DEUTSCHE BUNDESBANK
EUROSYSTEM

Explaining Anomalies using Denoising Autoencoders for Financial Tabular Data

Technical Report 2023-01

Deutsche Bundesbank, Research Data and Service Centre,
German Research Center for Artificial Intelligence (DFKI)

Timur Sattarov
Dayananda Herurkar
Jörn Hees

Research Data and Service Centre

Abstract

Recent advances in Explainable AI (XAI) increased the demand for deployment of safe and interpretable AI models in various industry sectors.Despite the latest success of deep neural networks in a variety of domains, understanding the decision-making process of such complex models still remains a challenging task for domain experts.Especially in the financial domain, merely pointing to an anomaly composed of often hundreds of mixed type columns, has limited value for experts.

Hence, in this paper, we propose a framework for explaining anomalies using denoising autoencoders designed for mixed type tabular data. We specifically focus our technique on anomalies that are erroneous observations. This is achieved by localizing individual sample columns (cells) with potential errors and assigning corresponding confidence scores. In addition, the model provides the expected cell value estimates to fix the errors.

We evaluate our approach based on three standard public tabular datasets (Credit Default, Adult, IEEE Fraud) and one proprietary dataset (Holdings).We find that denoising autoencoders applied to this task already outperform other approaches in the cell error detection rates as well as in the expected value rates. Additionally, we analyze how a specialized loss designed for cell error detection can further improve these metrics. Our framework is designed for a domain expert to understand abnormal characteristics of an anomaly, as well as to improve in-house data quality management processes.

Keywords: explainable AI, explainable anomaly detection, tabular data, cell error detection, neural networks, unsupervised

Version: 1.0

Citation: Sattarov, T., Herurkar, D., and Hees, J. (2023). Explaining Anomalies using Denoising Autoencoders for Financial Tabular Data (), Technical Report 2023-01 – Version 1.0. Deutsche Bundesbank, Research Data and Service Centre; German Research Center for Artificial Intelligence (DFKI) [1) 2)]

1 We thank the members of the statistics department at the Deutsche Bundesbank for their valuable review and remarks. Further, we would like to thank Marco Schreyer (University of St. Gallen) for his valuable feedback and advice. Parts of this work were supported by the BMBF/BMWK projects XAINES (Grant 01IW20005) and EuroDaT (Grant 68GX21010K).
2 This technical report was presented at the 'Workshop on Explainable AI in Finance' https://sites.google.com/view/2022-workshop-explainable-ai/home

Contents

1 Introduction

Financial regulatory authorities and supervisory agencies play one of the most important roles in the financial system of a country. The main objective of the authorities is to secure the financial and monetary stability, supervision of national credit institutions as well as the management of payment service mechanisms. To fulfill these objectives, national statistical offices of the regulatory authorities need to collect monetary, financial and external sector statistical data. After the Global Financial Crisis (GFC) of 2008 – 2009 the enhancement of the financial framework has become compelling[3)]. In addition to the stronger oversight of financial firms, the GFC led to the call for strengthening and extension of the financial statistics[4)]. Following the above-mentioned initiatives, the demand for high quality financial microdata has appeared. To monitor the vulnerability of the economy to shocks and identify systemic risks, collection of high-quality microdata plays a vital role. For National Competent Authorities (NCA), the correctness and completeness of the collected data has to be ensured. Moreover, given the large volumes of collected data today, NCAs have to develop and deploy efficient data quality check (QC) procedures. Hence, typically a set of handcrafted rules are developed as rudimentary hard-coded checks. However, these are only able to detect already known reported errors and are not capable of identifying new types of errors. Further, it is crucial to not only identify an anomalous observation, but also flag the field(s) that contain reporting error(s). Therefore, explaining which values caused an irregularity is essential for financial microdata.

Today, a number of deep learning based techniques are introduced for anomaly detection in tabular data (Pang, Shen, Cao, and Hengel, 2021). However, in practice such tools are often insufficient due to the lack of interpretation. The ability to explain anomaly characteristics is as important as the quality of the trained model. For a domain expert, it is crucial to obtain a comprehensive explanation that would build a connection between a high anomaly score and a set of features affecting this score. Moreover, an inquiry to the reporting agent about the erroneous observation can be made and help with the correction. Therefore, the utilization of the anomaly interpretation features would significantly improve the applicability of such models in regulatory practice.

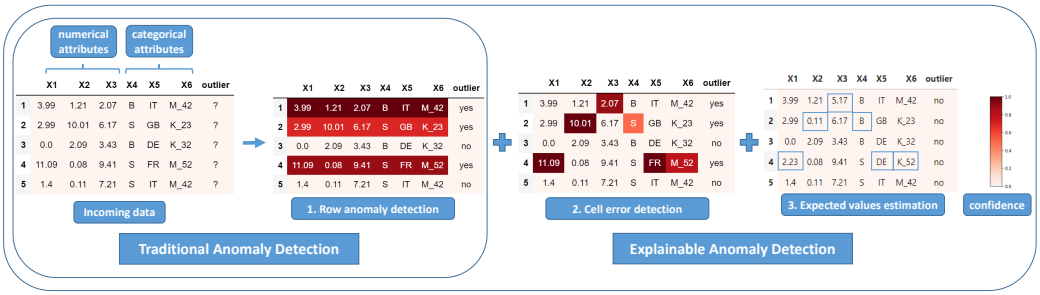


Figure 1: A schematic process overview of explainable anomaly detection (AD) for mixed type tabular data. In comparison to traditional AD which allows only row anomaly detection (1), explainable AD supplies the detection of cells responsible for high anomaly score (2) as well as the estimation of expected values for fixing an error (3). The coloring reflects the error confidence of a particular cell entry.

In this work, we propose a practical framework using denoising autoencoder (DAE) neural networks that not only isolates anomalous data points, but also flags the fields that caused the irregularity. The framework is designed for financial tabular data with categorical and numerical

3 https://ec.europa.eu/commission/presscorner/detail/de/MEMO_13_679
4 https://www.imf.org/external/np/g20/pdf/102909.pdf

(mixed) type. Figure 1 illustrates an example of traditional and explainable anomaly detection on financial tabular data. Traditional anomaly detection techniques flag the entire record as an anomaly (step 1) providing only a single score for each observation. This information is not enough to understand the cause of irregularity and only answers the question *"which samples are anomalies?"*. Our framework extends it to explainable anomaly detection providing cell error detection mechanism (step 2) which allows answering the question *"why is it an anomaly?"*. In addition, the model is capable of estimating the expected values, that should have been in place of the errors (step 3). This property allows answering the question *"what should have been reported instead?"*. These steps are utilized as the explainability properties of the model and help the domain expert to understand the anomalous characteristics of the detected anomalies.

