provide scalability, support for mixed-type data, and transparent anomaly scoring. Furthermore, their ability to reconstruct input features provides an inherent mechanism for contribution analysis, allowing supervisors to identify which specific variables triggered an anomaly. This interpretability ensures that model outputs are not merely black-box signals, but actionable insights that can be cross-referenced with institutional reporting.

This work proposes **AnoMATE** (**Ano**maly **M**ixed-type **A**utoencoding **T**abular **E**mbedding), an autoencoder-based framework for anomaly detection in mixed-type tabular data, with a particular focus on supervisory datasets such as AnaCredit. The main contributions are as follows:

- **End-to-end Python implementation:** We provide a complete Python package for anomaly detection on mixed-type tabular data. The package automates the transition from raw tabular data to interpretable anomaly scores. This includes robust preprocessing routines and a heuristic for latent space dimensionality, enhancing the deployment of deep autoencoders in diverse supervisory contexts.
- **Embedding strategy for high-cardinality data:** We introduce a native embedding approach for categorical features that avoids the sparsity one-hot encoding. By mapping categorical codes and numerical attributes into a shared latent space, the framework captures complex dependencies across mixed-type variables while remaining computationally efficient.
- **Empirical evaluation and benchmarking:** We validate the proposed framework against current state-of-the-art anomaly detection methods on three public mixed-type datasets and one proprietary dataset. The results demonstrate the robustness of our architecture in identifying anomalies within high-dimensional, heterogeneous datasets.

2 Related Work

Anomaly detection in tabular data has attracted sustained attention from both the machine learning and data mining communities, driven by applications in fraud detection, error identification, and quality assurance. This section reviews the landscape organized along three lines: conventional methods, deep learning models, and tabular foundation models.

2.1 Conventional Models for Tabular Anomaly Detection

Classical anomaly detection methods can be broadly categorized into density-based, distance-based, and isolation-based approaches. The Local Outlier Factor (LOF) [3] quantifies anomalousness via local density deviations among k-nearest neighbors, while One-Class Support Vector Machines (OC-SVM) [34] learn a decision boundary enclosing normal data in a kernel-induced feature space. Isolation Forest [23] introduced a fundamentally different paradigm based on recursive random partitioning, exploiting the observation that anomalies require fewer splits to isolate; its extension to non-axis-aligned hyperplanes [12] addresses biases from axis-parallel cuts. Tree-based ensemble methods, XGBoost [5], LightGBM [18], and CatBoost [27], have also been adapted for anomaly detection through one-class formulations or limited-label classification. The ADBench benchmark [11] evaluated 30 algorithms across 57 datasets and found that no single unsupervised method is statistically superior, while even 1% labeled anomalies suffice for semi-supervised methods to consistently outperform unsupervised approaches.

A persistent limitation of these methods is their difficulty with mixed-type tabular data. Most classical algorithms assume continuous feature spaces; categorical variables must be encoded via one-hot or ordinal schemes, which introduces sparsity, distorts distance metrics, and fails to capture semantic relationships between categories. This encoding bottleneck motivates deep learning approaches capable of learning richer representations from heterogeneous feature types.

2.2 Deep Learning Models for Tabular Anomaly Detection

Reconstruction-based methods. The premise of autoencoder-based anomaly detection is that models trained to reconstruct normal data incur higher reconstruction error on anomalies. Variational Autoencoders [19] impose probabilistic structure on the latent space for likelihood-based scoring, while the Deep Autoencoding Gaussian Mixture Model (DAGMM) [43] jointly optimizes an autoencoder and a Gaussian mixture model for density-based anomaly scoring. In the domain of financial auditing, [36] demonstrated the effectiveness of deep autoencoders for detecting anomalies in large-scale accounting data, later extending this with adversarial autoencoder networks for more interpretable latent-space anomaly localization [35]. While effective on continuous features, these approaches struggle with categorical variables. To address this, entity embeddings [10] map categorical variables into learned continuous spaces, placing semantically similar categories nearby and enabling autoencoders to process mixed-type data through a unified representation, a technique that directly informs the design of our proposed method. More recently, [33] proposed the Diffusion-Scheduled Denoising Autoencoder (DDAE), integrating diffusion-based noise scheduling and contrastive learning into the encoding process; evaluated on 57 ADBench datasets, DDAE

demonstrates that structured noise injection improves the discriminative quality of reconstruction errors.

Transformation-based methods. Deep SVDD [30] combines deep networks with the support vector data description principle, learning mappings that minimize a hypersphere enclosing normal representations. Subsequent methods include classification over geometric transformations [2], learned neural transformations scored for normality [28], and scale learning as a supervisory signal for tabular anomaly detection [42].

Attention-based architectures. TabNet [1] applies sequential attention to select salient features at each decision step, providing built-in instance-level feature importance. [39] leverage Non-Parametric Transformers to capture both feature-feature and sample-sample dependencies, achieving state-of-the-art results across 31 benchmark datasets.

Explainability in tabular anomaly detection. DIAD [4] adapts Generalized Additive Models as white-box detectors with inherent interpretability, improving AUC from 86.2% to 89.4% with as few as five labeled anomalies. RESHAPE [25] enhances Shapley-based explanations for autoencoder-detected anomalies in financial audits, demonstrating that reconstruction-driven attribution improves root cause identification. Despite this progress, most methods provide explanations as a post-hoc overlay rather than a native output of the detection model, creating an opportunity for architectures that jointly produce anomaly scores and attribute-level attributions.

2.3 Tabular Foundation Models

The emergence of transformer-based architectures has opened a new frontier for tabular data. TabTransformer [17] applies multi-head self-attention over categorical embeddings, producing contextual representations robust to missing and noisy data, though it processes numerical features separately through an MLP. The FT-Transformer [8] addressed this by tokenizing all features into a unified embedding space, achieving the best deep learning performance on a comprehensive benchmark, while showing that transformers do not consistently surpass gradient boosted trees [9]. SAINT [38] combines column attention with row attention to capture both feature interactions and inter-sample relationships.