In summary, we present the following contributions:

– We demonstrate that denoising autoencoder neural networks can be utilized to explain the cause of irregularity of a particular sample for mixed type tabular data.
– We show that such a model can successfully detect reporting errors on the attribute level (cell) providing corresponding confidence scores, as well as proposing the expected estimates for fixing the error.
– We propose an extension of the model with an enhanced loss and illustrate that such technique outperforms traditional methods based on the selected metrics.

The remainder of this paper is structured as follows: section 2 provides an overview of the related work. In section 3 we describe the autoencoder neural network model with its denoising extension together with the proposed methodology for detecting the erroneous cells. Next, section 4 and 5 outline the experimental setup and results. We conclude the paper with a summary and future research directions in section 6.

2 Related Work

The literature survey hereafter focuses on (1) developed row and cell anomaly detection techniques for financial tabular data, and (2) existing models designed for explainable anomaly detection.

2.1 Anomaly Detection in Financial Tabular Data

Anomaly detection has been an active research area in different domains, with a number of methods developed using deep learning (Pang et al., 2021). Especially, tabular data is becoming more and more attractive for deep learning techniques (Borisov et al., 2021). Nowadays, autoencoders have been widely used not only for representation learning but also for anomaly detection in variety types of financial data (Chalapathy and Chawla, 2019). Recently, a number of techniques were developed using autoencoders to detect anomalies in large scale accounting data (Schreyer, Sattarov, Borth, Dengel, and Reimer, 2018; Schreyer, Sattarov, Schulze, Reimer, and Borth, 2019; Schultz and Tropmann-Frick, 2020), identify traces of money laundering and fraud [Paula, Ladeira, Carvalho, and Marzagão (2016), 8324876} or learn behavioral fraud features (Wedge, Kanter, Rubio, Perez, and Veeramachaneni, 2017). Besides this, Schreyer et al. (Schreyer, Sattarov, and Borth, 2021) have demonstrated successful detection of accounting anomalies in a self-supervised learning setup together with downstream audit tasks. Moreover, autoencoders are a popular technique for detecting credit card fraud schemes (Kazemi and Zarrabi, 2017; Pumsirirat and Yan, 2018). In the context of financial fraud, a number of unsupervised and semi-supervised techniques are gaining popularity (Hilal, Gadsden, and Yawney, 2022).

Recently, Nazabal et al. (Nazabal, Olmos, Ghahramani, and Valera, 2020) proposed a framework to model variational autoencoders for fitting missing cells in the data. The technique includes handling not only categorical and numerical data types but also ordinal, interval and count. Also, similar to our approach, Eduardo et al. in (Eduardo, Nazábal, Williams, and Sutton, 2020) proposed the robust version of the VAE for cell-wise outlier detection for mixed type data.

2.2 Explainable Anomaly Detection

The field of "Explainable AI" (XAI) is rapidly developing, enhancing variety of the models which help the domain experts slightly open the "black-box" and understand the underlying decision-making process of the complex algorithms (Das and Rad, 2020). Recently, there have been a number of techniques introduced (Du, Liu, and Hu, 2018; Murdoch, Singh, Kumbier, Abbasi-Asl, and Yu, 2019) in the area of XAI. Such model agnostic methods like SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017), Local Interpretable Model-Agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin, 2016) or DeepLIFT (Shrikumar, Greenside, and Kundaje, 2017) showed significant success for their abilities to explain the output of almost any machine learning model. At the same time, the usage of shapley values is becoming popular in explaining anomalies. Antwarg et al. (Antwarg, Miller, Shapira, and Rokach, 2019) used the kernel SHAP to explain the anomalies detected by the autoencoder neural network in an unsupervised scenario. Similarly, Takeshi et al. (Takeishi, 2020) successfully used the power of shapley values in linear models such as PCA. Nguyen et al. (M.-N. Nguyen and Vien, 2018) have proposed the combined version of the autoencoder and OC-SVM to explain the decision-making process of detected outliers in unsupervised

anomaly detection tasks. Another unsupervised attempt was made by Chen et al. (Chen, Tian, Pang, and Carneiro, 2021) to localize structural and non-structural anomalies in computer vision. Previously, Bergmann et al. (Bergmann, Löwe, Fauser, Sattlegger, and Steger, 2019) proposed the perceptual loss for autoencoders to identify inter-dependencies between local image regions. Recently, Amarasinghe et al. (Amarasinghe, Kenney, and Manic, 2018) developed a framework using deep neural networks to explain the cause of detected DoS attacks in a supervised manner. A gradient-based approach was utilized by Nguyen et al. (Q. P. Nguyen, Lim, Divakaran, Low, and Chan, 2019) to develop a framework for detecting anomalies in a network traffic using variational autoencoder. Another attempts using attention learning mechanism were proposed by Venkataramanan et al. (Venkataramanan, Peng, Singh, and Mahalanobis, 2019) and Xu et al. (Xu et al., 2021). Also, an explainable recommendation system using autoencoder was developed by Haghighi et al. (Haghighi, Seton, and Nasraoui, 2019). The model was designed to explain the outputs of the recommender. Kauffmann et al. (Kauffmann, Müller, and Montavon, 2020) used a deep Taylor decomposition to explain various anomaly types. Another practical application to explain the output of black-box model was described by Ramamurthy et al. (Ramamurthy, Vinzamuri, Zhang, and Dhurandhar, 2020). They build a multilevel explanation tree that characterizes the local and global explanations of the records. A number of attempts were also made to model the detection of cell errors in medical and geoscience domains. Jan et al. (Walach, 2020) have proposed a cell outlier diagnostics detection technique and evaluated it on three different medical datasets. Similarly, the importance of multivariate outlier detection in the field of geosciences was recently demonstrated in by Filzmoser et al. (Filzmoser and Gregorich, 2020).

According to the systematic review of Riyanul et al. (Islam, Ahmed, Barua, and Begum, 2022) only 2% of the XIA research papers are focused on the finance domain. The literature survey above also demonstrated the overall popularity of XAI techniques, but very limited application of anomaly explanations for financial data, especially in combination with denoising autoencoder neural networks.

3 Methodology

In this section, we describe the autoencoder neural network, its denoising extension with the proposed loss, as well as the specification of the framework for explaining anomalies.

3.1 Autoencoder Neural Network

Formally, we denote a set of instances $x_1, x_2, ..., x_N$ in a tabular dataset X. Every instance encompasses a set of attributes $d \in \{1, ..., D\}$ with either numeric $x_n^{d_{\text{num}}} \in \mathbb{R}$ or categorical type $x_n^{d_{\text{cat}}} \in \{1, ..., C\}$ where C is the total number of unique categories of the feature d.

An autoencoder (AE) neural network is a type of feed-forward network that aims to perform a lossy data compression into a lower dimensional feature space and afterwards reconstruct it back to the original data space with minimal loss. The encoder network f_θ performs the data compression and the decoder network g_ψ accomplishes the reconstruction. Upon the successful model training with a set of parameters θ and ψ, the reconstruction error is often used to quantify the anomaly degree of an instance. The reconstruction error reflects how good an instance fits into the general patterns of the data. Hence, an inlier receives a relatively low reconstruction error, whereas an outlier obtains a higher one, which attests its deviation from the common data structure. The network is trained in an end-to-end unsupervised fashion by minimizing the reconstruction loss, formally defined as follows:

$$\arg\min_{\theta\psi} \|X - g_\psi(f_\theta(X))\| \tag{1}$$

Due to the mixed type nature of the data, we define the reconstruction loss of every instance as the sum of two losses. For each one-hot encoded representation of the categorical attribute the (1) negative-log-likelihood loss is calculated and (2) the mean-squared loss used for the numerical attributes, formally expressed by:

$$\mathcal{L}_{\theta,\psi}(x_n^d, \hat{x}_n^d) = \sum_{d=1}^{D_{\text{cat}}} \mathcal{L}_{\theta,\psi}^{\text{NLL}}(x_n^d, \hat{x}_n^d) + \sum_{d=1}^{D_{\text{num}}} \mathcal{L}_{\theta,\psi}^{\text{MSE}}(x_n^d, \hat{x}_n^d) \tag{2}$$

where $\hat{x}_n^d$ denotes the $n-th$ reconstructed sample and its attribute d. We have observed that such loss design suits better for mixed data type as it leads to a faster overall model convergence.

3.2 Denoising Autoencoder Neural Network

The denoising autoencoder (DAE) is an extension of the traditional autoencoder neural network with the goal of removing noise from the signal. Such a model is trained by disrupting the input data with random noise and reconstructing the clean data. At first, a corrupted instance $\tilde{x}$ is created by adding random noise to the clean input instance x. Next, the encoder network f_θ performs the compression of the corrupted instance $\tilde{x}$, and the decoder network g_ψ accomplishes the reconstruction $\hat{x}$ (noise removal). The training objective function is the same as in the Equation 1. In the inference phase, the trained model is capable of transforming the noisy data into noiseless.

In addition to the denoising capabilities, such model modification improves the robustness of the hidden layers (i.e., latent layer representation) (Vincent, Larochelle, Bengio, and Manzagol, 2008) as well as reduces the risk of overfitting.

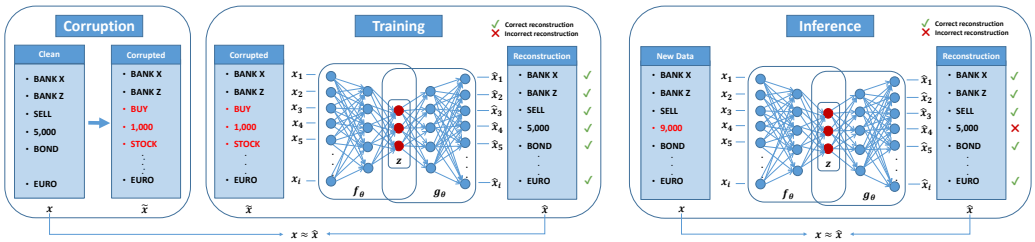


Figure 2: A schematic overview of the training and inference phases of DAE for cell error detection and estimation of expected values. In the corruption phase, random noise is added to a sample. Next, the DAE is trained on the corrupted data with the goal to reconstruct the clean data. In the inference phase, the cells that failed to reconstruct, are flagged as an error, and the reconstructed value is used as an estimation of the expected value to fix the error.

Enhanced loss. We have noticed, that the selection of the right amount of injected noise during training is quite challenging for mixed type tabular data. Too much noise leads the model to focus mainly on the noise removal task, and the network fails to reconstruct clean data sufficiently enough. On the contrary, too little noise decreases noise removal capabilities, which affects the overall cell error detection rate. Therefore, we propose an extension to the loss function of the DAE. Specifically, we introduce a parameter $\alpha = [0, 1]$ that allows us to weigh the noise removal vs. clean data reconstruction within the batch. In practice, we found that selection of a fixed α is challenging. Therefore, we propose to sample it from the *Beta(0.5, 0.5)* distribution. The random sampling of α can be understood as a regularization technique (somewhat similar to dropout) to optimize both of our goals (noise removal and reconstruction of clean data). The alternating nature of α (i.e., being sampled close to the extremes of 0 or 1) seems to be beneficial to training in many cases. The final loss has the following formulation:

$$\mathcal{L}_{\theta,\psi}(x_n^d; \hat{x}_n^d) = \alpha\ \mathbf{m} \odot \mathcal{L}_{\theta,\psi}(x_n^d; \hat{x}_n^d) + (1 - \alpha)\ \mathbf{m} \odot \mathcal{L}_{\theta,\psi}(x_n^d; \hat{x}_n^d) \tag{3}$$

where $\mathbf{m}$ is a binary mask vector $\mathbf{m} \in \{0, 1\}^d$ that yields 1 at the entry with noise or 0 otherwise,

$\mathbf{m}$ is its complement, and $\odot$ is the element-wise multiplication.

3.3 Anomaly Explainer

To explain the cause of an anomaly, we utilize the properties of the reconstruction error of each separate attribute of the trained DAE. Such an approach is also quite common in practice using traditional AE. Although this typically yields good performance on the detection of row anomalies, it becomes less precise in identifying the exact cells that contain errors. Hence, the goal of the proposed framework is to answer three questions by supplying the domain expert with the following information:

- **Which samples are anomalies?** *Row Anomalies*: identify a subset of K row anomalies with the highest reconstruction errors.
- **Why is it an anomaly?** *Cell Errors*: for every cell in K selected anomalies, compute the confidence π_n^d that the value x_n^d contains an error.
- **What should have been reported instead?** *Expected Values*: for every cell in K selected anomalies, collect the reconstructed value $\hat{x}_n^d$.