A paradigmatic shift was introduced by TabPFN [16], which frames tabular prediction as in-context learning: trained on millions of synthetic datasets, it performs inference in a single forward pass without hyperparameter tuning, outperforming boosted trees on the OpenML-CC18 benchmark. Its successor TabPFN-2.5 [15] scales to 50,000 samples and 2,000 features. For anomaly detection specifically, ICE-T [40] applies contrastive learning to heterogeneous tabular data by treating each column as a distinct modality, learning cross-column embeddings that scale linearly with feature count.

3 AnoMATE

In this section we describe the proposed framework with data transformation steps, technical details of the model architecture and interactive dashboard.

3.1 Framework Overview

The proposed framework is organized into three primary stages: Input Transformations, Model Learning, and Output Transformations, as illustrated in Figure 1. These modules work in sequence to preprocess raw mixed-type tabular data, fit the autoencoder model to the training set, and generate standardized output scores for downstream analysis.

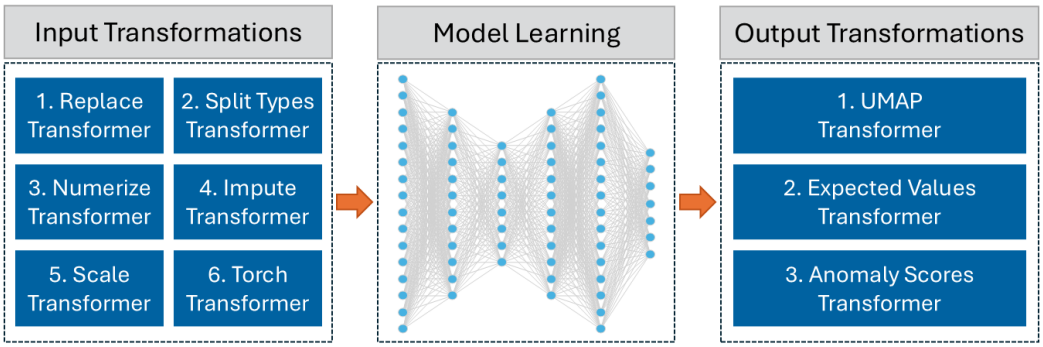

Figure 1: Schematic representation of the proposed framework. The workflow transitions from raw data preprocessing (Input Transformations) through autoencoder fitting (Model Learning) to final score preparation (Output Transformations).

Input Transformations. This component prepares the raw data for training the autoencoder through a sequence of modular preprocessing steps.

1. **Replace Transformer:** an optional step that replaces selected values in the dataset, for example to correct known errors or harmonize special codes. This step typically relies on domain knowledge to specify which values should be modified and how.
2. **Split Type Transformer:** this step partitions the input dataframe into subsets according to feature type, such as categorical, numeric, boolean, or temporal. This separation enables type-specific preprocessing and model handling.
3. **Numerize Transformer:** this step maps categorical features to integer indices required for embedding layers. Temporal features are converted to numeric values by computing the difference in days from a reference date, and boolean features (True/False) are mapped to 1/0.
4. **Impute Transformer:** this step handles missing values by imputing numeric features (e.g., with a fixed statistic or learned value) and creating an explicit "missing" category for categorical features. This ensures that the model can process incomplete records without discarding them.
5. **Scale Transformer:** this step normalizes numeric features to stabilize training and make features comparable in scale. We use a Quantile Transformer to map the distributions to a more regular form.
6. **Torch Transformer:** in the final step, all processed feature subsets are converted into PyTorch tensors. This produces a unified representation that can be directly fed into the autoencoder.

Model Learning. The model learning component trains the autoencoder on the transformed data to capture the joint structure of heterogeneous features. The architecture includes dedicated heads for different feature types and a shared latent space that encodes the underlying patterns. Training is performed in an unsupervised manner by minimizing reconstruction loss, which later serves as the basis for anomaly scoring.

Output Transformations. This component processes the outputs of the trained autoencoder to derive interpretable representations, expected values, and anomaly scores.

1. **UMAP Transformer:** this step extracts the activations from the latent layer of the autoencoder and, if the latent dimension exceeds two, reduces them to two dimensions using UMAP [24]. The resulting 2D embeddings facilitate visualization and exploratory analysis of the data structure.
2. **Expected Values Transformer:** this step converts the reconstructed outputs of the autoencoder back to the original feature space. Categorical reconstructions are mapped from predicted codes to category labels, while numeric and temporal features are inverse-transformed to their original scales.
3. **Anomaly Scores Transformer:** this step computes anomaly scores at both the feature and instance level based on reconstruction errors. Local scores highlight which features contribute most to an anomaly, and global aggregated scores summarize the overall degree of abnormality for each sample.

3.2 Model Architecture

The following section provides a technical description of the proposed architecture. We detail the feature encoding and decoding stages, the embedding learning strategy for categorical attributes, and the specific architectural configuration designed for mixed-type tabular data.

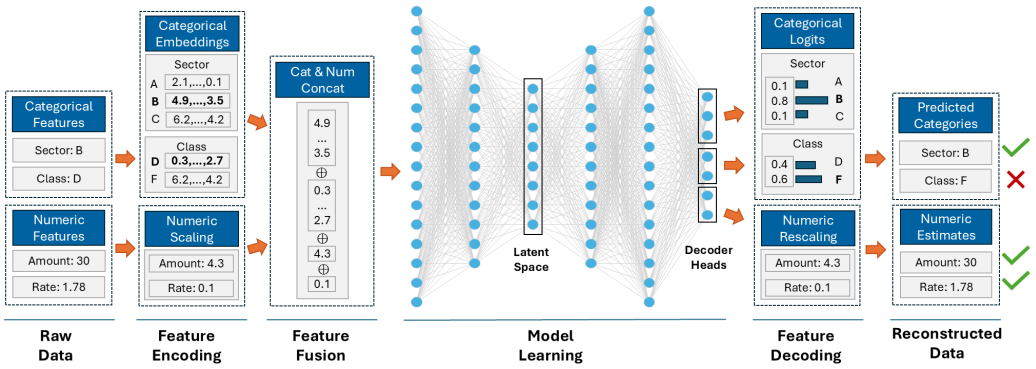

Figure 2: Schematic overview of the proposed architecture, illustrating categorical embedding layers, the latent learning process, and the feature decoding pipeline.