Training. As depicted in Figure 2 the DAE is trained to reconstruct a clean (noiseless) instance x_n from its corrupted counterpart $\tilde{x}_n$. During the training, the reconstruction error between the clean instance x_n and its reconstruction $\hat{x}_n$ is minimized.

Inference. Once the DAE is trained, the reconstruction error $\hat{x}_n^d$ of each attribute value of the test (unseen) data is calculated. Depending on the attribute type (either categorical or numerical) two different functions are applied to obtain the error confidence π_n^d of this cell. For categorical attributes, we compute the complement of the normalized reconstruction category c as the following:

$$\textbf{Cell:} \quad \pi_n^{d_{\text{cat}}} = 1 - a_n^{dc} \tag{4}$$

where $a(\cdot)$ is the softmax function $a_n^{dc} = \text{softmax}(\hat{x}_n^d)^c$ calculated on the reconstructed representation of the attribute d. The superscript c identifies a particular category in that attribute. For numerical attributes, we compute the complement of the negative exponential function between the input value x_n^d and its reconstruction $\hat{x}_n^d$ as the following:

$$\textbf{Cell:} \quad \pi_n^{d_{\text{num}}} = 1 - e^{-(x_n^d - \hat{x}_n^d)^2} \tag{5}$$

Correspondingly, the row anomaly score is computed as the sum of all categorical and numerical

cell scores π_n^d:

$$\textbf{Row:} \quad \pi_n = \sum_{d=1}^{D} \pi_n^d \tag{6}$$

The expected values are obtained by collecting the reconstructed values $\hat{x}_n^d$. For categorical attributes, we use the highest probability category $\arg\max_c a_n^{dc}$ of the softmax transformation a_n^d.

4 Experimental setup

In this section, we describe the details of the conducted experiments. We describe the datasets as well as the noise injection procedure that was applied to these datasets, together with the metrics used to evaluate the performance of the results. For training and evaluation of the neural network models, the PyTorch v1.10.2 (Paszke et al., 2019) framework was used.

4.1 Datasets

We benchmark the developed technique with open-source and real world datasets. Three public datasets and one proprietary dataset were selected to evaluate the performance of the proposed framework. Below, we provide the description of each dataset:

- **Credit Default** [5]: The dataset is taken from the UCI machine learning repository (Dua and Graff, 2017) and contains information on bill statements of credit card clients, their default payments, history of payment as well as the demographic factors of the clients in Taiwan during the period April 2005 to September 2005 (Yeh and Lien, 2009).
- **IEEE Fraud**[6]: The dataset consists of the electronic transactions from the e-commerce service provider Vesta Corporation. The dataset was published to improve the efficiency of the fraud detection alert system.
- **Adult**[7]: The dataset is taken from the UCI machine learning repository (Dua and Graff, 2017) and consists of personal income records, where the task is to predict whether an income exceeds $50k per year.
- **Holdings**[8]: This proprietary dataset consists of the individual holdings of the investment funds issued by investment companies (Blaschke, Haupenthal, Schuck, and Yalcin, 2021). Each record reflects the asset or liability value submitted by the reporting entity at the end of the month.

All datasets have mixed attribute types as described in Table 1. In the data preprocessing step, all categorical attributes are encoded using the one-hot encoding technique. Numerical attributes are standardized to have 0 mean and standard deviation 1. Afterwards, the one-hot encoded representation is combined with standardized numerical attributes. The final number of encoded attributes is reflected in the last column ("Encoded") of the Table 1.

Table 1: Descriptive statistics of the selected datasets

Data	Rows	Columns		
		Categ.	Num.	Encoded
Credit Default	30,000	10	13	160
IEEE Fraud	569,877	14	380	502
Adult	32,561	8	5	126
Holdings	118,569	7	129	203

5 The dataset is publicly available via: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
6 The dataset is publicly available via https://www.kaggle.com/c/ieee-fraud-detection/overview
7 The dataset is publicly available via https://archive.ics.uci.edu/ml/datasets/adult
8 In compliance with strict data privacy regulations, neither content nor the descriptive statistics of the dataset can be made publicly available.

4.2 Corruption process

To the best of our knowledge, there is no publicly available dataset with labeled cell errors. Therefore, it is a standard practice to artificially generate anomalies by randomly corrupting the clean data (Krishnan, Wang, Wu, Franklin, and Goldberg, 2016; Redyuk, Schelter, Rukat, Markl, and Biessmann, 2019). In our approach, we also follow a similar strategy and turn 3% of the inliers into the outliers by randomly corrupting attribute values in both training and test sets. Selection of the attributes for data corruption is also done at random. We corrupt at most 50% of the features which are selected uniformly at random as following: c = Unif(1, $\frac{c_{max}}{2}$), where c_{max} is the total number of features. Dataset Holdings already contains the real-world cell errors together with its clean value.

To artificially corrupt the samples, we applied different techniques for both categorical and numerical features.

Numerical features. The injection of noise for a numerical feature is performed using an additive noise process, with the corrupted value obtained as: $\tilde{x}_n^d = x_n^d + \delta$. Here δ is randomly sampled from one of the Gaussian, Laplace, or Log-Normal distributions with $\mu = 0$ and $\sigma = \sigma_d \gamma$. Selection of γ follows uniform distribution γ = Unif(3, 5) and σ_d is the standard deviation of the original attribute. The selection of the distribution at the corruption phase is also done uniformly at random.

Categorical features. Two alternatives are used to inject a noise into categorical attributes. With the first alternative, the original entry is replaced by picking a categorical entry uniformly at random from the distinct values of this attribute. The second option creates a new categorical entry by performing character manipulations (insertion, flipping or deletion) with the original categorical entry and ensuring a completely new entry is created. Such technique in practice imitates a typo that can often appear during the data insertion process.

4.3 Evaluation metrics

To assess the quality of the proposed technique, we utilize the following three metrics and measure the detection rate.

Precision at K (P@K). We utilized this metric for the traditional row anomaly detection to assess overall model capabilities to detect anomalies. Hence, this metric is referred to our first question, *"which samples are anomalies?"*. The metric is popular in recommendation system evaluation tasks, where the user is interested only in the top K predictions. Similarly, in a regulatory reporting environment, it is important that top K retrieved anomalies are indeed relevant, hence reducing the false positive rate as well as human effort.

$$\text{P@K}(y, \hat{y}) = \frac{\text{TP@K}(y, \hat{y})}{K} \tag{7}$$

where TP@K is the total number of true anomalies in the top K retrieved anomalies given the vectors y and $\hat{y}$ of true anomalies and row-wise reconstruction errors correspondingly. The value of K in our case is selected as the total number of true anomalies in the test set.