We use *Autoencoder* [14] neural network as the basis of our architecture. It learns compact data representations by training to reproduce its input. It is composed of an encoder–decoder pair: the encoder f_θ maps an input $\mathbf{x} \in \mathbb{R}^d$ to a lower-dimensional latent code $\mathbf{z} \in \mathbb{R}^k$ with $k < d$, while the decoder g_ϕ maps this code back to a reconstruction $\hat{\mathbf{x}} = g_\phi(\mathbf{z})$. The dimensionality bottleneck in the latent space encourages the model to retain only the most informative characteristics of

the data. Training proceeds by minimizing the mean squared reconstruction error between the input and its reconstruction $\mathcal{L}_{\text{rec}} = \mathbb{E}_{\mathbf{x}} \left\| \mathbf{x} - g_\phi(f_\theta(\mathbf{x})) \right\|^2$, where θ and ϕ are the parameters of the encoder and decoder, respectively. For anomaly detection, the autoencoder is trained on normal data so that it models their underlying distribution; samples that deviate from this distribution typically yield larger reconstruction errors and can thus be flagged as anomalies.

Categorical Embeddings. For high-cardinality categorical features common in financial datasets, one-hot encoding creates sparse representations that limits model expressiveness, increases memory overhead, and slows training. We instead utilize a learned *embedding layer* to map categories into a dense, continuous space. This layer maps discrete categories into a dense, continuous vector space via an embedding matrix $\mathbf{E} \in \mathbb{R}^{C \times m}$, where $m \ll C$. Each category i is represented by a vector $\mathbf{v}_i \in \mathbb{R}^m$, capturing latent semantic similarities that enhance autoencoder performance compared to sparse methods. As depicted in Figure 2 these dense representations, produced in the *feature encoding* step, are concatenated with the numerical features afterwards in the *feature fusion* step. Final representaion is used as an input to the autoencoder. Once the model is trained, the *feature decoding* step is utilized to map the softmax logits of categorical reconstructions to their original categories by selecting the index with the highest probability.

Multi-Head Decoder. To address tabular heterogeneity, we employ a multi-head decoder that partitions the reconstruction layer into feature-specific separate heads. All heads share the same latent representation but use type-specific output layers and loss functions. Such design allows the model to reconstruct numerical, categorical, boolean, and temporal features in a type-consistent way.

- **Numerical:** Reconstructed via a dense linear layer with N_{num} outputs. We minimize the Mean Squared Error (MSE) on normalized targets to maintain consistency with the input scaling.
- **Categorical:** Processed through N_{cat} independent dense layers, where each layer i produces K_i logits for its respective feature. A softmax activation generates class probabilities, optimized via categorical Cross-Entropy (CE).
- **Boolean:** Handled by a dedicated dense layer with N_{bool} outputs. A sigmoid activation is applied to each unit and optimize the reconstruction using binary cross-entropy against the ground-truth targets.
- **Temporal:** Mapped through a linear layer with N_{temp} outputs representing scaled time-based values (e.g., days since a reference date). We apply an MSE loss to these outputs, which are later inverse-transformed to recover the original time domain.

Loss Function. To address the heterogeneous nature of tabular data, we define a composite reconstruction loss that accounts for varying feature types. For a given sample $\mathbf{x}$, the total loss $\mathcal{L}_{\text{total}}$ is decomposed into specific error metrics tailored to numerical ($\mathbf{x}_{\text{num}}$), categorical ($\mathbf{x}_{\text{cat}}$), boolean ($\mathbf{x}_{\text{bool}}$), and temporal ($\mathbf{x}_{\text{temp}}$) components:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}}(\mathbf{x}_{\text{num}}, \hat{\mathbf{x}}_{\text{num}}) + \mathcal{L}_{\text{CE}}(\mathbf{x}_{\text{cat}}, \hat{\mathbf{x}}_{\text{cat}}) + \mathcal{L}_{\text{BCE}}(\mathbf{x}_{\text{bool}}, \hat{\mathbf{x}}_{\text{bool}}) + \mathcal{L}_{\text{MSE}}(\mathbf{x}_{\text{temp}}, \hat{\mathbf{x}}_{\text{temp}}), \tag{1}$$

where $\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_{\text{CE}}$, and $\mathcal{L}_{\text{BCE}}$ denote Mean Squared Error, Cross-Entropy, and Binary Cross-Entropy,

respectively.

Latent Space Compression. To optimize the latent representation for anomaly detection, we determine the bottleneck dimension d by estimating the **Intrinsic Dimension (IntrDim)** of the data manifold. We employ the **Two-NN algorithm** [6], which utilizes the ratio of distances to the two nearest neighbors, $\mu_i = r_{i,2}/r_{i,1}$. Under the assumption of local homogeneity, the cumulative distribution follows $P(\mu) = 1 - \mu^{-d}$. The global intrinsic dimension is computed directly as:

$$d = -\frac{\sum_{i=1}^{n} \ln(1 - \hat{P}(\mu_i))}{\sum_{i=1}^{n} \ln(\mu_i)} \tag{2}$$

where $\hat{P}(\mu_i)$ represents the empirical cumulative distribution. We set the autoencoder's latent dimension to $\lceil d \rceil$ to ensure sufficient capacity for normal patterns while restricting the model from learning the identity mapping.

Feature Anomaly Scores. In addition to providing a single anomaly score per instance, our method performs *feature-wise reconstruction* to derive localized anomaly indicators. For each feature (column), we compute an individual reconstruction error, using squared error for numerical variables and negative log-likelihood for categorical variables, resulting in a detailed error vector for every transaction. This per-feature error decomposition naturally functions as an explanation mechanism, highlighting which attributes contribute most strongly to the detected anomaly.

Feature Expected Values. The decoder output $\hat{\mathbf{x}}$ also provides *expected values* for each feature, i.e., the model's estimate of what a typical (normal) observation would look like given the latent representation of the input. For numerical features, we directly use the reconstructed value as the expected value. For categorical features, we take the category corresponding to the highest reconstructed logit (i.e., the argmax over the predicted class scores) as the expected value. Comparing the original input $\mathbf{x}$ to these reconstructed expectations quantifies the deviation of each feature from the learned notion of normality. This contrast offers a transparent reference point for interpreting how an anomalous record diverges from the model's internal concept of legitimate behavior, while the probabilistic outputs for categorical variables additionally enable uncertainty-aware explanations.

3.3 Interactive Dashboard

To support inspection and interpretation of model outputs, we provide an interactive dashboard with three complementary views (Figure 3). These views enable analysis at the global, instance, and neighborhood levels, allowing users to move from high-level anomaly ranking to detailed case analysis and latent-space exploration.

- **Global Anomaly Ranking.** This view presents the top-(N) observations ranked by their global anomaly scores. Users can hover over individual points to access meta-information (e.g., feature values, identifiers), load additional anomalies, and apply filters based on selected features. It is designed to support quick screening and prioritization of anomalous cases.

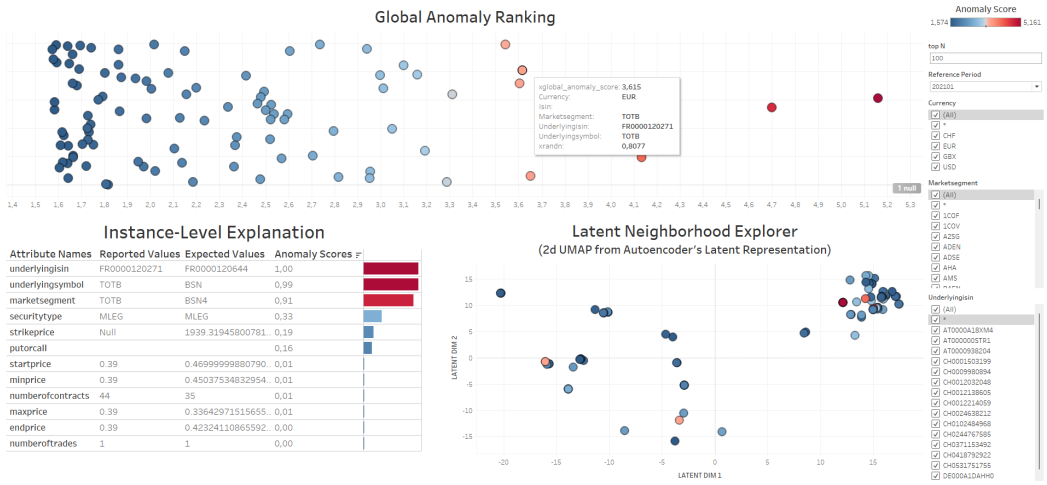

Figure 3: Interactive anomaly analysis dashboard with three coordinated views: *Global Anomaly Ranking* (top row), showing the top 100 datapoints by anomaly score; *Instance-Level Explanation* (bottom left), detailing observed values, expected values, and feature-wise anomaly scores for a selected point; and *Latent Neighborhood Explorer* (bottom right), displaying a 2D UMAP projection of the latent representations for these 100 datapoints.

- **Instance-Level Explanation.** This view provides a detailed breakdown for a selected anomalous observation. For each feature, it displays the *reported values* (input), the *expected values* from the model reconstruction, and the corresponding feature-wise *anomaly scores*. This enables users to identify which features drive the anomaly and how the model's expectations deviate from the observed data.
- **Latent Neighborhood Explorer.** This view shows a two-dimensional UMAP projection of the latent representations, focusing on the top-(N) anomalies and their neighboring observations. It supports the analysis of groups of anomalies with similar latent patterns, as well as clusters of normal observations and their relationships. This view is useful for identifying anomaly subtypes and understanding the global structure of the data in the learned latent space.

4 Benchmarks

In this section we describe the experimental settings and present the corresponding results used to evaluate the proposed model.

4.1 Experimental Setup

Datasets. To evaluate the proposed anomaly detection framework, we utilized three publicly available datasets and one proprietary dataset covering financial tabular domain. These datasets were selected to represent diverse anomaly scenarios, feature sets, and degrees of class imbalance.

- **Vehicle Insurance**[6] Focused on insurance claims, this dataset includes records describing policy holders, vehicles, and accident details. It presents a significant class imbalance, with fraudulent claims making up roughly 6% of the total entries.
- **Fraud E-commerce**[7] This dataset contains simulated e-commerce transactions capturing user behavior and device metadata. It is primarily used to analyze fraud patterns linked to anomalies between account creation and initial purchase.
- **IEEE-CIS Fraud Detection**[8] This large-scale benchmark contains real-world e-commerce transactions provided by Vesta Corporation. The data features high dimensionality and extreme imbalance, with 3.5% of transactions labeled as fraudulent.
- **AnaCredit German Subset**[9] This dataset provides highly granular, loan-by-loan information on credit instruments granted by German financial institutions to legal entities. It features a multidimensional structure with over 90 attributes covering counterparty characteristics, loan types, and credit risk indicators. The final dataset is constructed by preprocessing and joining several underlying tables at the instrument level. Target labels are defined by validation errors derived from internal, rule-based data validation processes.