Mean Average Precision (mAP). This metric reflects the performance of the model in answering the second question, *"why is it an anomaly?"*. Thus, it estimates the quality of cell error detection across all attributes. The confidence of the cell error π^d, described in subsection 3.3, is used as the input to the function for computing the Average Precision (AP). The positive labels in this case are the cells with noise. Formally, it is defined as:

$$\mathrm{AP}(\pi^d) = \sum_{i=1}^{N} (R_i - R_{i-1}) P_i \tag{8}$$

where $R_i(\pi^d) = TP/(TP + FN)$ denotes the detection recall and $P_i(\pi^d) = TP/(TP + FN)$ denotes the detection precision of the $i - th$ anomaly score threshold. The mean Average Precision (mAP) is computed as the average of the AP scores across all attributes $\mathrm{mAP} = \frac{1}{D} \sum_{d=1}^{D} \mathrm{AP}(\pi^d)$.

Mean Expected Value (mEV). With this metric, we evaluate the ability of the model to answer the third question, *"what should have been reported instead?"*. In order to assess the correctness of the expected (or fixed) values, we compute the Standardized Mean Squared Error (SMSE) between the original ground truth and its reconstruction.

For numerical attributes, it is additionally normalized by the empirical variance of this attribute and has the following form:

$$\mathrm{EV}(x^{d_{\mathrm{num}}}) = \frac{1}{\hat{N}} \sum_{n=1}^{\hat{N}} \frac{(x_{no}^d - \hat{x}_n^d)^2}{\sigma^2} \tag{9}$$

where σ is the standard deviation and $\hat{N}$ is the total number of corrupted cells in the attribute x^d. The subscript o in x_{no}^d denotes the ground truth original value (i.e., without error).

For categorical features, we utilize the *Brier score* (Brier, 1950) between the one-hot representation of the ground truth value and the reconstructed softmax representation of this category and has the following form:

$$\mathrm{EV}(x^{d_{\mathrm{cat}}}) = \frac{1}{2\hat{N}} \sum_{n=1}^{\hat{N}} \sum_{c=1}^{C} (x_{no}^{dc} - \hat{x}_n^{dc})^2 \tag{10}$$

where x_{no}^{dc} is the one-hot encoded value of the category c. The factor $\frac{1}{2}$ is used to normalize the score to the range [0, 1].

The mean Expected Value (mEV) is computed as the average of Expected Values (EV) mEV = $\frac{1}{D}\sum_{d=1}^{D}$ EV(x^d) across all attributes.

4.4 Model training setup

We split every dataset into training and test sets by a fraction of 70 and 30 correspondingly. According to the anomaly injection process described in section 3 the test set (and if necessary, train set) is populated with noise. Once the model is trained, all evaluation metrics are collected on the test set. The exact network architecture used for each dataset is described in Table 2.

Table 2: Selected architecture setup of the (denoising) autoencoder neural network used for each dataset

Dataset	Neurons per hidden layer
Credit Default	160-128-64-128-160
Adult	126-128-64-128-126
IEEE Fraud	502-512-256-512-502
Holdings	203-256-128-256-203

We train every model for a maximum of 5000 epochs with a mini-batch of size 128 and use the Adam optimizer (Kingma and Ba, 2014) with β_1 = 0.9, β_2 = 0.999 in combination with a cosine learning rate scheduler. The parameters of the encoder and decoder are randomly initiated as described in (Glorot and Bengio, 2010).

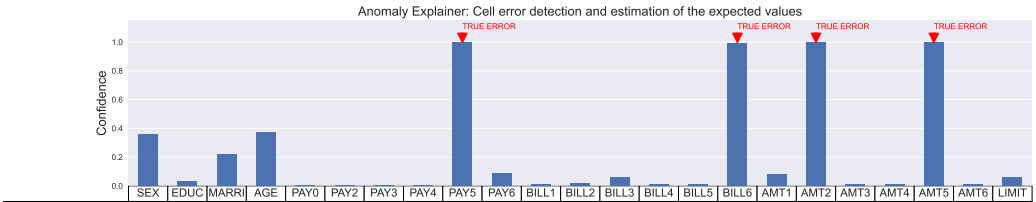
Baseline models. To illustrate the practical applicability of the proposed technique, we compare its performance against several methods where cell error detection is feasible. Therefore, we select PCA (S., 1901), *Marginal Distribution* and traditional AE (Hinton and Salakhutdinov, 2006). For *Marginal Distribution,* we follow the same approach described by Eduardo et al. (Eduardo et al., 2020) and fit a Gaussian mixture model on every numeric attribute separately, using the negative log-likelihood as the cell error. For categorical attributes, a normalized category frequency is used for expected value estimation. For the AE, we evaluate two scenarios: the training set contains anomalies (AE with anomalies) and the training set does not contain anomalies (AE no anomalies). The first scenario imitates the case in industry when the AE is trained from scratch every time new data arrives, without any knowledge about the historical data. The second scenario imitates the case when the historical data with cell errors and their corrected values is available. Here it is possible to train the model on a pure "clean" version of the historical data and evaluate on the unseen data with anomalies.

5 Experimental results

In this section, we describe the results of the conducted experiments. We demonstrate the practical applicability with qualitative assessment as well as the efficiency of the proposed technique, providing the quantitative results.

5.1 Qualitative evaluation

To explain the cause of irregularity of a potential anomaly, the framework arms the domain expert with a powerful visual inspection tool. Every potential anomaly can be quickly screened and the question *"why is it an anomaly?"* can be answered. This is achieved by flagging individual cells with detected errors. Figure 3 depicts the interface of the cell error detection framework. Here, the height of the bars reflects the model's confidence about the reported errors of a new sample. Next, to allow the domain expert to answer the question *"what should have been reported instead?"*, the framework proposes the expected sample. It gives an estimation of the expected values to be reported. In addition to the cell scores, five similar data samples picked from the original dataset are shown under the screened sample. This allows the domain expert to compare certain entries of the screened sample with the entries of its closest neighbors. Selection of such samples is computationally inexpensive, since the pairwise distances are computed on the transformed representation of the bottleneck layer of the DAE. With this tool, the domain expert can pick any data sample, produce such a graphic to quickly assess the nature of the reported errors and execute necessary steps, if required. Such a compact form (1) provides more explanation capabilities about the anomaly nature, (2) saves the screening time, and (3) reduces the human error during the quality checks.