Table 1: Summary of Dataset Statistics

		#columns				
Dataset	**#rows**	**num**	**cat**	**bool**	**temp**	**%anomalies**
Vehicle Insurance	15,420	8	24	0	0	5.99
Fraud E-commerce	151,112	2	6	0	0	9.37
IEEE-CIS	590,540	61	6	0	0	3.50
AnaCredit	16,848,299	21	34	58	10	2.80

Baselines. To evaluate the efficiency of our proposed method, we compare it against twelve established outlier detection models across four categories. Statistical baselines include PCA [37], HBOS [7], and the parameter-free cumulative distribution methods COPOD [21] and ECOD [22]. We include proximity-based approaches LOF [3] and CBLOF [13], alongside ensemble methods IForest [23], FeatBagging [20], and the lightweight LODA [26]. Finally, we compare against deep learning architectures, including standard AutoEncoder [14], VAE [19], and DeepSVDD [31]. For all baseline models, missing values in categorical features were imputed by introducing a new

6 The dataset is available at: https://www.kaggle.com/datasets/khusheekapoor/vehicle-insurance-fraud-detection
7 The dataset is available at: https://www.kaggle.com/vbinh002/fraud-ecommerce
8 The dataset is available at: https://www.kaggle.com/c/ieee-fraud-detection/overview
9 In compliance with strict data privacy regulations, content of the dataset cannot be made publicly available.

category before applying a one-hot encoding. For numeric features, missing data were imputed using the mean strategy, followed by a quantile transformation.

Evaluation Setting. We adopt a semi-supervised evaluation setup. For the public datasets, the normal samples are split equally, with 50% allocated to the training set and the remaining 50% reserved for the test set alongside the anomalies. This data partitioning is repeated across five independent runs using different random seeds, and the performance metrics are averaged. For the AnaCredit dataset, which naturally follows a temporal split, the training set consists of data from February 2026 (approximately 8 million records, excluding anomalies), while the evaluation set comprises data from March 2026 (approximately 8 million records, including anomalies).

Evaluation Metrics. To measure performance, we use PR-AUC (Precision-Recall AUC). We prioritize PR-AUC as it provides a more reliable assessment than common metrics like accuracy due to the extreme class imbalance of the labels. All results are reported as the mean ± standard deviation across five runs with different random seeds.

4.2 Experimental Results

Comparison against Baselines. We benchmarked AnoMATE against a diverse set of classical and deep anomaly detection methods on four tabular datasets. Our model is also compared against existing LLM-based approaches, with baseline scores sourced directly from the original paper [41]. Due to the scale of the supervisory datasets, several baseline models failed to produce results. Specifically, certain methods were terminated after exceeding a 24-hour runtime (T/O), while others encountered out-of-memory (OOM) errors due to high-cardinality categorical features.

Across datasets, AnoMate achieves the best performance on Vehicle, IEEE, and AnaCredit datasets with PR-AUC scores of 0.144, 0.275, and 0.353, respectively, outperforming both classical and deep baselines. On the Fraud dataset, IForest attains the highest PR-AUC (0.344), while AnoMATE reaches 0.254 and is competitive with several classical methods. We attribute this performance to the linear separability of the Fraud dataset, a characteristic further evidenced by the high scores achieved by several classical methods. Deep baselines generally perform strongly on IEEE and AcaCredit (e.g., AutoEncoder at 0.269 and 0.281 respectively), but AnoMATE still provides a clear margin over them on this dataset. Variability across runs is modest for most methods, indicating stable training and evaluation.

These results indicate that AnoMATE is particularly effective on datasets with richer feature interactions and non-linear patterns, such as Vehicle, IEEE, and AnaCredit. On these datasets, it consistently outperforms both classical and deep baselines, suggesting that its architecture can better capture complex decision boundaries. In contrast, its weaker performance on Fraud suggests that tree-based ensembles like IForest may be better suited to that dataset, possibly because they handle sparse or highly skewed features well. Overall, the strong gains suggest that the design choices in AnoMATE can provide practical benefits, while also highlighting that the most suitable anomaly detector depends on the data characteristics.

Embeddings vs. One-Hot Encodings. We next conducted a study to compare two ways of handling categorical features in an AutoEncoder: (i) standard one-hot encoding and (ii) learned

Table 2: PR-AUC scores of various anomaly detection models. Bold values indicate best performance. OOM denotes Out-of-Memory; T/O denotes a Time-out (exceeded 24h)

	Datasets			
Model	**Vehicle**	**Fraud**	**IEEE**	**AnaCredit**
Classical Models				
CBLOF [13]	0.117 ± 0.005	0.186 ± 0.001	0.250 ± 0.020	0.057 ± 0.010
COPOD [21]	0.119 ± 0.001	0.184 ± 0.000	0.230 ± 0.012	OOM
ECOD [22]	0.121 ± 0.001	0.184 ± 0.000	0.238 ± 0.001	OOM
FeatBagging [20]	0.132 ± 0.002	0.294 ± 0.005	0.107 ± 0.010	T/O
HBOS [7]	0.125 ± 0.001	0.271 ± 0.002	0.241 ± 0.000	0.101 ± 0.023
IForest [23]	0.129 ± 0.002	**0.344 ± 0.035**	0.242 ± 0.027	0.232 ± 0.092
LODA [26]	0.132 ± 0.013	0.238 ± 0.033	0.083 ± 0.005	0.073 ± 0.024
LOF [3]	0.126 ± 0.002	0.268 ± 0.003	0.111 ± 0.002	T/O
PCA [37]	0.120 ± 0.001	0.173 ± 0.000	0.103 ± 0.002	OOM
Deep Learning Models				
AutoEncoder [14]	0.136 ± 0.002	0.300 ± 0.019	0.269 ± 0.020	0.281 ± 0.000
VAE [19]	0.124 ± 0.001	0.221 ± 0.001	0.100 ± 0.001	0.051 ± 0.000
DeepSVDD [31]	0.124 ± 0.002	0.191 ± 0.001	0.120 ± 0.056	0.173 ± 0.001
AnoMATE (ours)	**0.144 ± 0.005**	0.254 ± 0.001	**0.275 ± 0.002**	**0.353 ± 0.030**
LLM Models				
SmolLM-360M[41]	0.143 ± 0.001	—	—	—

**Scores are derived from the averaged results and standard deviations of five experiments, each initiated with distinct random seeds*

embeddings. We evaluated three aspects: model size (number of trainable parameters), training time, and detection performance measured by PR-AUC. All other architectural and training settings were kept identical.

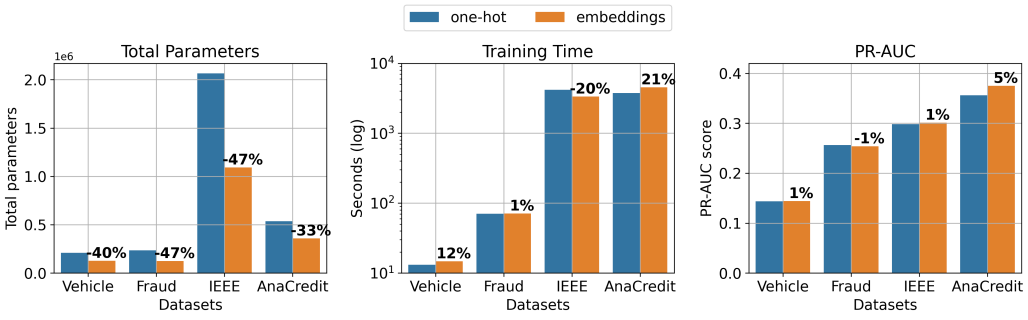

Figure 4: Model parameter reduction (left), total training time (middle), and PR-AUC scores (right) across various datasets. Percentages indicate the reduction achieved by the proposed method compared to the baseline.

Using embeddings led to a substantial reduction in model size across all datasets. On Vehicle, the number of parameters decreased by 40%, while on Fraud and IEEE the reduction was 47%. Despite this compression, the PR-AUC scores remained almost unchanged on three datasets, with 5% improvement on anacredit dataset, indicating that the more compact models did not sacrifice anomaly detection performance.

The effect on training time was mixed and depended on the vocabulary size of the categorical features. On Vehicle, Fraud and AnaCredit, training with embeddings was slightly slower (+12%, +1%, and +21% respectively), which can be attributed to the overhead of embedding lookups when the categorical vocabularies are relatively small (fewer than 200 categories). In contrast, on the IEEE dataset with vocabulary size of 1916, embeddings yielded a 20% reduction in training time, as the cost of one-hot representations grows with vocabulary size while the embedding lookup overhead becomes negligible. These findings suggest that embeddings are particularly advantageous for datasets with larger categorical vocabularies, offering substantial parameter savings and potential speedups without degrading detection quality.

Effect of Categorical Vocabulary. We further analyzed the effect of categorical vocabulary size (total number of unique codes across all categorical features) on the efficiency of using embeddings versus one-hot encodings. To this end, we constructed a synthetic dataset with 5 numerical and 5 categorical features, and systematically varied the vocabulary size of the categorical features across {50, 100, 200, 500, 1000, 2000, 5000, 10000}. For each vocabulary size, we trained AnoMATE with either one-hot or embedding-based representations. We then compared the total number of trainable parameters and the training time for both configurations.

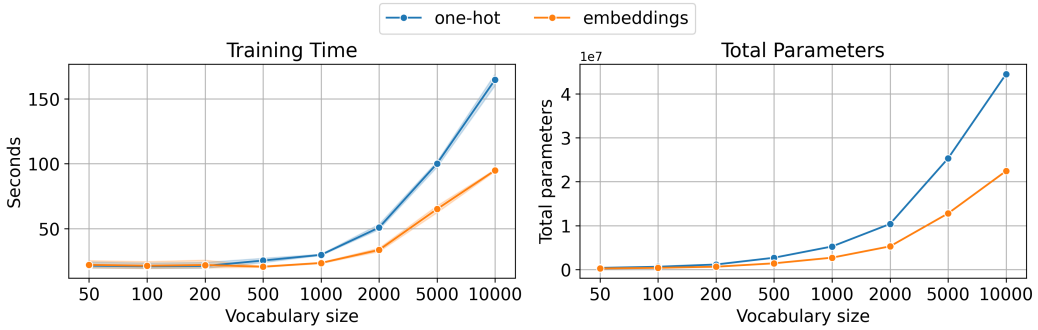

Figure 5: Performance scaling of ohe-hot encoding vs. embeddings on synthetic dataset. Total training time (left) and total number of model paramenters (right).

The results show that for small vocabularies (below 200 categories), the two approaches behave almost identically in terms of both model size and training time. Starting from a vocabulary size of 200, the embedding-based models begin to show clear advantages. As the vocabulary size increases further, the reduction in the number of parameters with embeddings becomes substantial, and the training time also decreases relative to one-hot encodings. The gap between the two representations widens monotonically with vocabulary size.

These findings provide a controlled confirmation of the trends observed on real datasets. They indicate that embeddings become increasingly beneficial as the categorical vocabulary grows, both in terms of memory footprint and computational cost. From a practical standpoint, this suggests that for models operating on high-cardinality categorical features, embedding layers are a more scalable choice than one-hot encodings, while for very small vocabularies the simpler one-hot representation remains competitive.

Latent Space Compression. This experiment evaluated the use of Intrinsic Dimension (IntrDim) as a heuristic for determining the optimal latent space dimensionality. We assessed anomaly detection performance across a range of latent dimensions {2, 4, 8, 16, 32, 64, 128, 256, 512} and compared these results against IntrDim estimates.

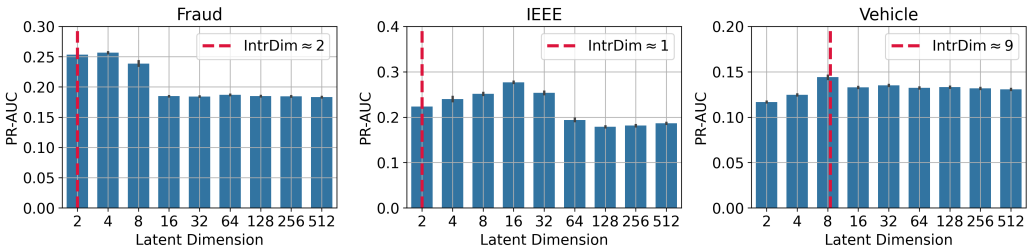

Figure 6: Evaluation of the optimal latent space compression evaluated across [2, 4, 8, 16, 32, 64, 128, 256, 512] dimensions. The red dash line reflects the estimated Intrinsic Dimension.