	SEX	EDUC	MARRI	AGE	PAY0	PAY2	PAY3	PAY4	PAY5	PAY6	BILL1	BILL2	BILL3	BILL4	BILL5	BILL6	AMT1	AMT2	AMT3	AMT4	AMT5	AMT6	LIMIT
new sample	1	1	2	40	-1	-1	-1	-1	6	-1	3457	1997	5115	3295	222.	-142	2020	3635	3295	222.	-114	2129	3600
expected sample	1	1	2	40	-1	-1	-1	-1	-1	-1	1149	1212	2186	9009	-417	-125	7310	2011	4741	-148	3611	3531	3259
1st closest neighbor	1	2	1	41	0	4	8	0	0	0	2167	7142	2371	2412	2468	2522	3400	2640	2792	1100	1100	1103	7000
2nd closest neighbor	1	2	2	35	1	-2	-1	0	0	-1	0.0	0.0	8167	4617	3308	2912	0.0	2202	7308	0.0	2912	0.0	2200
3rd closest neighbor	1	2	2	28	0	0	0	0	0	6	3910	4014	7314	7560	4301	4386	-17.	1675	1655	1710	1719	1616	9945
4th closest neighbor	1	1	1	38	0	0	0	0	0	0	8242	-142	1536	1553	1552	1608	5526	2259	5504	5655	9471	5269	4700
5th closest neighbor	2	2	1	41	-2	-1	-1	-1	-2	-2	852.	1738	1866	5517	2660	1480	1746	1874	5517	2660	1480	0.0	2400

Figure 3: Anomaly explainer dashboard that illustrates the outcome yield by the model trained on the Credit Default dataset. A random anomaly ("new sample") is picked from the test set. Potential cell errors are colored with red gradient and corresponding confidence is reflected in bar graphs. The red arrows point to the position of the true errors. The second row contains the expected values estimated by the model, and the remaining rows are the 5 closest original data instances based on the transformed low dimensional latent representation.

Latent space. Sampling of the anomalies for screening can also be done using latent data representation produced by the model. This is another powerful property of the framework that allows the domain expert to visually inspect the groups of observations. AEs possess this capability as they are able to learn expressive low dimensional representations of the data in the latent space. Albeit tree based models have the ability to plot the decision tree hierarchy, which makes them indeed preferable tool in industry, they lack the ability to provide the learned data representation with

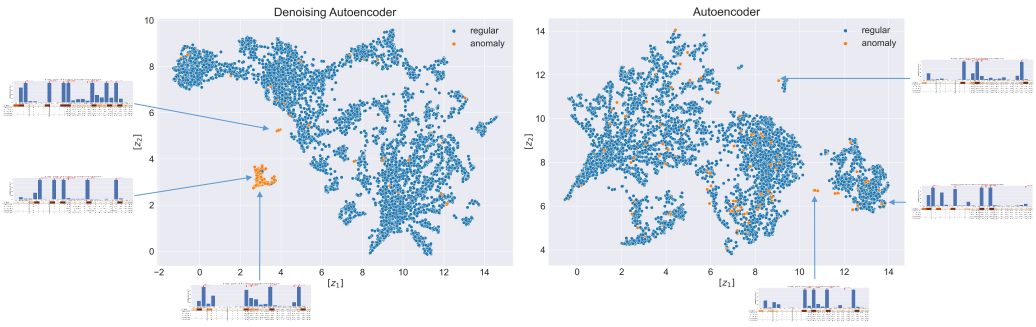


Figure 4: UMAP embedding of the latent representation z between the denoising autoencoder (left) and autoencoder (right) neural networks. The embedding is done on the Credit Default dataset and is projected from 64 to 2 dimensional space of the converged model after 5000 training epochs.

relative similarities between the observations. An example of such data representation is depicted on Figure 4. Here we plot the latent representation of the trained DAE and AE. The former provides a better isolation of anomalies from the regular data points by grouping them into a single cluster. In addition, the DAE seems to provide a more compact form of grouping the regular data points with similar characteristics. In contrast, the AE yields a sparse form of the data representation and anomalies also scattered across the whole latent space. Such representation becomes less valuable for the domain expert who expects to have more compact data representation with more or less clear group separations. This property is especially important in industry because it allows to "walk in the data" and quickly sample the data for screening. The domain expert can sample any data point and produce the anomaly explainer dashboard like on Figure 3. This gives an opportunity to quickly audit (financial) entities with similar underlying characteristics (1) as well as the entities that change their cluster assignment (2) which could lead to behavioral changes.

5.2 Quantitative evaluation

We are interested in the precise localization of errors in cells, as this explains the characteristics of an anomaly. As described earlier, we assess the quality of the proposed technique using different metrics, datasets and baseline models. Table 3 contains the scores collected from the conducted experiments. Based on these, the DAE outperforms the baseline AEs on almost all metrics and datasets. We believe that this is due to the fact that the traditional AE yields high reconstruction errors not only on the corrupted cells, but on other (neighboring) cells as well, which produces lots of false positives. Instead, the DAE due to its training nature, reconstructs each cell more precisely and hence, produces less false positives. In the cases where it concedes (mAP of numerical attributes of Adult and IEEE Fraud), we believe the reason lies in the uninformativeness of certain attributes (Grinsztajn, Oyallon, and Varoquaux, 2022). We have noticed, that for such scenarios, the Marginal model yields better results.

In addition, the AE trained only on clean data without anomalies (AE clean) almost always outperforms its counterpart trained on the data with anomalies (AE). We believe, that this happens because the AE trained with anomalies at some point during training shifts its focus towards learning the anomalies since they are responsible for the highest reconstruction errors. As a result, in the inference phase, the anomalies are getting lower reconstruction errors. That implies that in