The results demonstrate a strong correlation between IntrDim and optimal model architecture. On Vehicle and Fraud datasets IntrDim yielded values of 9 and 2, respectively, aligning closely with the best results from our grid search. On IEEE dataset IntrDim suggested a dimension of 1, while the model peaked at 16. However, due to the dataset's size, IntrDim was computed on a subsample. This highlights a primary limitation: the quadratic complexity of computing all pairwise distances makes the heuristic less feasible for large-scale datasets.

Overall, this heuristic serves as a reliable estimate for autoencoder latent dimensionality. Given that the bottleneck size is a critical factor in anomaly detection performance, leveraging IntrDim can significantly reduce tuning time and enhance model productivity.

5 Limitations and Future Work

In this section, we discuss the main limitations of the current approach and outline directions for future work. Where appropriate, we also indicate potential remedies or extensions.

5.1 Limitations

- **Limited temporal modeling.** Temporal information is currently incorporated only via simple transformations. As a consequence, the model does not fully exploit sequential dependencies or dynamic patterns over time, which can be crucial in many real-world anomaly detection tasks.
- **Hyperparameter sensitivity.** The choice of embedding dimensionality and the design of the model architecture (e.g., number of layers, hidden sizes, and activation functions) can have a significant impact on performance. These architectural and hyperparameter choices may require careful tuning for each new dataset or application.
- **Dependence on normal data.** The method assumes access to a sufficiently large and representative set of (normal) non-anomalous observations for training. If the training data is scarce, heavily imbalanced, or contaminated with a substantial fraction of anomalies, the quality of the learned representation and the resulting anomaly scores may degrade.
- **Computational resource requirements.** Although embedding-based encoding enhances efficiency, GPU acceleration remains essential for viable training times. Current hardware constraints limit simultaneous processing to only several reference months of the AnaCredit dataset, while high-dimensional UMAP visualizations introduce further bottlenecks. These requirements may restrict scalability for institutions without dedicated hardware acceleration.
- **Score-based output only.** The current model produces a continuous anomaly score rather than a binary classification label. While this is often desirable in anomaly detection settings, it requires domain expertise to select an appropriate threshold in order to obtain binary decisions.

5.2 Future Work

- **Advanced Temporal Modeling.** A promising direction for future research involves enhancing the framework's capacity to model complex temporal dynamics. Rather than relying solely on point-in-time transformations, future iterations could integrate:
  - **Feature Engineering:** Incorporating lagged variables and rolling-window statistics to capture local historical context.
  - **Sequence Architectures:** Utilizing Recurrent Neural Networks (RNNs), Temporal Convolutional Networks (TCNs), or Transformers to model long-range temporal dependencies.
  - **Decomposition Techniques:** Explicitly isolating seasonality and trend components to improve signal-to-noise ratios.

  These extensions would enable the model to better characterize non-stationary behaviors and improve detection sensitivity in highly dynamic financial environments.
- **Performance and scalability improvements.** Future work will focus on improving both runtime and memory efficiency. One promising direction is to employ GPU-accelerated dimensionality reduction techniques, such as CUDA-enabled UMAP, to speed up the embedding and visualization steps. Additional optimizations may include more efficient batching strategies, mixed-precision training, and distributed or multi-GPU training setups.

6 Conclusion

This work introduced AnoMATE, an autoencoder-based framework tailored for anomaly detection in large-scale, mixed-type financial microdata. While traditional detectors struggle with the high categorical cardinality and heterogeneity typical of datasets like AnaCredit and EMIR, AnoMATE addresses these challenges through a unified, type-consistent architecture. By replacing sparse one-hot encodings with learned entity embeddings, utilizing a multi-head decoder for varied feature types, and automating latent space sizing via intrinsic dimension estimation, the framework offers a principled alternative to manual, trial-and-error pipelines.

Empirical evaluations confirm the efficacy of this approach. AnoMATE achieved superior PR-AUC on the Vehicle Insurance and IEEE-CIS benchmarks while maintaining competitive performance on Fraud E-commerce. Crucially, the transition to learned embeddings reduced trainable parameters by up to 47%, with synthetic tests demonstrating that these efficiency gains scale favorably as categorical vocabulary sizes increase. These results suggest that embedding-based encoding is not only methodologically superior for mixed-type data but also practically essential for reducing the computational footprint in resource-constrained environments.

Beyond predictive accuracy, AnoMATE provides intrinsic interpretability critical for supervisory contexts. By decomposing anomaly scores into feature-level reconstruction errors and expected values, the model offers a transparent justification for every flagged record. This native explainability, paired with an interactive dashboard for real-time analysis, ensures that the system serves as an actionable decision-support tool rather than a "black-box" signal. While limitations regarding temporal dependencies and computational scaling persist, AnoMATE establishes a robust, scalable, and interpretable foundation for modern financial supervision.

References

[1] Sercan Ö. Arık and Tomas Pfister. 'TabNet: Attentive Interpretable Tabular Learning'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 8. 2021, pp. 6679–6687. DOI: 10.1609/aaai.v35i8.16826.

[2] Liron Bergman and Yedid Hoshen. 'Classification-Based Anomaly Detection for General Data'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2020.

[3] Markus M. Breunig et al. 'LOF: Identifying Density-Based Local Outliers'. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. 2000, pp. 93–104. DOI: 10.1145/342009.335388.

[4] Chia-Yuan Chang et al. 'Data-Efficient and Interpretable Tabular Anomaly Detection'. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2023. DOI: 10.1145/3580305.3599294.

[5] Tianqi Chen and Carlos Guestrin. 'XGBoost: A Scalable Tree Boosting System'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[6] Elena Facco et al. 'Estimating the intrinsic dimension of datasets by a minimal neighborhood information'. In: *Scientific reports* 7.1 (2017), p. 12140.

[7] Markus Goldstein and Andreas Dengel. 'Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm'. In: *KI-2012: Poster and Demo Track* (2012).

[8] Yury Gorishniy et al. 'Revisiting Deep Learning Models for Tabular Data'. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 34. 2021.

[9] Léo Grinsztajn, Edouard Oyallon and Gaël Varoquaux. 'Why Do Tree-Based Models Still Outperform Deep Learning on Typical Tabular Data?' In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35. 2022.

[10] Cheng Guo and Felix Berkhahn. 'Entity Embeddings of Categorical Variables'. In: *arXiv preprint arXiv:1604.06737* (2016).

[11] Songqiao Han et al. 'ADBench: Anomaly Detection Benchmark'. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35. 2022, pp. 32142–32159.

[12] Sahand Hariri, Matias Carrasco Kind and Robert J. Brunner. 'Extended Isolation Forest'. In: *IEEE Transactions on Knowledge and Data Engineering* 33.4 (2021), pp. 1479–1489. DOI: 10.1109/TKDE.2019.2947676.

[13] Zengyou He, Xiaofei Xu and Shengchun Deng. 'Discovering cluster-based local outliers'. In: *Pattern Recognition Letters*. 2003.

[14] Geoffrey E Hinton and Ruslan R Salakhutdinov. 'Reducing the dimensionality of data with neural networks'. In: *science* 313.5786 (2006), pp. 504–507.

[15] Noah Hollmann et al. 'TabPFN-2.5: Advancing the State of the Art in Tabular Foundation Models'. In: *arXiv preprint arXiv:2511.08667* (2025).

[16] Noah Hollmann et al. 'TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2023.

[17] Xin Huang et al. 'TabTransformer: Tabular Data Modeling Using Contextual Embeddings'. In: *arXiv preprint arXiv:2012.06678* (2020).

[18] Guolin Ke et al. 'LightGBM: A Highly Efficient Gradient Boosting Decision Tree'. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017.

[19] Diederik P. Kingma and Max Welling. 'Auto-Encoding Variational Bayes'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2014.

[20] Aleksandar Lazarevic and Vipin Kumar. 'Feature bagging for outlier detection'. In: *ACM SIGKDD*. 2005.

[21] Zheng Li et al. 'COPOD: copula-based outlier detection'. In: *IEEE International Conference on Data Mining (ICDM)*. 2020.

[22] Zheng Li et al. 'ECOD: Accelerated System-agnostic Outlier Detection through Unsupervised Cumulative Distribution Fitting'. In: *IEEE International Conference on Data Mining (ICDM)*. 2022.

[23] Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou. 'Isolation Forest'. In: *Proceedings of the 2008 IEEE International Conference on Data Mining (ICDM)*. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[24] Leland McInnes et al. 'UMAP: Uniform Manifold Approximation and Projection'. In: *The Journal of Open Source Software* 3.29 (2018), p. 861.

[25] Ricardo Gaspar Müller et al. 'RESHAPE: Explaining Accounting Anomalies in Financial Statement Audits by Enhancing SHapley Additive exPlanations'. In: *Proceedings of the 3rd ACM International Conference on AI in Finance (ICAIF)*. 2022. DOI: 10.1145/3533271.3561667.

[26] Tomáš Pevný. 'LODA: Lightweight on-line detector of anomalies'. In: *Machine Learning* (2016).

[27] Liudmila Prokhorenkova et al. 'CatBoost: Unbiased Boosting with Categorical Features'. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 31. 2018.

[28] Chen Qiu et al. 'Neural Transformation Learning for Deep Anomaly Detection Beyond Images'. In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2021.

[29] *Regulation (EU) 2016/867 of the European Central Bank of 18 May 2016 on the collection of granular credit and credit risk data (ECB/2016/13)*. EU Regulation. OJ L 144, 1.6.2016, p. 44–86. 18th May 2016.

[30] Lukas Ruff et al. 'Deep One-Class Classification'. In: *Proceedings of the International Conference on Machine Learning (ICML)*. Vol. 80. 2018, pp. 4393–4402.

[31] Lukas Ruff et al. 'Deep one-class classification'. In: *International Conference on Machine Learning (ICML)*. 2018.

[32] Timur Sattarov, Dayananda Herurkar and Jörn Hees. 'Explaining anomalies using denoising autoencoders for financial tabular data'. In: *arXiv preprint arXiv:2209.10658* (2022).

[33] Timur Sattarov, Marco Schreyer and Damian Borth. 'Diffusion-Scheduled Denoising Autoencoders for Anomaly Detection in Tabular Data'. In: *Proceedings of the ACM International Conference on AI in Finance (ICAIF)*. 2025. DOI: 10.1145/3711896.3736910.

[34] Bernhard Schölkopf et al. 'Estimating the Support of a High-Dimensional Distribution'. In: *Neural Computation* 13.7 (2001), pp. 1443–1471. DOI: 10.1162/089976601750265003.

[35] Marco Schreyer et al. 'Detection of Accounting Anomalies in the Latent Space using Adversarial Autoencoder Neural Networks'. In: *KDD Workshop on Anomaly Detection in Finance*. 2019.

[36] Marco Schreyer et al. 'Detection of Anomalies in Large Scale Accounting Data using Deep Autoencoder Networks'. In: *arXiv preprint arXiv:1709.05254* (2017).

[37] Mei-Ling Shyu et al. 'A novel anomaly detection scheme based on principal component classifier'. In: *Miami Univ Coral Gables FL Dept of Electrical and Computer Engineering* (2003).

[38] Gowthami Somepalli et al. 'SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training'. In: *arXiv preprint arXiv:2106.01342* (2021).

[39] Hugo Thimonier, Fabrice Popescu and Gaël Richard. 'Beyond Individual Input for Deep Anomaly Detection on Tabular Data'. In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2024.

[40] Tomas Tokar and Scott Sanner. 'ICE-T: Interactions-Aware Cross-Column Contrastive Embedding for Heterogeneous Tabular Datasets'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2025. DOI: 10.1609/aaai.v39i25.35385.

[41] Che-Ping Tsai et al. 'AnoLLM: Large language models for tabular anomaly detection'. In: *The Thirteenth International Conference on Learning Representations*. 2025.

[42] Hongzuo Xu et al. 'Fascinating Supervisory Signals and Where to Find Them: Deep Anomaly Detection with Scale Learning'. In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2023.

[43] Bo Zong et al. 'Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection'. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2018.