Dataset	Model	P@K ↑	mAP ↑		mEV ↓	
			categorical	numerical	categorical	numerical (log)
Credit Default	PCA	0.584 ± 0.004	0.709 ± 0.013	0.211 ± 0.001	0.446 ± 0.001	2.750 ± 0.001
	Marginals	0.577 ± 0.013	0.539 ± 0.000	0.444 ± 0.002	0.343 ± 0.000	1.975 ± 0.022
	AE	0.651 ± 0.022	0.350 ± 0.029	0.238 ± 0.012	0.821 ± 0.011	2.846 ± 0.016
	AE clean	0.814 ± 0.008	0.615 ± 0.034	0.476 ± 0.009	0.633 ± 0.024	2.177 ± 0.028
	DAE	0.826 ± 0.005	0.818 ± 0.007	0.617 ± 0.005	0.245 ± 0.004	0.428 ± 0.037
	DAE*	**0.835 ± 0.007**	**0.821 ± 0.006**	**0.635 ± 0.015**	**0.243 ± 0.004**	**0.415 ± 0.035**
Adult	PCA	0.328 ± 0.000	0.135 ± 0.001	0.228 ± 0.001	0.422 ± 0.000	2.069 ± 0.001
	Marginals	0.620 ± 0.004	0.192 ± 0.000	**0.626 ± 0.008**	0.299 ± 0.000	1.988 ± 0.061
	AE	0.478 ± 0.007	0.144 ± 0.013	0.294 ± 0.011	0.953 ± 0.005	2.966 ± 0.012
	AE clean	0.634 ± 0.014	0.262 ± 0.009	0.493 ± 0.006	0.867 ± 0.012	2.536 ± 0.005
	DAE	0.636 ± 0.015	**0.544 ± 0.013**	0.528 ± 0.013	0.451 ± 0.020	**1.572 ± 0.043**
	DAE*	**0.638 ± 0.003**	0.532 ± 0.010	0.538 ± 0.007	**0.440 ± 0.006**	1.725 ± 0.035
IEEE Fraud	PCA	0.906 ± 0.001	0.622 ± 0.006	0.352 ± 0.001	0.487 ± 0.001	4.554 ± 0.001
	Marginals	0.972 ± 0.001	0.325 ± 0.000	**0.819 ± 0.001**	0.293 ± 0.000	4.474 ± 0.001
	AE	0.802 ± 0.008	0.531 ± 0.005	0.265 ± 0.006	0.787 ± 0.012	4.587 ± 0.032
	AE clean	**0.975 ± 0.001**	0.445 ± 0.014	0.510 ± 0.008	0.555 ± 0.021	4.412 ± 0.102
	DAE	**0.975 ± 0.004**	**0.766 ± 0.020**	0.784 ± 0.011	0.228 ± 0.004	**4.005 ± 0.003**
	DAE*	0.974 ± 0.001	0.765 ± 0.014	0.786 ± 0.006	**0.227 ± 0.006**	4.015 ± 0.125
Holdings	PCA	0.200 ± 0.007	0.005 ± 0.001	0.042 ± 0.003	**0.500 ± 0.001**	13.325 ± 0.001
	Marginals	0.092 ± 0.002	0.001 ± 0.000	0.083 ± 0.001	0.535 ± 0.000	11.610 ± 0.001
	AE	0.163 ± 0.017	0.010 ± 0.010	0.040 ± 0.004	0.974 ± 0.028	13.177 ± 0.100
	AE clean	0.157 ± 0.019	0.012 ± 0.006	0.039 ± 0.003	0.942 ± 0.077	13.169 ± 0.115
	DAE	**0.206 ± 0.005**	**0.045 ± 0.010**	**0.098 ± 0.009**	0.736 ± 0.035	**11.576 ± 0.018**
	DAE*	0.201 ± 0.007	0.030 ± 0.002	0.081 ± 0.006	0.709 ± 0.066	11.632 ± 0.034

Table 3: Comparative performance evaluation of the proposed model against the baselines on all datasets using three metrics. The model marked with asterisk was trained using the enhanced loss described in subsection 3.2. Every score reflects the mean and standard deviation from 5 experiments with varying initialization seeds. We can see that DAE outperforms its counterparts on average by 5%-30%.

practice it's better to deploy the model trained on the "clean" data (if available) rather than to retrain an AE on new (potentially noisy) data from scratch. Even more efficient is to use the historical anomalies and let the model learn from this.

The results on the real world dataset Holdings attest this. Since the dataset Holdings contained the clean and noise versions, we were also able to compare the performance of the DAE with various noise types. Three models were trained using only artificial noise (1), only real world noise (2) and real world + artificial noise (3). Based on the collected scores, using both (3) boosts the performance notably. This is expected, as it allows the DAE (unlike the AE) to also learn from the distribution of real world noise during training.

6 Conclusion and Future work

In this work, we proposed a framework for explaining detected anomalies using denoising autoencoder neural networks for mixed type tabular data. To explain the cause of an anomaly, the framework produces confidence scores of potential errors for every cell entry, as well as proposes corresponding estimated values to fix the errors. In addition, we propose the enhanced extension using the extended loss specifically designed for cell error detection.

We evaluated the proposed approach on three publicly available datasets and one proprietary financial dataset with mixed type attributes. The produced results are compared against the baseline and underpin the practical applicability of the proposed technique.

We believe that such a framework can become a helpful toolbox for data quality experts in their daily tasks and can be easily integrated into the corresponding procedural pipeline. We believe the technique can also be applied in a variety of other domains outside the financial field in the future.

References

Amarasinghe, K., Kenney, K., and Manic, M. (2018). Toward explainable deep neural network based anomaly detection. *2018 11th International Conference on Human System Interaction (HSI)*, 311–317. https://doi.org/10.1109/HSI.2018.8430788

Antwarg, L., Miller, R. M., Shapira, B., and Rokach, L. (2019). Explaining anomalies detected by autoencoders using SHAP. *arXiv Preprint arXiv:1903.02407*.

Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., and Steger, C. (2019). Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science; Technology Publications. https://doi.org/10.5220/0007364503720380

Blaschke, J., Haupenthal, H., Schuck, V., and Yalcin, E. (2021). *Investment funds statistics base 09/2009-06/2021*.

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., and Kasneci, G. (2021). *Deep neural networks and tabular data: A survey*. arXiv. https://doi.org/10.48550/ARXIV.2110.01889

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability.

Chalapathy, R., and Chawla, S. (2019). *Deep learning for anomaly detection: A survey*. Retrieved from https://arxiv.org/abs/1901.03407

Chen, Y., Tian, Y., Pang, G., and Carneiro, G. (2021). Unsupervised anomaly detection and localisation with multi-scale interpolated gaussian descriptors. *ArXiv*, *abs/2101.10043*.

Das, A., and Rad, P. (2020). *Opportunities and challenges in explainable artificial intelligence (XAI): A survey*. Retrieved from https://arxiv.org/abs/2006.11371

Du, M., Liu, N., and Hu, X. (2018). *Techniques for interpretable machine learning*. arXiv. https://doi.org/10.48550/ARXIV.1808.00033

Dua, D., and Graff, C. (2017). *UCI machine learning repository*. University of California, Irvine, School of Information; Computer Sciences. Retrieved from http://archive.ics.uci.edu/ml

Eduardo, S., Nazábal, A., Williams, C. K., and Sutton, C. (2020). Robust variational autoencoders for outlier detection and repair of mixed-type data. *International Conference on Artificial Intelligence and Statistics*, 4056–4066. PMLR.

Filzmoser, P., and Gregorich, M. (2020). Multivariate outlier detection in applied data analysis: Global, local, compositional and cellwise outliers. *Mathematical Geosciences*, *4*. https://doi.org/10.1007/s11004-020-09861-6

Glorot, X., and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). *Why do tree-based models still outperform deep learning on tabular data?* arXiv. https://doi.org/10.48550/ARXIV.2207.08815

Haghighi, P. S., Seton, O., and Nasraoui, O. (2019). *An explainable autoencoder for collaborative filtering recommendation*. Retrieved from https://arxiv.org/abs/2001.04344

Hilal, W., Gadsden, S. A., and Yawney, J. (2022). Financial fraud: A review of anomaly detection techniques and recent advances. *Expert Syst. Appl.*, *193*(C). https://doi.org/10.1016/j.eswa.2021.116429

Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*, 504–507.

Islam, M. R., Ahmed, M. U., Barua, S., and Begum, S. (2022). A systematic review of explainable artificial intelligence in terms of different application domains and tasks. *Applied Sciences*, *12*(3). https://doi.org/10.3390/app12031353

Kauffmann, J., Müller, K.-R., and Montavon, G. (2020). Towards explaining anomalies: A deep taylor decomposition of one-class models. *Pattern Recognition*, *101*, 107198. https://doi.org/10.1016/j.patcog.2020.107198

Kazemi, Z., and Zarrabi, H. (2017). Using deep networks for fraud detection in the credit card transactions. *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, 0630–0633.

Kingma, D. P., and Ba, J. (2014). *Adam: A method for stochastic optimization*. Retrieved from http://arxiv.org/abs/1412.6980

Krishnan, S., Wang, J., Wu, E., Franklin, M. J., and Goldberg, K. (2016). ActiveClean: Interactive data cleaning for statistical modeling. *Proc. VLDB Endow.*, *9*(12), 948–959. https://doi.org/10.14778/2994509.2994514

Lundberg, S. M., and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 4765–4774). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, *116*(44), 22071–22080. https://doi.org/10.1073/pnas.1900654116

Nazabal, A., Olmos, P. M., Ghahramani, Z., and Valera, I. (2020). *Handling incomplete heterogeneous data using VAEs*. Retrieved from https://arxiv.org/abs/1807.03653

Nguyen, M.-N., and Vien, N. A. (2018). *Scalable and interpretable one-class SVMs with deep learning and random fourier features*. Retrieved from https://arxiv.org/abs/1804.04888

Nguyen, Q. P., Lim, K. W., Divakaran, D. M., Low, K. H., and Chan, M. C. (2019). GEE: A gradient-based explainable variational autoencoder for network anomaly detection. *CoRR*, *abs/1903.06661*. Retrieved from http://arxiv.org/abs/1903.06661

Pang, G., Shen, C., Cao, L., and Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, *54*(2). https://doi.org/10.1145/3439950

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Paula, E. L., Ladeira, M., Carvalho, R. N., and Marzagão, T. (2016). Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 954–960. https://doi.org/10.1109/ICMLA.2016.0172

Pumsirirat, A., and Yan, L. (2018). Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine. *International Journal of Advanced Computer Science and Applications*, *9*(1). https://doi.org/10.14569/IJACSA.2018.090103

Ramamurthy, K. N., Vinzamuri, B., Zhang, Y., and Dhurandhar, A. (2020). *Model agnostic multilevel explanations*. Retrieved from https://arxiv.org/abs/2003.06005

Redyuk, S., Schelter, S., Rukat, T., Markl, V., and Biessmann, F. (2019). Learning to validate the predictions of black box machine learning models on unseen data. *Proceedings of the Workshop on Human-in-the-Loop Data Analytics*. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3328519.3329126

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. *CoRR*, *abs/1602.04938*. Retrieved from http://arxiv.org/abs/1602.04938

S., K. P. F. R. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. https://doi.org/10.1080/14786440109462720

Schreyer, M., Sattarov, T., and Borth, D. (2021). *Multi-view contrastive self-supervised learning of accounting data representations for downstream audit tasks*. arXiv. https://doi.org/10.48550/ARXIV.2109.11201

Schreyer, M., Sattarov, T., Borth, D., Dengel, A., and Reimer, B. (2018). *Detection of anomalies in large scale accounting data using deep autoencoder networks*. Retrieved from https://arxiv.org/abs/1709.05254

Schreyer, M., Sattarov, T., Schulze, C., Reimer, B., and Borth, D. (2019). *Detection of accounting anomalies in the latent space using adversarial autoencoder neural networks*. Retrieved from https://arxiv.org/abs/1908.00734

Schultz, M., and Tropmann-Frick, M. (2020). *Autoencoder neural networks versus external auditors: Detecting unusual journal entries in financial statement audits*.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. *CoRR*, *abs/1704.02685*. Retrieved from http://arxiv.org/abs/1704.02685

Takeishi, N. (2020). *Shapley values of reconstruction errors of PCA for explaining anomaly detection*. Retrieved from https://arxiv.org/abs/1909.03495

Venkataramanan, S., Peng, K.-C., Singh, R. V., and Mahalanobis, A. (2019). *Attention guided anomaly localization in images*. arXiv. https://doi.org/10.48550/ARXIV.1911.08616

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *International conference on machine learning proceedings*.

Walach, J. et al. (2020). Cellwise outlier detection and biomarker identification in metabolomics based on pairwise log ratios. *Chemometrics*, *34*(1). https://doi.org/https://doi.org/10.1002/cem.3182

Wedge, R., Kanter, J. M., Rubio, S. M., Perez, S. I., and Veeramachaneni, K. (2017). *Solving the "false positives" problem in fraud prediction*. Retrieved from https://arxiv.org/abs/1710.07709

Xu, H., Wang, Y., Jian, S., Huang, Z., Wang, Y., Liu, N., and Li, F. (2021). Beyond outlier detection: Outlier interpretation by attention-guided triplet deviation network. *Proceedings of the Web Conference 2021*, 1328–1339. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3442381.3449868

Yeh, I. C., and Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients.