technische universität dortmund

fakultät für informatik

Master thesis

Context Matters: Contextual Anomaly Detection in Cash Withdrawal Resource Distribution

Supervisors:
Prof. Dr. Emmanuel Müller
M. Sc. Simon Klüttermann,

Author: Nidhi Patel

May 4, 2025

Contents

1 Introduction

Financial inclusion serves as the principal component of economic empowerment, allowing individuals and communities to actively participate in the financial ecosystem. In an increasingly digitized economy, ensuring equitable access to cash withdrawal facilities remains a vital aspect of promoting financial inclusion. Facilities such as ATMs, bank branches, and cashback stores play a critical role in modern societies by bridging gaps in financial accessibility. This study aims to examine and highlight disparities in the accessibility of these resources across regions in Germany by focusing on their relationship with various socioeconomic factors.

Building on this foundation, contextual anomaly detection offers a sophisticated analytical framework to identify regions or demographic groups where the availability of financial resources deviates significantly from expected patterns. Unlike traditional anomaly detection methods, which solely focus on outlier detection based on raw data, contextual anomaly detection incorporates the influence of contextual variables. These variables, such as population density, unemployment rates, income levels, or urbanization, provide a richer understanding of the socioeconomic environment in which financial services operate.

For example, the absence of ATMs in a sparsely populated rural area may not be considered anomalous, given the lower demand. However, a similar absence in a densely populated urban area with high transaction volumes would be flagged as a significant anomaly. By incorporating such context, this approach helps differentiate between expected variations and genuine disparities in resource distribution.

This study explores and compares different frameworks for contextual anomaly detection to analyze the availability of bank branches, ATMs, and cashback resources (target variables) in relation to socioeconomic indicators (contextual variables). This research was conducted **in collaboration with Barkow Consulting GmbH** to analyze how Cash withdrawal distributions relate to the characteristics of each region and to uncover patterns that highlight areas of under-service or over-service, while also providing explanations for why these anomalies occur. This involves identifying the factors contributing to disparities, such as socioeconomic conditions or regional imbalances, and quantifying their impact on resource availability. These anomalies are not merely statistical outliers but represent meaningful deviations that may point to systemic inequities, inefficiencies, or areas of potential improvement.

By utilizing explainability techniques, such as SHAP Additive Explanations (SHAP), this study aims to make the detection process interpretable and actionable, allowing policymakers and stakeholders to address the root causes effectively. Ultimately, this

study aims to use contextual anomaly detection and explainability to provide a clear, interpretable understanding of financial resource distribution disparities.

1.1 Research Question and Contributions

As discussed before, the research questions to be answered are four folds:

- **Distribution Analysis:** How are Cash withdrawal networks distributed across different regions in Germany?
- **Contextual anomaly detection:** Where are the most significant blind spots (anomalies) in Germany's Cash withdrawal network? And how do these anomalies correlate with different Socioeconomic conditions?
- **Interpretability:** What is the relationship between the distribution of Cash withdrawals and key Socioeconomic factors? Which Socioeconomic variables have the strongest influence on financial service accessibility at the regional level?
- **Ideal Distribution Projection:** If there are clear blind spots in the distribution of cash withdrawals, how should an ideal distribution look like based on current socioeconomic indicators?

1.2 Structure of the Thesis

This thesis is organized into distinct sections. Section 2 briefly introduces traditional anomaly detection methods, providing the basis for developing contextual anomaly detection models in subsequent sections. Section 3 covers various machine learning models, evaluation criteria, and interpretability methods, which are essential for understanding the frameworks defined in this study. Section 4 formally defines the problem of contextual anomaly detection and establishes its theoretical foundation. Section 5 discusses different frameworks for addressing contextual anomaly detection. Section 6 describes the datasets and preparatory steps required to conduct the experiments outlined in the previous section. Section 7 then compares and evaluates the experimental results from proposed frameworks by means of performance metrics. Section 8 translates the results derived in previous section into real-world conclusions related to the cash withdrawal use case, while also providing insights on interpretability. Finally, Section 9 concludes the study and provides suggestions for future work.

2 Background: Anomaly Detection

Anomaly detection, also referred to as outlier detection, constitutes a fundamental task in data analytics that focuses on identifying observations, patterns, or events exhibiting substantial deviations from expected behavior (11). Such deviations often correspond to infrequent yet critical incidents, including financial fraud, cybersecurity breaches, mechanical failures, or medical abnormalities. The field further categorizes anomalies into two distinct types: global outliers that deviate significantly from the complete dataset's distribution, and local outliers that appear anomalous only within their immediate neighborhood of data points (23; 26).

Outlier detection is a long-standing concept with deep roots in statistical analysis. From a statistical perspective, anomalies are typically defined as observations that exhibit an exceptionally low probability of occurrence under a given probability distribution $P(x)$ for a dataset $\{x_1, x_2, \ldots, x_n\}$ (2). However, this traditional approach faces significant limitations, particularly in practical applications where specifying an accurate probability distribution $P(x)$ for real-world data proves challenging. These limitations have motivated the development of machine learning-based approaches that employ algorithmic techniques to address the shortcomings of conventional statistical methods (26).

Anomaly detection techniques can be distinguished based on their required level of supervision during the learning process. These approaches are broadly divided into three categories: **supervised**, **unsupervised**, and **semi-supervised** learning frameworks (57; 26). The **supervised** techniques rely on completely annotated datasets where each instance is classified as either normal or anomalous. While they perform well with sufficient labeled examples, their capability to detect new anomaly types is constrained. On the other hand, **unsupervised** methods examine unannotated data by spotting observations that substantially differ from common patterns, utilizing natural data properties like distribution density or cluster formations. The **semi-supervised** approaches combine aspects of both, working with a small set of labeled data alongside a larger unlabeled collection. Among these, **unsupervised** techniques prove particularly valuable for practical applications since they don't depend on labeled datasets, which are often scarce in real-world scenarios. This practical advantage makes them the preferred choice for our study.

2.1 Conventional methods for Anomaly Detection

As mentioned in the previous section, various statistical and algorithmic approaches to outlier detection exist, each with varying levels of supervision. Despite their differences in

operation, all these methods share the common goal of assessing the degree of outlierness in the data.

Certain anomaly detection techniques output continuous scores representing the probability of a data point being an outlier, whereas others produce discrete binary labels classifying observations as either normal or anomalous. The scoring-based methods facilitate the ordering of data points according to their degree of outlierness, allowing practitioners to determine how anomalous each observation appears to be. These scores can then be transformed into binary classifications through thresholding (11; 26). This section provides detailed descriptions of these approaches.

2.1.1 Distance-Based (Proximity-based)

Data points are separated using distance-based outlier detection techniques according to how far away they are from their neighbors. A data point is categorized as an outlier if it is significantly apart from its neighbors, and as a normal data point if it is close to them (42; 26). The example of the most common outlier identification technique under distance-based category is k-nearest neighbor (KNN) (19). The most common method for distances measure is the Euclidean distance. Since the method requires to calculate the distance between every single pairs of points within dataset which leads to higher computational complexity which is one of the biggest disadvantages of the method. Another disadvantage of given method is, the volume of the space grows exponentially with the increase in number of dimensions consequently making the data point more and more spread out. Since these methods relies solely on the distance to nearest neighbors, distance becomes less and less meaningful as the data space becomes sparse. This issue is also referred to as **Curse of Dimensionality** (49) which mostly exists in Distance, Density and Cluster based methods.

2.1.2 Density-Based (Proximity-based)

Density-based anomaly detection identifies outliers by analyzing data point distribution, assuming normal points cluster in dense areas while anomalies appear in sparser regions. It evaluates the "crowdedness" around each point, flagging isolated ones as outliers and dense ones as normal. Various algorithms calculate density differently, leading to diverse detection methods. For example, the Local Outlier Factor (LOF) compares a point's density to its neighbors, assigning high outlier scores to lower-density points (8). However, these methods struggle with high-dimensional data due to the "curse of dimensionality," where distances lose meaning and computations become complex.

2.1.3 Clustering-Based (Proximity-based)

Clustering methods work on the concept of partitioning or grouping the data points such that points belonging to same clusters are more similar to each other than points from different groups. It aims to maximize homogeneity within the groups and heterogeneity amongst different groups (25). These techniques can be further categorized based on how they partition the data and define **anomalousness** (26). One strategy assumes that **normal data points** are members of large, dense clusters, while **outliers** reside in smaller or sparser groupings (8). Another variant considers points that fail to associate with any cluster as outliers (21). A third perspective evaluates the **distance** of each point from the **cluster centroids** (i.e., geometric centers), designating those closest to centroids as normal and those farthest away as outliers (24; 26). Among these, **K-means clustering** stands out as a widely adopted technique. However, due to its reliance on computing distances between every point and all centroids, its computational complexity increases significantly with higher **dimensionality** or larger **datasets**.

2.1.4 Deviation Based

The fundamental concept behind type of anomaly detection method is, points that significantly deviate from the dataset's general characteristics or expected behavior are known as outliers. One such example under this category is Variance Based anomaly detection method (23). The method relies on statistical measures of **variance or spread** to detect outliers. The assumption is that normal data points follow a certain pattern or distribution whereas outliers are points that deviate significantly from this pattern consequently **causing an increase** in the overall variance of the dataset. By analyzing how much each data point contributes to the variance, we can identify outliers. Outliers are identified by **discarding points** and then studying how the **variance of the dataset changes** as a result (1). The point might be an anomaly if the variance drops, and vice versa. This method fails to detect outliers if they are in the middle of a dataset as it assumes that the outliers are the points which are furthest away in the dataset. Also, the method is not very efficient due to comparatively high time complexity (44).

2.1.5 Model-Based (Dependency based)

The primary concept behind these methods is to construct a model representing the structure of typical data distribution in a given dataset, with outliers defined as observations that do not fit the model (1). Neural networks and the statistical techniques mentioned before fall under the model-based category.

Statistical Methods

Statistical outlier detection includes a range of methods, from those that rely on strict assumptions about data distributions to completely data-driven techniques. **Parametric methods** (37) assume specific probability distributions, like the Gaussian distribution, and identify outliers as points that significantly differ from what is expected. While these methods are efficient, they can struggle when **real data doesn't fit the assumed distributions**. **Semi-parametric** methods offer a middle ground by combining some distributional assumptions with **more flexibility**, often using mixture models or robust estimation to handle more complex data patterns. **Non-parametric** (37; 26) methods are notable for making few assumptions about how the data is generated, allowing the **data itself to define what is considered normal**. These techniques, such as histogram analysis, kernel density estimation, learn patterns directly from the data, making them especially useful for complex, high-dimensional datasets where **strict distributional assumptions** may not apply.

Neural Networks

Neural network-based outlier detection leverages the power of **deep learning** to identify anomalies in complex datasets. These methods use neural networks, such as autoencoders or generative adversarial networks (GANs), to learn intricate patterns and representations of normal data during training. For instance, an autoencoder is trained to **reconstruct input data**, and points with **high reconstruction errors** are flagged as **outliers** since they deviate from the learned norm (45; 46). This approach excels in handling high-dimensional and unstructured data, like images or time series, where traditional methods often struggle. Neural networks can adaptively capture non-linear relationships and subtle patterns without requiring explicit assumptions about data distribution (10). However, they typically demand **large amounts of data** and computational resources for effective training. The **black-box** nature of these models can also make it challenging to interpret why a specific point is labeled as an outlier.

In this study, we mostly use **dependency-based approaches**, either alone or in combination with other techniques, as they are better suited for **contextual anomaly detection**. Given the **simplicity** of our data and the high importance of **model interpretability**, we avoid neural network-based models to maintain transparency.

2.1.6 Isolation-Based

All of the models mentioned previously are based on fundamental idea of defining normal data in the dataset and then classifying the odd data as outliers (26). Isolation-based outlier detection relies on a straightforward but effective idea that **anomalies are easier to**

separate from normal data points. This approach works by **recursively partitioning** the data space and measuring how quickly individual observations become separated from the rest of the dataset. Anomalous points typically **require fewer partitioning** steps to become **isolated**, as they often reside in sparser regions of the data space. Different algorithms can be used to perform this isolation, but they all aim to measure **how easily a point can be separated**, turning this into an anomaly score (31). This methodology excels at handling high-dimensional data where traditional distance-based methods struggle, and it generally requires fewer computational resources than density-based techniques. A major benefit is its flexibility, as it can detect outliers without needing specific assumptions about the underlying data distribution or requiring explicit distance metrics. A more detailed explanation of tree-based isolation techniques will be provided in section 3.3.

We can summerize all the methods for anomaly detection as follows:

Method	**Outlier definition**	**Output**
Distance-based	Calculates the distance between all points, the point that is significantly distant from its neighbors is identified as an outlier.	Score
Density-based	Points with densities that deviate from those of their surrounding areas are very likely to be anomalies.	Score
Cluster-based	Outliers are the points that don't fit into the big, dense clusters in the dataset.	Binary/Score
Deviation-based	Outliers are points that deviate significantly from this pattern consequently causing an increase in the overall variance of the dataset.	Binary/Score
Model-based	Constructs a model representing the structure of typical data distribution in a given dataset, outliers are defined as observations that do not fit the model.	Binary/Score
Isolation-based	Instead of screening normal cases first, these methods directly isolate outliers.	Score

Table 1: Summary of Outlier Detection Methods

3 Methodology

This section in details describes different machine learning models which will be useful for implementation of different approaches for contextual anomaly detection.

3.1 Quantile Regression Forests

Quantile Regression Forests (QRF) (38) extend traditional Random Forests (6) to estimate conditional quantiles instead of conditional means. Unlike standard regression trees, which yields a single prediction as output for a given input, QRF provides a full distributional view of the target variable.

A QRF consists of an ensemble of decision trees trained on bootstrapped samples of the data. Given a new observation x, each tree provides a set of observed values from its leaf node rather than a mean prediction. The final quantile estimation is computed by aggregating these values across all trees in the forest.

The conditional quantile function is estimated as:

$$\hat{Q}_\alpha(x) = \inf\{y : \hat{F}(y|x) \geq \alpha\}, \tag{1}$$

Here, α represents the **quantile level** (a value between 0 and 1) in Quantile Regression Forests (QRF). It indicates the specific quantile of the conditional distribution of the target variable that is being estimated (e.g., $\alpha = 0.5$ corresponds to the median). The QRF model predicts the value y such that the probability of the target variable being less than or equal to y is at least α. $\hat{F}(y|x)$ is the empirical **cumulative distribution function** of the response variable conditioned on context variables.

$$\hat{F}(y \mid X = x) = \hat{P}(X \leq x \mid Y = y) = \hat{E}(\mathbb{I}(X \leq x) \mid Y = y) \tag{2}$$

Each tree in the QRF partitions the feature space into regions where the conditional distribution of the target variable is approximately constant. By collecting all values within the terminal nodes, QRF can estimate any quantile of interest.

The advantages of QRF (16) include :

- The ability to model both linear and non-linear dependencies between features.
- Robustness to outliers and heteroscedasticity (variable spread of data).
- The capability to estimate the entire conditional distribution rather than just the mean.

3.2 Random Forest

Before moving on to concept of random forest, it is important to understand the concept decision trees which act as a fundamental part of random forest.

3.2.1 Decision trees

A **decision tree** is a sequential learning model that functions as a decision-support mechanism by applying a series of logical evaluations, where each test involves comparing a numerical feature against a threshold (7). The model classifies data instances by querying the attributes associated with the current node and the input instance. Each internal node in the tree contains a single question, and each possible answer routes to a corresponding child node, forming a **hierarchical structure** that resembles a tree. During training, new decision nodes are added iteratively to improve the classification accuracy of the model on the training data. This is achieved by recursively partitioning the data based on selected features, leading to increasingly refined subgroups that are integrated into a comprehensive tree. At each step, the best split is determined by either maximizing **information gain** or minimizing **impurity**. The ideal objective is to produce nodes with uniform class labels using as few splits as possible. Measures such as **entropy** and the **Gini index** are commonly used to assess impurity (41).

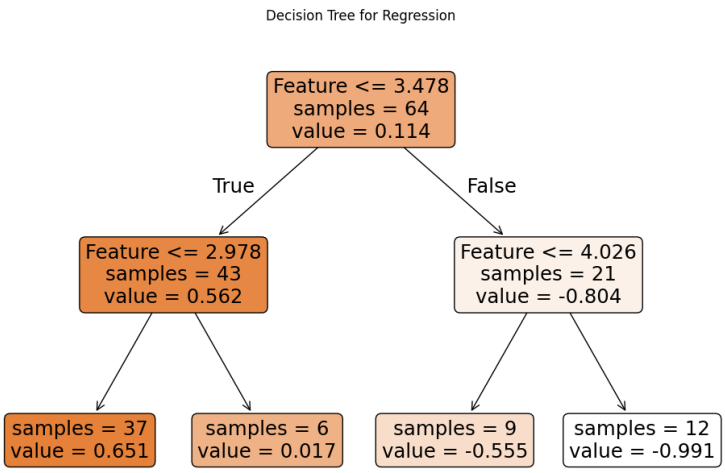


Figure 1: Single decision tree for regression

When classifying a new instance, it begins at the **root node** and traverses down the tree by following branches based on the outcomes ("True" or "False") of internal node evaluations until it reaches a **leaf node**, as illustrated in Figure 1 (generated using sklearn (39) package). The instance is then assigned the output value associated with that terminal node. Due to their transparent structure, **decision trees** are considered highly

interpretable and **intuitive**, making it easy to trace and understand the reasoning behind a given prediction.

3.2.2 Random Forest regressor

Random Forest Regressor is an ensemble learning method used for regression tasks. It combines the predictions of multiple decision trees to improve accuracy and reduce overfitting (6). Each tree in the forest is trained on a random subset of the training data and a random subset of features which introduce diversity among the trees.

For a given input vector x, the predicted output $\hat{y}$ is the average of the predictions from all individual trees as described by Figure 2:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} f_t(x), \tag{3}$$

where T is the total number of trees, and $f_t(x)$ is the prediction of the t-th tree. During training, each tree f_t is constructed by recursively splitting the data based on features that minimize the mean squared error (MSE) or another loss function.

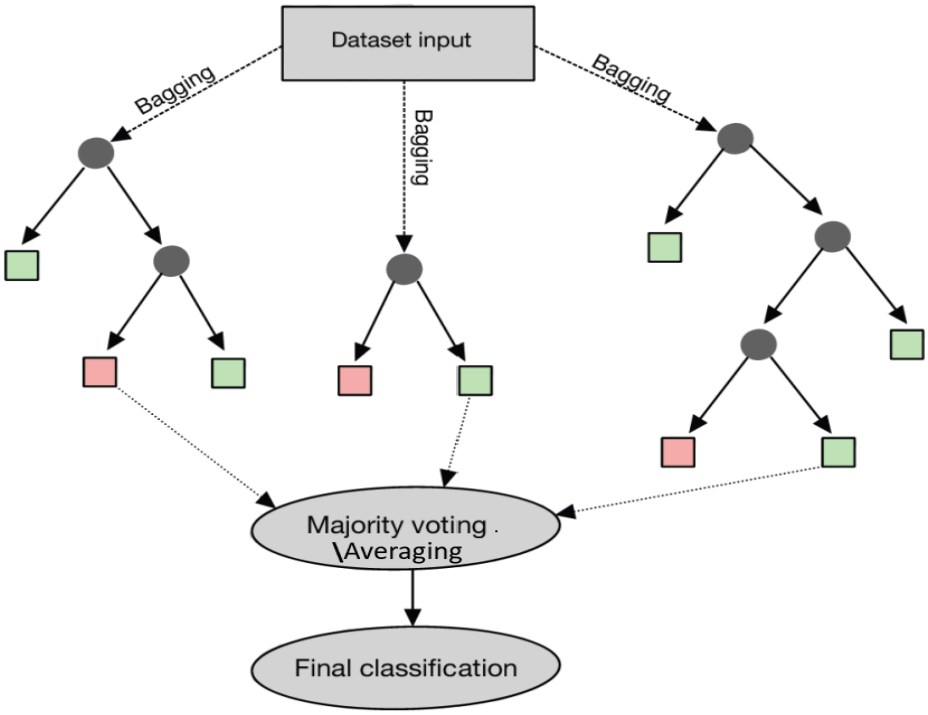


Figure 2: Random forest with three trees and a max depth of three (26)

The ensemble approach ensures robustness and generalization, making random forests a powerful tool for regression problems.

The Random Forest was chosen as the primary estimation model in the first place over other regression models due to following advantages (5):

Better Generalization: A diverse collection of trees improve generalization on unknown data, which boosts performance on complex or high-dimensional regression problems.

More robust towards noise and outliers: Due to random sub sampling and averaging the predictions across multiple trees, the model is less sensitive to the potential noise in dataset.

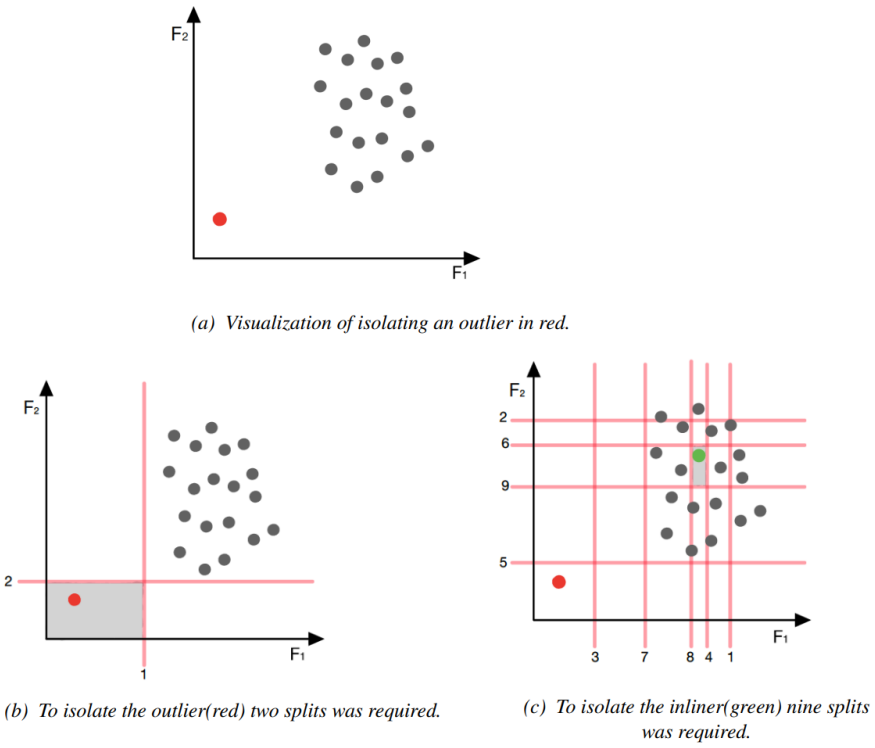
Interpretability: It provides interpretability in terms of feature importance, being transparent about model's decision making process by rule extraction. Additionally, SHAP values can also be integrated for local interpretability.

3.3 Isolation Forest

As discussed in Section 2.1.6, the **Isolation Forest (iForest)** is a **tree-based** anomaly detection method that identifies anomalous data points through a strategy of random partitioning (31). The core principle of **iForest** is to recursively divide the dataset by selecting a random feature and then choosing a random split value between its minimum and maximum values, resulting in the construction of multiple **isolation trees (iTrees)**. These trees are subsequently employed to evaluate the dataset, with the aim of **isolating individual observations**. Each observation may require a different number of partitions to become isolated, and this is reflected in the **path length**, which varies according to the splits encountered in each tree.

Typically, data points that are **isolated** near the **root node** with shorter **path lengths** are more likely to be **outliers**, as they require fewer partitions to be separated from the rest of the data. This suggests that such points exhibit more extreme characteristics compared to those requiring more splits for isolation, as illustrated in Figure 3.

The **iForest** algorithm consists of two main phases: (1) **Training** and (2) **Testing**. During the training stage, a collection of **isolation trees (iTrees)** is constructed. In the testing phase, each observation is evaluated across all trees to calculate its individual **path lengths**. These are then aggregated to compute an **anomaly score**, typically defined as the average path length across the ensemble of trees.



(a) Visualization of isolating an outlier in red.

(b) To isolate the outlier(red) two splits was required.

(c) To isolate the inliner(green) nine splits was required.

Figure 3: Two dimensional dataset with outlier in red (26)

1) Training stage:

During the **training phase**, the input dataset is recursively partitioned to build the individual **isolation trees (iTrees)**, ultimately forming the complete **Isolation Forest (iForest)**, until each data point is successfully **isolated**. The dataset used for training can either be the entire dataset under consideration or randomly selected **subsamples** (ϕ). Unlike many **classification techniques**, where training and evaluation are performed on separate datasets, **iForest** uses the same data for both building the model and evaluating anomaly scores.

To improve efficiency, a maximum **tree height** l is imposed, which approximates the **average tree height**. Since **anomalies** are often isolated in fewer steps, further splitting beyond this height is generally unnecessary. The expected path length is given by $l = \lceil \log_2 \psi \rceil$, where ψ is the subsample size, based on principles from **graph theory**.

Being an **ensemble method**, **Isolation Forest** generates multiple iTrees due to its inherent randomness. Empirical evidence shows that the average path length stabilizes before $t = 100$, making the use of 100 trees sufficient for most datasets. A detailed outline of the training phase is provided in Algorithms 1 and 2.

Algorithm 1 `Building an Isolation Forest`(X, t, ψ)

Inputs: X - dataset, t - number of isolation trees, ψ - sample size per tree
Output: A collection of t isolation trees (iTrees)
Initialize an empty set `Forest`
Compute the maximum depth $l = \lceil \log_2 \psi \rceil$
for $i = 1$ to t **do**
    Draw a random subset $X_i \subset X$ of size ψ
    Build an iTree from X_i with initial depth 0 and height limit l
    Add the constructed iTree to `Forest`
end for
return `Forest`

Algorithm 2 `Constructing a Single Isolation Tree`(X, e, l)

Inputs: X - input dataset, e - current depth, l - maximum tree depth
Output: An isolation tree node
if $e \geq l$ **or** $|X| \leq 1$ **then**
    return `LeafNode`{Size $\leftarrow |X|$}
else
    Define Q as the set of all features in X
    Choose a feature q randomly from Q
    Select a random split point p within the range of values for feature q in X
    Partition X into:
        $X_{\text{left}} \leftarrow \{x \in X \mid x_q < p\}$
        $X_{\text{right}} \leftarrow \{x \in X \mid x_q \geq p\}$
    return `InternalNode`{
            LeftChild $\leftarrow$ `iTree`$(X_{\text{left}}, e + 1, l)$,
            RightChild $\leftarrow$ `iTree`$(X_{\text{right}}, e + 1, l)$,
            SplitFeature $\leftarrow q$,
            SplitPoint $\leftarrow p$
    }
end if

2) Testing stage:

In the **evaluation phase**, each observation in the dataset is assigned an **anomaly score**. As observations traverse all **iTrees** in the **iForest**, the score is computed using the **expected path length** $E(h(x))$ rather than the raw path length $h(x)$ alone, since some paths may have been prematurely terminated due to the imposed maximum tree height.

The anomaly score s for an observation x is defined as:

$$s(x, n) = 2^{\frac{-E(h(x))}{c(n)}} \tag{4}$$

where $E(h(x))$ is the average path length across the iTrees, and $c(n)$ is the average path length in a **Binary Search Tree (BST)** of size n, used for normalization.

A score closer to 1 suggests a strong **anomaly**, while scores near 0 indicate **normality**.

The algorithm 3 step by step depicts the testing phase.

Algorithm 3 Isolation Forest path length and anomaly score calculation

Inputs: X - dataset with n instances, T - an iTree, e - current path length; to be initialized to zero when first called

Output: Anomaly scores for each instance in X

procedure PATHLENGTH(x, T, e)
  if T is an external node **then**
    return $e + c(T.size)$ ▷ $c(.)$ is the average path length adjustment
  end if
  if $x_a < T.splitValue$ **then**
    return PATHLENGTH($x, T.left, e + 1$)
  else
    return PATHLENGTH($x, T.right, e + 1$)
  end if
end procedure
procedure COMPUTESCORES($X, Forest$)
  for each $x \in X$ **do**
    $h(x) = \frac{1}{t} \sum_{i=1}^{t}$ PATHLENGTH($x, T_i, 0$)
    $s(x) = 2^{-\frac{h(x)}{c(\psi)}}$ ▷ Anomaly score
  end for
  return $\{s(x) : x \in X\}$
end procedure

3.4 Methods for performance evaluation

We will use several well recognized metrics to get feedback from our tests and gauge how well a model works on various data. We will be able to comprehend the model's performance for various data distributions and its sensitivity while altering the input parameters by using the various scoring measures discussed in following sections.

3.4.1 Confusion matrix

When the true values are known, the confusion matrix is a table that is used to assess how well classification models perform. The matrix consists of four components: true negative, false positive, false negative, and true positive as mentioned in Table 2. The sections also demonstrate how its four components are used to evaluate performance in different ways.

	Predicted	
Actual	**Positive**	**Negative**
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Table 2: Confusion matrix for performance evaluation

Different measures for evaluation

Accuracy

Accuracy is simply a measure of how likely the classifier produce correct predictions out of all predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Recall

The number of data points of positive class that are predicted to be positive is known as recall, or true positive rate (TPR). For applications where the cost of false negatives is high (e.g., disease detection), recall is an important metric.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

Precision

The precision indicates how many predictions are actually positive out of all the predicted positives. It is useful for applications where the rate of false positives is of higher concern than false negatives(e.g., spam detection).

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

3.4.2 Area Under the Receiver Operating Characteristics

Area Under the Curve (AUC), derived from the **Receiver Operating Characteristic (ROC)** curve, quantifies the likelihood that a classifier ranks a randomly chosen

positive instance higher than a negative one (50). The ROC curve illustrates the trade-off between the **True Positive Rate (TPR)** also called **recall** and the **False Positive Rate (FPR)** across varying thresholds. It originates at point (0, 0) and ends at (1, 1), with a diagonal line between them representing a **random classifier**. In contrast, a **perfect classifier** follows the top-left boundary of the ROC space, achieving maximal separation between classes. A competent classifier will expand over the diagonal line.

Similarly, the **Area Under the Precision-Recall Curve (AUPRC)** is a widely adopted metric, especially suitable for **imbalanced datasets** common in **outlier detection** tasks (12). It focuses on the balance between **precision** and **recall** at different thresholds, excluding **True Negatives (TN)** from consideration. This makes AUPRC more informative than AUC when the positive class is rare.

In a precision-recall curve, all observations are initially classified as negative, starting at the point where **precision = 1** and **recall = 0**. As the threshold decreases, more instances are predicted as positive, gradually moving the curve toward full recall. An ideal model achieves a perfect AUPRC score of 1, indicating that it can correctly identify all positive instances with no false positives at some threshold.

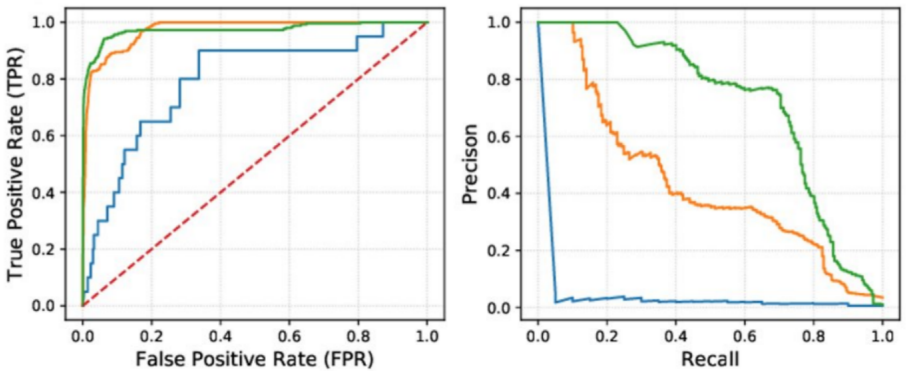


Figure 4: An illustration of the ROC curve (left) and Precion-Recall curve (right) for different classifier results (26).

3.5 Interpretablity

Interpretability in machine learning refers to the ability to understand and explain how a model makes predictions. It helps identify which features influence decisions (globally) and why a specific prediction was made (locally). Methods like SHAP, and feature importance scores provide insights into model behavior, ensuring transparency, trust, and compliance with regulations.

3.5.1 Shapley additive explanations (SHAP)

Cooperative game theory is where SHAP values first originated. In essence, a Shapeley value is a player's marginal contribution after accounting for all potential player coalitions or combinations in the game with other players (48; 29). If S is a coalition of M such players, then $v(S)$ is a value function that represents the pay-off of the coalition. This can be represented as,

$$\phi_j(v, M) = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|!(|M| - |S| - 1)!}{|M|!} \left(v(S \cup \{j\}) - v(S)\right) \tag{8}$$

Where, $\phi_j(v, M)$ is the Shapley value of the j^{th} of M players, v is the Shapley value function, M is the total number of players in the game and $M \setminus \{j\}$ denotes all the players in the game, the j^{th} player excluded.

This phenomena is directly applicable to machine learning models (33), where a model $f(X)$ trained on M predictors represents the value function, $v(S)$. After that, $\phi_j(v, M)$ becomes the contribution or significance of the model's j^{th} of M predictors.

Shapley additive explanations (SHAP) is a technique to explain the individual predictions of a model. This is executed by allocating attribution scores to each feature or variable X_j, $j = 1, 2, \ldots, M$using Shapley values. As the acronym indicates, SHAP values represent the degree by which each variable shifts the value of the model prediction up or down from its base value, which is the mean of all the predictions,$E[f(X)]$.

For a given prediction $\hat{y}$, the SHAP value ϕ_f for feature f represents the change in the expected model prediction when f is included versus excluded.

$$\hat{y} = \phi_0 + \sum_{f=1}^{M} \phi_f,$$

where, ϕ_0 is the baseline prediction (average prediction over the dataset ($E[f(X)]$), ϕ_f is the SHAP value for feature f derived from equation 8, M is the total number of features.

4 Contextual Anomaly Detection

Even though applications of contextual anomaly detection are widespread, the concept itself is still comparatively unexplored. In a compilation of existing research, a number of statistical methods for identifying outliers have been suggested (1; 11). In addition to statistical methods, other methods include density, distance, clustering, and anomaly-detection methods based on machine learning. But a major issue with most conventional anomaly-detection techniques is that they have a tendency to overlook the context of data generation. Incomplete and erroneous results may arise from this disregard for context. Taking the context into account is especially important while looking for irregularities (51).

For instance, a heart rate of more than 100 beats per minute, is abnormal for a grown-up at rest but normal for a baby or an adult working out at the gym. In these situations, characteristics like age and level of physical activity do not necessarily indicate abnormality straight away rather, they offer insight into whether the heart rate is within the normal range in a certain context.

In contextual or conditional anomaly detection, an observation is considered anomalous if it significantly deviates from the group of observations sharing similar contextual information (51). These groups are referred to as a reference group. The primary objective of this method is to find objects that are anomalous within certain contexts but can be identified as normal globally. For this purpose, all the attributes of dataset are divided into two disjoint groups namely: Contextual attributes and Behavioral attributes (11). Contextual attributes are set of parameters which are used to define the context and determine the reference groups which serves as means for comparison, whereas behavioral attributes contain the parameters of interest within which the anomalies are supposed to be detected. The main idea behind the contextual anomaly detection is to measure whether the behavioral value of given object deviates significantly with respect to behavioral value of its reference group.

The process of contextual anomaly detection can be efficiently implemented by using combination of different conventional anomaly detection methods (1) mentioned in Section 2. Various combinations of different methods such as proximity based, deviation based and dependency based are being used in different order to fulfill the purpose of contextual anomaly detection. Additionally, similar to conventional anomaly detection, the problem of contextual anomaly detection can be perceived from both a global and local standpoint (40). The key difference between the two approaches is that the local method uses a reference group of instances similar to the given observation, while the global method uses the entire dataset as the reference group. While the local approach efficiently captures context-specific variations, the global approach is valuable for mod-

eling dependencies when set of objects with similar contextual values are sparse. All the frameworks will be described in detail in following sections (30).

4.1 Problem Formulation

Before delving into the methodological details, it is essential to establish the fundamental **notations** and **terminologies** that will be consistently used throughout this study to ensure clarity and coherence of the key concepts.

Consider a collection of objects, where the i^{th} object is represented as:

$$z^{(i)} = \begin{pmatrix} x^{(i)} \\ y^{(i)} \end{pmatrix} = \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_C^{(i)}, y_1^{(i)}, y_2^{(i)}, \ldots, y_B^{(i)}\right)^T$$

Here, $z^{(i)}$ denotes the full **attribute vector**, consisting of the **contextual attribute vector** $x^{(i)}$ and the **behavioral attribute vector** $y^{(i)}$. We assume that $x^{(i)}$ spans C dimensions, while $y^{(i)}$ spans B dimensions. Based on this structure, the complete dataset is defined as $Z = \{z^{(1)}, z^{(2)}, \ldots, z^{(N)}\}$, where N is the total number of objects. Among these, let O represent the subset of **contextual outliers**.

Provided the dataset Z and a reference group R, the objective is to derive a **contextual outlier score** S_i (with respect to contextual attribute vector $(x_1^{(i)}, x_2^{(i)}, \ldots, x_C^{(i)})$) for each object i, such that objects in the outlier set O receive significantly higher scores than the rest.

5 Approaches for Contextual Anomaly Detection

As discussed in Section 4, contextual anomaly detection can be approached from global or local perspectives, often combining different conventional methods. This section examines these frameworks, detailing their pros and cons. Eventually, a hybrid approach will be proposed to achieve the best of both worlds, integrating their strengths and addressing limitations.

5.1 Local approach

The local approach for contextual anomaly detection focuses on identifying anomalies within specific, localized contexts or subgroups of the data. Unlike the global approach, which considers the entire dataset as a whole, the local approach examines smaller and contextually more similar subsets of the data to detect deviations that may not be apparent at a global level. This method is particularly useful when the points are comparatively dense in context (given observation has comparatively higher numbers of similar data points) or anomalies only manifest within specific regions or groups of the data. One way to achieve the same is by combining proximity (e.g., k-nearest neighbors (19)) and dependency (e.g., Quantile Forest (16)) based anomaly detection methods described in the following section.

5.1.1 Quantile Forest Contextual Anomaly Detection (QCAD)

To elaborate further on above mentioned approach in technical terms, contextual attributes are first sorted or arranged using proximity-based method to form a different group of instances which shares the similar characteristics in terms of contextual attribute values. Followed by applying dependency-based approach on each group to model the relationship between contextual and behavioral features, exclusively specific to given group of instances. Finally, as per definition of dependency-based outlier detection, if the object deviates significantly from the expected behavior as per modeled relationship within its respective group, it identifies as an anomaly with respect to the group (set of similar instances) which can be also referred to as contextual anomaly. This approach is inspired by the work presented in the paper (28).

The problem statement in the section indicates a three step process to proceed with the task of contextual anomaly detection: (1) context generation, (2) Relationship modeling, and (3) Anomaly score computation

1. **Context Generation:** Identifying reference groups for each data point based on contextual similarity.

2. **Relationship modeling:** Using Quantile Regression Forests (QRF) to model dependencies between contextual and behavioral features.
3. **Anomaly score computation:** Assign anomaly scores according to the degree of deviation from the expected values derived from the modeled relationships based on reference group.

1. Context Generation

Contextual features define a reference group for each data point using similarity measures. Given a dataset $\mathbf{Z}$ with C contextual features $\mathbf{x} = \{x_1, x_2, \ldots, x_C\}$ and B behavioral features $\mathbf{y} = \{y_1, y_2, \ldots, y_B\}$, similarity is computed using **Gower's distance** (18):

$$d(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) = 1 - \frac{1}{C} \sum_{p=1}^{C} s_p^{(ij)}, \tag{9}$$

where $s_p^{(ij)}$ is the partial similarity for feature p:

- **Numerical features:** $s_{ij}^{(p)} = 1 - \frac{|x_p^{(i)} - x_p^{(j)}|}{\max(x_p) - \min(x_p)}$.
- **Categorical features:** $s_{ij}^{(p)} = \delta(x_p^{(i)}, x_p^{(j)})$, where $\delta(a, b) = 1$ if $a = b$, else 0.

Gower's distance was chosen as the primary distance measure for our mixed-type datasets because it preserves the semantics of each data type while handling numerical, categorical, and ordinal variables efficiently and without the need for data transformation.

A reference group $R(\mathbf{z}^{(i)}, k)$ is determined using k-nearest neighbors (k-NN) based on contextual similarity.

2. Relationship modeling (Using Quantile Regression Forests)

The fundamental idea behind this method is to estimate the deviation of an object's behavioral values within a given context using uncertainty quantification around predictions, where the predictions are assumed to capture 'normal' behavior. Higher uncertainty indicates a higher deviation within the context and thus a higher degree of anomalousness.

To model dependencies between contextual and behavioral features, we use **Quantile Regression Forests (QRF)** as described in Section 3.1, which estimates the conditional quantiles (α) for q^{th} behavioral features (y_q) conditioned on the contextual features (x):

$$Q_\alpha(y_q | \mathbf{X} = \mathbf{x}) = \inf\{y_q : P(Y_q \leq y_q | \mathbf{X} = \mathbf{x}) \geq \alpha\}. \tag{10}$$

Here, Quantile Regression Forests as a dependency-based approach offers significant advantages over similar methods. A major advantage is its ability to capture complete conditional distributions rather than depending on mean-based estimates.

3. Anomaly score computation

This anomaly score computation method evaluates whether a behavioral feature deviates significantly from its expected range based on conditional percentiles. Using the difference between the upper and lower conditional quantiles the width of the percentile interval for each behavioral feature b_q is computed initially. An extrapolation-based anomaly score is assigned based on how far it is from the extreme quantiles Q_{low} and Q_{upper} if the observed value b_q falls outside this range which is further normalized by the interquartile range. Finally, the overall anomaly score for an instance x_i is obtained by summing the individual scores across all behavioral features.

For each behavioral feature $b_q \in \mathbf{y}^{(i)}$, the **percentile interval width** is computed as:

$$w_q^{(i)} = Q_{(\alpha+0.01)}(b_q|\mathbf{x}^{(i)}) - Q_\alpha(b_q|\mathbf{x}^{(i)}). \tag{11}$$

If the observed value $b_i^{(q)}$ falls outside the predicted range, an extrapolation-based anomaly score is assigned:

$$s_q^{(i)} = \begin{cases} 1 + \frac{Q_{low} - b_q^{(i)}}{Q_{75} - Q_{25}} \cdot \max(w_q^{(i)}), & \text{if } b_q^{(i)} < Q_{low} \\ 1 + \frac{b_q^{(i)} - Q_{upper}}{Q_{75} - Q_{25}} \cdot \max(w_q^{(i)}), & \text{if } b_q^{(i)} > Q_{upper} \\ w_q^{(i)}, & \text{otherwise} \end{cases} \tag{12}$$

The final **anomaly score** for an instance $\mathbf{z}^{(i)}$ is computed as:

$$S_i = \sum_{q=1}^{Q} s_q^{(i)}. \tag{13}$$

Where, Q is the count of behavioral variables.

5.1.2 Algorithm: Quantile-based Contextual Anomaly Detection

The proposed QCAD algorithm follows a structured approach to detect contextual anomalies. The formal algorithm of the same is mentioned in the Algorithm 4 :

Algorithm 4 Quantile-based Contextual Anomaly Detection (QCAD)

Require: Dataset $\mathbf{Z}$, Number of nearest neighbors k, Number of trees T, Quantile levels Q

Ensure: Anomaly scores for all data points

Initialize empty anomaly score list S
for each object $\mathbf{z}^{(i)} \in \mathbf{Z}$ **do**
    Compute reference group $R(\mathbf{z}^{(i)}, k)$ using Gower's distance
    for each behavioral feature $b_q \in \mathbf{y}^{(i)}$ **do**
        Train a Quantile Regression Forest (QRF) using set of contextual features from reference group $R(\mathbf{x}^{(i)}, k)$
        Compute conditional quantiles $Q_\alpha(b_q|\mathbf{x} = \mathbf{x}^{(i)})$
        Compute interval width $w_q^{(i)} = Q_{(\alpha+0.01)}(b_q|\mathbf{x}^{(i)}) - Q_\alpha(b_q|\mathbf{x}^{(i)})$
        if $b_q^{(i)}$ falls outside estimated quantile range **then**
            Assign extrapolated anomaly score $s_q^{(i)}$ (as per eq 12)
        else
            Assign normal anomaly score $s_q^{(i)} = w_q^{(i)}$ (as per eq 11)
        end if
    end for
    Compute final anomaly score $S_i = \sum_{q=1}^{Q} s_q^{(i)}$
    Append $(\mathbf{z}^{(i)}, S_i)$ to anomaly score list S
end for
return anomaly scores S

5.1.3 Advantages of local approach

1). Better Handling of Data Heterogeneity: Detects anomalies within specific subgroups, accounting for natural variations rather than assuming uniform behavior across the entire dataset.

2). Detects Anomalies in Specific Contexts: Identifies deviations that are only anomalous within a narrow subgroup. Without missing subtle anomalies that might be overshadowed by broader patterns.

5.1.4 Disadvantages of local approach

1). High Sensitivity to Context Definition: The effectiveness of defined approach highly depends upon how well the contextual groups are defined as, sometimes poorly chosen context may lead to misidentification of normal points and anomalies.

2). Limited Generalization: The framework may struggle with detecting anomalies that exist across multiple contexts or on a broader scale, as it focuses only on localized deviations.

3). Vulnerability to Sparse Contexts: The model might struggle to establish reliable normal behavior if a given observation has very few similar data points in the context, which leads to unstable or unreliable anomaly scores.

Some of these limitations could easily be overcome by the global approach which will be discussed in details in following section.

5.2 Global approach

In contrast to the local approach to anomaly detection which focuses on the immediate neighborhood or context of a data point, the global approach for contextual anomaly detection focuses on exploring the relationships and dependencies between different data points across the entire dataset to identify anomalies. This approach can detect anomalies that deviate from the expected global behavior by analyzing how data points influence one another in a big picture even if they appear normal in a localized context. For example, regions where ATM availability significantly deviates from expected patterns when analyzing nationwide dependencies rather than just local conditions.

While local approaches are effective at identifying point anomalies or deviations within a small context, they tend to miss anomalies that only become apparent when considering the overall dependencies and interactions in the dataset. For instance, a local approach might fail to detect a subtle but significant variation in global trends or relationships, which a dependency-based global approach would capture straight away. Thus, while local methods are computationally efficient and suitable for detecting isolated anomalies, global approaches provide a deeper understanding of anomalies by considering the interrelated tendency of the data.

This approach considers the global structures and interactions within the data by leveraging dependency-based techniques such as probabilistic dependencies or machine learning models as mentioned in Section 2.1.5 to capture global patterns. The details of the same are discussed in following sections.

5.2.1 Dependency based contextual Anomaly Detection (DepCAD)

As mentioned in Section 5.2, the method exploits the dependency among variables. This approach was initially proposed in the paper (32). The core idea involves first uncovering the common dependency structures shared across the majority of objects, and then assessing the anomalousness of each object based on how well it follows these dependencies. Objects that exhibit substantial deviation from the typical dependency patterns are flagged as anomalies. A natural way to quantify such dependency deviation for a given

object is by computing the distance between its actual values and the corresponding expected values, where the latter are inferred from the established dependency relationships.

A key challenge that arises in real-world datasets is that typically only a specific subset of variables significantly contributes to the underlying data generation process of the target variable. The remaining (irrelevant) variables not only have minimal impact on the prediction values, but may also degrade prediction accuracy by introducing additional noise. Therefore, it becomes essential to accurately select the subset of variables that best captures the relevant dependencies, especially given the vast number of possible combinations in high-dimensional settings.

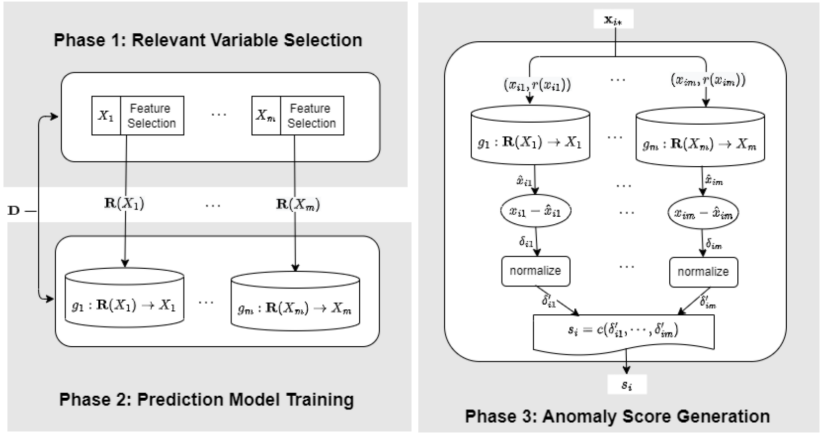


Figure 5: Dependency based contextual anomaly detection framework (32)

As illustrated in Figure 5, the method mainly consists three phases: 1) Relevant variable selection, 2) Expected value estimation and 3) Anomaly Score Generation.

1) Relevant variable selection

In order to fulfill the purpose of relevant variable subset selection which efficiently captures the dependency between contextual and target (behavioral) variables, we introduce the concept of **Markov Blanket (MB)** (58).

The Markov Blanket is a primary concept in probability theory and machine learning in the context of Bayesian Networks and dependency analysis specifically. It provides a way to identify the minimal set of variables that completely shields or isolates a target variable from the influence of all other variables in a dataset. In other words, the Markov Blanket of a target variable includes all the variables that are directly relevant to it such that the target variable becomes conditionally independent of all other variables outside this set.

Formally, a Bayesian Network is represented as (G, P), where $G = (V, E)$ is a **Directed Acyclic Graph (DAG)** (52) with nodes V (random variables) and edges E (dependencies). The joint probability distribution $P(V)$ factorizes as:

$$P(V) = \prod_{X \in V} P(X \mid \text{Pa}(X)), \tag{14}$$

where, $\text{Pa}(X)$ are the parents of X.

$\text{MB}(X)$ consists of:

- **Parents (Pa(X))**: Variables directly influencing X. Mathematically, $Y \in \text{Pa}(X)$ if there exists a directed edge $Y \to X$ in the graph G.
- **Children (Ch(X))**: Variables directly influenced by X. Formally, $Z \in \text{Ch}(X)$ if there exists a directed edge $X \to Z$ in G.
- **Spouses (Sp(X))**: Variables sharing a common child with X. These are other parents of X's children, i.e., $W \in \text{Sp}(X)$ if there exists a variable Z such that $W \to Z$ and $X \to Z$.

Given $\text{MB}(X)$, X is conditionally independent of all other variables:

$$P(X \mid V \setminus \{X\}) = P(X \mid \text{MB}(X)). \tag{15}$$

Thus, $\text{MB}(X)$ is the smallest set of variables needed to fully capture the dependencies of X, making it essential for efficient probabilistic inference and dependency analysis.

2) Expected value estimation

This phase focuses on estimating the expected value of a variable in a given object using the variables selected in a prior phase. The function (or relationship) is derived using a prediction model. A prediction model is built to predict the expected value of each target variable using the selected relevant variables (Markov Blankets) as predictors.This allows us to break down the anomaly detection problem into distinct classification or prediction problems. The verity of prediction models can be chosen to suit the requirements of the data. For our application we are opting for **Random Forest Regressor** models explained in Section 3.2. Therefore, the final equation representing the process in this phase would be

$$\hat{Y_j} = g_j(MB(Y_j)) \tag{16}$$

Where, $g_j()$ is Random Forest model trained using $MB(Y_j)$, set of relevant features (Markov Blanket) for j^{th} target variable.

3) Anomaly score generation

During this stage, a combination function is applied across value-wise deviations to obtain the vector-wise deviation or anomaly score.

Value-wise Deviation:

Given an object $\mathbf{z}^{(i)}$, its value-wise deviation (δ'_{ij}) with respect to variable j is defined as:

$$\delta_j'^{(i)} = |y_j^{(i)} - \hat{y}_j^{(i)}| \tag{17}$$

where, $y_j^{(i)}$ is the observed value of j^{th} target variable for object $\mathbf{z}^{(i)}$, and

$$\hat{y}_j^{(i)} = g_j(\mathbf{X}' = \mathbf{x}') \tag{18}$$

is the expected value of y_j for object $\mathbf{z}^{(i)}$, estimated using the function $g_j()$ based on the values of other variables $\mathbf{X}' \subseteq \mathbf{X} \setminus \{X_j\}$. where, X_j is set of irrelevant contextual variables.

Vector-wise Deviation:

The vector-wise deviation of object $\mathbf{z}^{(i)}$ is the aggregation of value-wise deviations for all its respective variables, calculated using an aggregation function:

$$\delta^{(i)} = \text{aggregation}(\delta'^{(i)}{}_1, \ldots, \delta'^{(i)}{}_{(B)}) \tag{19}$$

Assuming the target variable set of the given dataset as $\{1, 2, \ldots, B\}$.

For this study we are considering *average* function as *aggregation* function.

According to the definitions given above, value-wise deviation assesses how well an object complies with the dependencies with respect to a particular variable, while vector-wise deviation assesses how well an object complies with the dependencies overall.

5.2.2 Algorithm: Dependency based Contextual Anomaly Detection method

Algorithm 5 summerise the phases mentioned in the Section 5.2.1 for contextual anomaly detection.

The obtained value-wised deviations need to be normalized prior to computing vector-wise deviations. Especially, for each object on each target variable y_j, δ'_{ij} is normalized as the Z-score using the mean and standard deviation of δ'_j. After normalization, negative values indicate the small deviations. Since large deviations are of interest to us, the vector-wise deviation is calculated by adding the positive normalized value-wise deviations in the way

Algorithm 5 Dependency based Contextual Anomaly Detection (DepCAD)

Require: $\mathbf{Z}$, a dataset with N objects, where each object $\mathbf{z}^{(i)} = (\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ consists of contextual attributes $\mathbf{x}^{(i)}$ with C dimensions and behavioral attributes $\mathbf{y}^{(i)}$ with B dimensions; k, the number of anomalies to output

Ensure: Top-k detected anomalies

Initialize deviation matrix $\Delta_{N\times(C+B)}$
for each attribute $X_j \in \{\mathbf{x}_1, \ldots, \mathbf{x}_C, \mathbf{y}_1, \ldots, \mathbf{y}_B\}, j \in \{1, \ldots, C+B\}$ **do**
    Discover $MB(Y_j)$ using fast-IAMB algorithm ▷ Relevant variable selection
    Train a prediction (Random Forest) model g_j: $\hat{Y}_j = g_j(MB(Y_j))$
    for each object $\mathbf{z}^{(i)} \in \mathbf{Z}, i \in \{1, \ldots, N\}$ **do**
        Predict $\hat{y}_{ij}$ with g_j using Equation 18
        Compute ${\delta'^{(i)}}_j$ using Equation 17 ▷ Value-wise deviation
    end for
end for
Normalize Δ ▷ Normalization
for each object $\mathbf{z}^{(i)} \in \mathbf{Z}, i \in \{1, \ldots, N\}$ **do**
    Compute anomaly score S_i using Equation 19 ▷ Vector-wise deviation
end for
Output top-k scored objects based on descending order of $S_i (i \in \{1, \ldots, N\})$

described below:

$$S_i = \sum_{j=1}^{m} max(0, \delta_{ij}), \tag{20}$$

5.2.3 Interpretablity

As mentioned in previous subsection, one of the key advantages of Random Forest model is it's interpretablity. Which can be achieved in different forms such as feature importance, rule extraction and SHAP values. Each of the above mentioned form are discussed as follows:

1). Interpretablity by Feature Importance:

Random Forests by default provides a measure of feature importance (6), which quantifies the contribution of each feature in the model's predictions. For each feature, the algorithm calculates the total reduction in impurity (e.g., Gini impurity or variance) across all trees when that feature is used for splitting.Features that cause higher reductions in impurity are considered more important.

$$\text{Importance}(f) = \frac{1}{T} \sum_{t=1}^{T} \sum_{s \in S_t(f)} \Delta\text{Impurity}(s),$$

where,

- T is the number of trees,
- $S_t(f)$ is the set of splits in tree t that use feature f,
- $\Delta\text{Impurity}(s)$ is the reduction in impurity achieved by split s.

Advantages:

- Helps identify the most influential features in the dataset.
- Useful for feature selection and understanding the underlying data structure.

2). Interpretablity by Rule Extraction:

Random Forests can be interpreted by extracting decision rules from individual trees in the forest (3). Each tree in the forest can be traversed to extract decision paths (rules) from the root to the leaf nodes. These rules represent the conditions under which the model makes specific predictions.

For example, A rule might look like:

```
IF feature_1 > 5 AND feature_2 <= 3 THEN predict 10.5.
```

Advantages:

- Provides a human-readable explanation of the model's decision-making process.
- Useful for debugging and validating the model's behavior.

3). Interpretablity by SHAP (SHapley Additive exPlanations) Values:

As described in Section 3.5.1, SHAP values provide a methodology for interpreting model predictions by assigning each feature an importance value for a specific prediction (33).

SHAP values are based on cooperative game theory and fairly distribute the "contribution" of each feature to the given prediction. For a given prediction $\hat{y}$, the SHAP value ϕ_f for feature f represents the change in the expected model prediction when f is included versus excluded.

$$\hat{y} = \phi_0 + \sum_{f=1}^{M} \phi_f,$$

where,

- ϕ_0 is the baseline prediction (average prediction over the dataset),
- ϕ_f is the SHAP value for feature f, derived from equation 8,
- M is the total number of features.

Advantages:

- Provides local interpretability (explains individual predictions).

- Consistent and additive, ensuring fair attribution of feature contributions.
- Works very well with tree-based models like Random Forests.

5.2.4 Integrating additional phase: Isolation Forest

Phase 4: Overall Anomaly score calculation using Isolation Forest

To further enhance anomaly detection, a fourth phase is introduced that applies the Isolation Forest (iForest) algorithm (31) on the individual anomaly scores derived from value-wise deviations. Rather than aggregating anomaly scores across all target values (vector-wise deviation), this approach leverages the individual scores (value-wise deviation) directly as inputs to the iForest model. If we refer to Algorithm 5, the only change takes place by integrating this additional phase is in step 11, the for loop to calculate Vector-wise deviation as an anomaly score δ_i would be eliminated, and the calculation of final anomaly score using iForest ($\delta_i \leftarrow iForest(Z, t, \phi)$) as per algorithm 1 will be intoduced instead.

By treating each value-wise deviation as an independent signal of anomaly, the method captures complex patterns that might be missed when aggregating scores. The iForest algorithm efficiently isolates anomalous objects by recursively partitioning the feature space, assigning higher anomaly scores to observations that are more easily isolated. This additional phase enables a more refined and generalized anomaly detection mechanism, improving robustness against noise and irrelevant variations while enhancing the model's ability to distinguish subtle but significant deviations.It was general observation that the evaluation results obtained by using isolation forest as a final step for score aggregation provided much better results as compared to the Vactor-wise deviation. The detailed introduction of Isolation Forest method for anomaly detection can be found in Section 3.3.

5.2.5 Advantages of global approach

The global approach offers several advantages over the local approach:

1). Captures complex, Dataset-Wide dependencies: Identifies deviations that only become apparent when considering large-scale patterns. Catches subtle but impactful trends that local methods might miss

2). Robust to irrelevant local variations: Filters out localized noise by focusing on dominant global dependencies. Prioritizes variables with the strongest influence on target behaviors

3). Provides a unified Anomaly Interpretation: Assigns anomaly scores based on dataset-wide expectations, simplifying interpretation (e.g., "This value violates the global dependency rule"). Avoids fragmentation of models across subgroups.

5.2.6 Disadvantages of global approach

While global dependency-based method models dataset-wide relationships, they face significant drawbacks:

1). Limited flexibility: Less flexible, as it assumes a single global model and consistent relationships between variables for the entire dataset. As a consequence, global approaches may be less effective in datasets with diverse or complex patterns.

2). Bias toward Global Patterns: May Over-rely on training-data patterns reduces generalization, increasing misclassification rates (false positives/negatives) for novel datasets.

3). Inability to capture Local Context: Focuses on overall relationships across the entire dataset, which may miss local patterns or context-specific anomalies. It may fail to detect anomalies that are only apparent in specific contexts or regions of the data.

5.3 Hybrid approach

By having a closer look into the disadvantages of using local and global approaches from the sections 5.1.4 and 5.2.6 respectively, we came to the realization that both approaches tend to complement each other. Local approaches excel at capturing context-specific patterns and anomalies within smaller subsets of the data, making them highly sensitive to local variations and less effective when data is sparse in context. On the other hand, global approaches provide a broader perspective by modeling relationships across the entire dataset, making it efficient in handling cases with sparse context unlike local approach. However, it is unable to capture context-specific anomalies which apparently local approaches are very good at.

To address these limitations, we took the inspiration from the research paper (30), which propose a hybrid approach for contextual anomaly detection namely, **(RObust COntextual anomaly Detection (ROCOD))**. The proposed framework combines the strengths of both local and global methods. By integrating local sensitivity with global context, the hybrid approach can achieve a more comprehensive and robust anomaly detection process. For instance, local methods can be used to identify anomalies within specific contexts or regions of the data, while global methods can validate these anomalies by assessing their consistency with overall data patterns. This collaboration not only

improves detection accuracy but also enhances interpretability, as the hybrid approach can provide both local and global explanations for flagged anomalies.

5.3.1 RObust COntextual Outlier Detection (ROCOD)

As mentioned in Section 5.3, this method combines the strengths of both local and global methods by particularly addressing the problems caused by the contextual sparsity, making it more robust towards broad outlier detection tasks.

Definition: Expected Behavior

For object i with contextual attribute $\mathbf{x}^{(i)}$, its expected behavior $\hat{y}^{(i)}$ is given by $f(\mathbf{x}^{(i)})$, where f models the dependent relationship between contextual and behavioral attributes.

$$\hat{y^{(i)}} = f(x^{(i)}) \tag{21}$$

The approach captures both **local expected behavior** and **global expected behavior** to examine the dependency relationship between behavioral and contextual attributes from distinct perspectives. The local expected behavior is estimated by analyzing the behavioral patterns of objects sharing similar contextual features, known as **contextual neighbors**. In contrast, the global expected behavior **characterizes the overall dependency** between contextual and behavioral attributes across the full dataset, providing a more generalized inference. To effectively merge these two behavioral models, a **regularized integration function** is introduced, which primarily relies on the number of contextual neighbors, supporting cases where contextual information is limited or sparse.

The entire process primarily consists of four phases: 1) Local expected behavior modeling, 2) Global expected behavior modeling, 3) Deriving ensemble expected behavior and 4) Anomaly score generation

1) Local expected behavior modeling

In this phase the dependent pattern from the local aspect is derived in order to obtain the value of Local expected behavior for given object. For that purpose, We first define the set of contextual neighbors for given object.

Contextual neighbor group formation:

As mentioned in Section 4, the **contextual neighbors** or reference group of a specific object comprise those objects that exhibit similarity in their contextual attributes. For-

mally, the contextual neighbors of object i are defined as:

$$CN^{(i)} = \left\{ j : j \in D \wedge j \neq i \wedge \text{sim}(x^{(i)}, x^{(j)}) \geq \alpha \right\} \tag{22}$$

Here, α represents a predefined similarity threshold, and $\text{sim}(\cdot)$ denotes a similarity function applied to the contextual vectors $x^{(i)}$ and $x^{(j)}$. The set $D = \{1, 2, ..., N\}$ contains the indices of all objects. Although various similarity measures can be used for $\text{sim}(\cdot)$, cosine similarity is employed in this study.

Threshold selection:

The accuracy of the local expected behavior values directly depend upon the selection of contextual neighbors which further relies on the selection of threshold α. For that reason, it is very crucial to opt for optimal threshold when forming the set contextual neighbors.

The threshold α determines the **minimum similarity required** for two objects to be considered contextual neighbors. It is chosen based on the distribution of similarity values among a random sample of object pairs. We proceed by computing the pairwise similarity values for a subset of object pairs. Followed by ranking the similarity values in descending order. Finally choosing α as the n^{th}percentile of these values.

$$\alpha = \text{Percentile}_n(\text{sim}(x^{(i)}, x^{(j)})) \tag{23}$$

Where, n is variable which basically manages the trade off between effect of Local and Global expected values in final predictions of given dataset. If n is low (e.g., 5), only a small fraction of object pairs are considered similar, is suitable for datasets with strong contextual relationships. Where as, with higher n (e.g., 80 or 95), a larger fraction of objects qualify as neighbors, useful for datasets with weaker context-behavior correlations which ensures that every object has sufficient neighbors.

Local expected behavior calculation:

The **local expected behavior** for a given object is defined as the average of the **behavioral attributes** of its **contextual neighbors**. This concept is based on the core assumption of **contextual outlier detection**, which implies that objects exhibiting similar **contextual attributes** are likely to display comparable **behavioral patterns**. Formally, the local expected behavior of object i, characterized by contextual attributes $x^{(i)}$, is expressed as:

$$\textbf{LEB} = \hat{y}_{Loc}^{(i)} = \Phi(x^{(i)}) = \frac{\sum\limits_{j \in CN_i} y^{(j)}}{|CN^{(i)}|} \tag{24}$$

where Φ denotes local behavior pattern.

Although the original paper suggested using a simple averaging method to derive Local Expected Behavior, we explored several advanced techniques (e.g., linear regression, XGBoost regression) to achieve more accurate results than mere averaging. A comprehensive overview of the implementation and performance comparison of all models is provided in Section 7.1.4.

The local behavior pattern does not make any prior assumption on the distribution of data. However, since it relies on the presence of contextual neighbors, it becomes ineffective for objects that lack sufficient contextual similarity. In such cases, the local expected behavior cannot be reliably computed for all objects. This limitation demands the need for a more robust and universally applicable approach to estimate expected behavior.

2) Global expected behavior modeling

The **global expected behavior** refers to the global modeling approach used to uncover the dependency structure within the data and estimate the corresponding behavioral attributes. A natural choice for capturing this global relationship between **contextual** and **behavioral attributes** is to use a **regression model**. Specifically, a separate regression model can be trained for each behavioral attribute, where the contextual attributes serve as input features and the behavioral attribute is treated as the response variable. In this framework, the **global expected behavior** of an object corresponds to the predicted values of its behavioral attributes, obtained by applying the learned regression models to its contextual attributes. Formally, the global expected behavior is defined as:

$$\mathbf{GEB} = \hat{y}^{(i)}_{Glob} = \Psi(x^{(i)}) \tag{25}$$

Here, $\Psi(\cdot)$ denotes the mapping function learned through **regression models** trained on the entire dataset, where the **contextual attributes** serve as independent variables and the **behavioral attributes** as dependent variables. To achieve optimal performance, we experimented with a range of **linear and non-linear regression models**.

3) Deriving ensemble expected behavior

Thus far, we have explored two distinct strategies for estimating expected behavior, each offering complementary strengths. The **local expected behavior** approach (as per in the original paper) is **model-free** and exhibits low bias when sufficient contextual neighbors are available. However, it is prone to noise and becomes inapplicable when

no neighbors exist. In contrast, the **global expected behavior** approach captures attribute dependencies across the entire dataset, resulting in reduced variance and improved robustness but might suffer from higher bias by overlooking fine-grained contextual distinctions.

To combine their strengths, this study introduces an **adaptive weighted sum** approach. Unlike fixed-weight schemes, this approach adjusts the weighting dynamically based on the number of contextual neighbors for each object. This allows for improved estimation accuracy and ensures applicability even in cases with no contextual neighbors. The combined **ensemble expected behavior** is defined as:

$$\mathbf{EEB} = \hat{y}_{Ens}^{(i)} = \lambda_i \cdot \Phi(x^{(i)}) + (1 - \lambda_i) \cdot \Psi(x^{(i)}) \tag{26}$$

Where,

$$\lambda_i = \frac{\sqrt{|CN^{(i)}|}}{\max\limits_{1 \leq j \leq N} \sqrt{|CN^{(j)}|}}. \tag{27}$$

Here, $\Phi(x^{(i)})$ and $\Psi(x^{(i)})$ are local expected behavior and global expected behavior as defined in Equations 24 and 25 respectively.

The weighting scheme prioritizes local expected behavior when sufficient contextual neighbors exist ($|CN^{(i)}| > 0$), otherwise defaults to global metrics. A $\sqrt{|CN^{(i)}|}$ transformation improves distribution normality and model robustness. When $|CN^{(i)}| = 0$, local contribution is automatically eliminated.

4) Anomaly score generation

The outlierness of an object is quantified using the L2-norm of the difference between its actual and expected behavior. To accommodate the varying relevance of each target variable, individual weights are assigned based on the coefficient of determination (R^2), which indicates the degree to which the expected behavior aligns with the observed values. As a result, attributes demonstrating higher predictive consistency are given proportionally more weight.

For the j^{th} behavioral attribute, the coefficient of determination is defined as:

$$R^2(y_j, \hat{y}_{Ens}^{(i)}) = 1 - \frac{\sum_{i=1}^{N}(y_j^{(i)} - \hat{y}_{Ens}^{(i)})^2}{\sum_{i=1}^{N}(y_j^{(i)} - \bar{y}_j)^2} \tag{28}$$

where $y_j^{(i)}$ is the value of the j^{th} behavioral attribute for object i, $\hat{y}_j^{(i)}$ is the expected value, and $\bar{y}_j$ is the mean of the j^{th} attribute. The weight of the j^{th} attribute is defined

as:

$$w_j = \max(R^2(y_j, \hat{y}_{Ens}^{(i)}), 0), \quad w_j \in [0, 1] \tag{29}$$

The outlierness score for object i is computed as:

$$S^{(i)} = \left\| W^T (y^{(i)} - \hat{y}_{Ens}^{(i)}) \right\|_2 \tag{30}$$

where $W = (w_1, w_2, \ldots, w_B)^T$ represents the weights of all attributes.

The ROCOD method identifies outliers by selecting the n objects with the highest outlierness scores, where n is a user-defined parameter.

5.3.2 Algorithm: Robust Contextual Outlier Detection

The complete process of contextual outlier detection using hybrid approach is summarized in the Algorithm 6.

Algorithm 6 Robust Contextual Outlier Detection (ROCOD)

Require: Dataset Z, Similarity threshold α, Regression model Ψ
Ensure: Anomaly scores for all objects
Initialize empty anomaly score list S
for each object $z^{(i)} \in Z$ **do**
    Compute pairwise similarity using contextual attributes $x^{(i)}$ and define a distance threshold α
    Define contextual neighbors: $CN^{(i)} = \{j \mid j \neq i, sim(x^{(i)}, x^{(j)} \geq \alpha\}$
    if $|CN^{(i)}| > 0$ **then**
        Compute Local Expected Behavior ($\hat{y}_{Loc}^{(i)}$) :
based on Equation 24
    end if
    Train regression model Ψ using contextual attributes as input
    Compute Global Expected Behavior ($\hat{y}_{Glob}^{(i)}$):
based on Equation 25
    Compute ensemble (weighted) expected behavior ($\hat{y}_{Ens}^{(i)}$):
as mentioned on Equation 26
for that, calculate weights as per Equation 27
    Compute anomaly score:
    Compute the weight w_j of each target variable j by calculating goodness of fit as per Equation 28
    Calculate combine anomaly score $S^{(i)}$ by agreegating deviation with weight for all target variables:

$$S^{(i)} = \|W^T (y^{(i)} - \hat{y}_{Ens}^{(i)})\|_2$$

    Append $(z_i, S^{(i)})$ to anomaly score list S
end for
Rank objects by anomaly scores
Select top k objects with highest scores as anomalies
    return Anomaly scores S

6 Experimental setup

This section details the preparation for conducting the experiments described through out the study, including the details of datasets used and the procedure for generating approximate labels for contextual anomalies. The setup ensures a standardized evaluation of the proposed approaches under controlled conditions.

6.1 Data Description

This study mainly utilizes two databases. First is Domain-specific database (related to our main application) comprising cash withdrawal infrastructure and associated socioeconomic indicators, and the second is database containing benchmark datasets for comparative validation. The former integrates geo-spatial and socioeconomic variables to analyze regional patterns, while the latter provides standardized references for methodological evaluation. Detailed descriptions of data collection, preprocessing, and contextual groupings are elaborated in the following subsections.

6.1.1 Domain-specific database:

This subsection details the integrated datasets central to our analysis:
1) Cash withdrawal resources including ATMs, Bank branches, and Cashback locations across Germany geo-located and aggregated at the regional level. These locations were **manually scraped and validated** from different banking resource locator websites to ensure comprehensive coverage.
2) Socioeconomic indicators spanning socioeconomical factors related to different themes. These datasets enable the examination of spatial and contextual relationships between financial access points and regional socioeconomic conditions.

1). Database containing cash withdrawals locations:

To gather comprehensive location data of cash withdrawal resources across Germany, more than 6 branch locator websites from different bank groups as listed in Table 3 were systematically scraped using Python packages like Requests (43) and Scrapy (13). The implemented function was designed to retrieve ATM, branch, and cashback locations for respective bank groups situated in queried postal code per request. This function was executed for approximately 8,700 postal codes across Germany for each of the listed locator websites. The gathered responses were structured systematically and stored in the appropriate storage format to ensure efficient organization and retrieval.

Retrieved Locations	Website Source
Sparkasse (ATMs + Branches)	https://www.sparkasse.de/standorte
Volksbank (ATMs + Branches)	https://www.vr.de/privatkunden/filialsuche.html
All Cashgroups: (especially) Deutsche Bank (ATMs + Branches), Cashback stores	https://www.deutsche-bank.de/pk/filialsuche.html
All Cashgroups: (especially) Commerzbank (ATMs + Branches), Cashback stores	https://filialsuche.commerzbank.de/
Overall Cashgroups (ATMs + Branches), Cashbacks	https://www.cashgroup.de/
PSD and Volksbanks (ATMs + Branches)	https://www.psd-bank.de/banking-service/geldautomaten/
All Cashpools (ATMs + Branches)	https://www.degussa-bank.de/geldautomaten-suche

Table 3: Retrieved ATM and Branch Locations with Website Sources

Following the data collection, the stored ATM, branch, and cashback location data for each bank group underwent extensive cleaning and processing. This involved normalizing data formats, resolving inconsistencies, and ensuring completeness. To guarantee data authenticity, the processed data was thoroughly cross-validated across multiple sources, including various scraped references from branch locator websites. This rigorous validation step eliminated errors, verify location accuracy, and enhance the reliability of the dataset to ensure that it could be effectively utilized for downstream applications.

After the rigorous cross-validation mentioned above, the final cash withdrawal database was generated, which mainly includes around 18,977 locations of bank branches and 27,408 ATMs locations from 20 different bank groups, as well as supermarkets (POS terminals) of various brands across Germany. Banks are further classified into four networks: Savings Banks, Cooperative Banks, Cash Groups, and CashPools (also known as international banks). The networks are formed based on interbank cooperation, where member banks share infrastructure and resources to provide better services to their customers. Each network consists of various member bank groups. The hierarchy is explained in Table 4.

Bankgroups	Banks	ATM Locations	Total ATM Locations	Branches Locations	Total Branches Locations
Savings bank	Sparkasse	12,036	12,036	7,768	7,768
Co-operative bank	Volks- und Raiffeisenbanken	10,260	10,260	8,102	8,102
Cashgroup	Deutsche Bank	554	2,766	527	2263
	Postbank	1,095		905	
	Commerzbank Total	643		405	
	Hypovereinsbank AG	288		272	
	Oldenburgische Landesbank	50		17	
	PSD Bank	33		33	
	Baden-Württembergische Bank	153		104	
Cashpool	TARGOBANK	331	2,346	331	844
	ING-Geldautomat	945		-	
	Santander Bank	190		157	
	BBBank	102		80	
	Südwestbank	18		33	
	Degussa Bank	119		16	
	National-Bank	30		18	
	Bankhaus Max Flessa KG	28		12	
	Sparda-Banken	527		180	
Total			27,408		18,977

Table 4: Banking Groups: ATMs and Branches Locations count

Additionally, the dataset includes more than 27,000 locations from 30 different brand cashback stores (e.g., Aldi, Lidl, Netto, REWE, Penny, Edeka, etc.), as mentioned in Table 5. This covers the locations of nearly every store that offers cashback services in Germany.

All obtained locations were further mapped within the geolocation boundaries of **400 different administrative regions** of Germany. From this, the counts and densities of cash withdrawal entities per km^2 and per 10,000 residents were derived by combining information about the area and population of each respective administrative region. All of these derived variables (count, density_km^2, and density_10000_residents) for each cash withdrawal entity (bank branches, ATMs, and cashback locations) will be used as target or dependent variables in this study.

Cashback_Store_chain	**Locations_all over_Germany**	**Total_cashback _locations**
ALDI SÜD	1,995	
EDEKA	2,527	
LIDL	3,195	
PENNY Markt	2,090	
REWE	3,629	
Rossmann	2,133	
Norma	1,312	
DM-drogerie Markt	1,732	
Shell Tankstelle	1,174	
Netto Marken-Discount	4,231	
Müller Drogeriemarkt	547	
Kaufland	744	27,316
tegut	314	
toom BauMarkt	280	
E center	161	
nah and gut	103	
SB-Tankstelle	98	
HELLWEG Baumärkte	88	
GLOBUS-Baumarkt/ werhouse	153	
BUDNI	1	
Diska	94	
Active markt	27	
Eckert	27	
NP-markt	222	
Wasgau	73	
Famillia	82	
Tankstelle	4	
SB-Tankstelle	73	
OMV Tankstelle	204	
bft-Tankstelle	3	

Table 5: Cashback Store Chains: location counts in Germany

2). Database containing Socioeconomical factors:

The Socioeconomic Database further contains datasets categorized by multiple themes. Each dataset represents a unique socioeconomic theme, which we refer to as a context. A context consists of a set of different socioeconomic factors that are interrelated or similar in nature.

Context	**Description**	**No. of features**
Demographics	Combines population breakdowns by gender, age, marital status, nationality, and immigration history.	24
Household Economics	Covers Economic Indicators (GDP, GVA, purchasing power), Income distribution and Living Standards (living space per person, rent/m^2) on personal or household level.	29
Sectoral Economics	Covers GDP, GVA, (self)employment, working hours, business registrations, and labor metrics by sectors.	90
Workforce Composition	Workforce Breakdown by Qualifications (academic, professional or no qualification), Nationality (native or immigrant) and Gender.	44
Unemployment	Highlights labor force participation and unemployment across demographics.	25
Vehicle and Crime Statistics	Combines crime rates, car ownership by fuel types, and charging infrastructure.	12
Income Taxpayer and Tax Revenue Metrics	Focuses on counts and amount of taxes in different income ranges, tax revenue, property/trade tax breakdowns and municipal tax metrics.	55
Land Use and Infrastructure	Combines land area, traffic area proportions, and settlement area breakdowns.	22

Table 6: Description and feature counts of contexts

The database consists of more than 300 socioeconomic factors, grouped under 8 distinct contexts/themes (such as taxes, rent and salaries, education, economic sectors, number and types of cars, employment, and others) across 400 German administrative districts. All features were retrieved from open government data sources ((9) and (53)). Most features are well-maintained and complete (no null values) for all 400 regions. In the database, there were negligible amounts of null values, which were easily derivable from other indicators. Additionally, some null values were filled using references from authentic online sources. A detailed description of the contexts and their corresponding features is provided in Table 6. Further list of socioeconomical variables in each dataset are available in the Appendix A.

The aim of this study is to identify regions with anomalous cash withdrawal values relative to the aforementioned contexts. Note that different regions may exhibit anomalies under different contexts. In other words, a region anomalous in one context is not necessarily unusual in another.

6.1.2 Benchmark datasets:

To validate the robustness of our methodological approaches, this study incorporates standardized benchmark datasets with pre-labeled contextual anomalies obtained from the research work of another paper themed on contextual anomaly detection (28). The labels are derived by simply perturbing the random observations from the observations with similar contexts. These datasets serve as a controlled reference for evaluating detection performance across diverse scenarios, complementing the domain-specific analysis of cash withdrawal and socioeconomic data. By checking our results against known benchmark data, we make sure that our methods are reliable and our findings apply to real-world situations.

The datasets span a diverse range of fields, including healthcare and life sciences (e.g., Bodyfat, Heart Failure, Indian Liver Patient, Hepatitis, Parkinson , Abalone, Fish Weight, Toxicity), social sciences (e.g., Boston House Price), environmental protection area (e.g., Forest Fires), and engineering (e.g., Energy, Yacht Hydrodynamics) and so on. We use these datasets as they are representative of the potential application domains of contextual anomaly detection.

Summary of the selected benchmarks and their relevance to our study are outlined in Table 7 where, #Num, #Cat, #Con and #Beh represent the count of numerical features, the count of categorical/nominal features, the count of contextual features, and the count of behavioral features, respectively. All behavioral features are numerical whereas, the contextual features can be mixed. Detailed description of datasets is given in Appendix C.

Dataset	#Num	#Cat	#Samples	#Anomalies (Ratio)	#Con	#Beh
Abalone	8	1	4177	100 (2.4%)	4	5
AirFoil	6	0	1503	70 (4.7%)	5	1
BodyFat	15	0	252	20 (7.9%)	13	2
Boston	12	2	506	40 (7.9%)	13	1
Concrete	9	0	1030	50 (4.8%)	8	1
Energy	10	0	768	50 (6.5%)	8	2
FishWeight	6	1	157	15 (9.5%)	6	1
ForestFires	11	2	517	50 (9.7%)	4	9
GasEmission	11	0	7384	100 (1.4%)	8	3
HeartFailure	1	5	299	30 (10%)	6	5
Hepatitis	11	2	615	30 (4.9%)	3	10
LiverPatient	9	2	579	30 (5.2%)	3	8
Parkinson	21	1	5875	100 (1.7%)	20	2
PowerPlant	5	0	9568	100 (1%)	4	1
Toxicity	7	0	908	50 (5.5%)	6	1
Yacht	7	0	308	30 (9.7%)	6	1

Table 7: Summary banchmark datasets

6.2 Assigning approximate labels for Contextual Anomaly Detection

As the given bank resources dataset is a real-world dataset, one of the most challenging tasks in the analysis was to find the ground truth or reference to compare and validate the results of the algorithms. Even though all approaches mentioned in this study are compared with respect to several benchmark datasets with predefined contextual anomaly labels, it is important to identify how efficiently they perform on real-time datasets. Since we have no prior knowledge about the standards of banking resource allocation concerning socioeconomic factors, or in simpler terms, we don't know the count or density of banking resources that a given region with specific socioeconomic values is supposed to contain, we take multiple regions similar in context to the given region and consider their bank resource counts and densities. This ultimately provides a rough estimation of the count and density of banking resources that the given region is expected to have.

For this purpose, approximate labels are assigned using a very fundamental definition of anomaly stated as follows: "An observation is considered anomalous if it deviates significantly (more than 2 standard deviations) from its respective group's mean." Here, the group refers to its reference group, consisting of observations with similar contextual values, which is constructed using a proximity-based approach such as k-nearest neighbors. For starters, we consider 20 contextual nearest neighbors. We compare the behavioral values of the observation with the mean and standard deviation of the behavioral attributes of the respective reference group, and the label is assigned according to the above-mentioned definition. The procedure for assigning labels is described in Algorithm 7.

One thing to consider is that the labels derived from the method mentioned above should not be regarded as a solid ground truth; instead, they can act as a loose reference for validating the results. Therefore, the evaluation scores derived from comparing the results against these approximate labels should be interpreted with caution, as they provide a preliminary indication of performance rather than an absolute measure of accuracy. To ensure a robust comparison of the performance of different models, we will also utilize pre-labeled benchmarking datasets, which serve as a reliable and standardized basis for evaluation.

Algorithm 7 Contextual Anomaly label assignment

Require: D: Dataset with records (C_i, B_i), C_i: Contextual variables (e.g., socioeconomic factors), B_i: Behavioral variable (e.g., cash withdrawal amounts), k: Number of neighbors for reference group, τ: Threshold multiplier (default $\tau = 2$)
Ensure: Anomaly labels for all records
  for each record $x_i = (C_i, B_i) \in D$ **do**
    Step 1: Find Reference Group
      Compute distances $d(C_i, C_j)\, \forall x_j \in D \setminus \{x_i\}$
      Select $R_i \leftarrow$ top-k neighbors with smallest $d(C_i, C_j)$
    Step 2: Calculate Reference Statistics
      $\mu_i \leftarrow \text{mean}(\{B_j \mid x_j \in R_i\})$
      $\sigma_i \leftarrow \text{std}(\{B_j \mid x_j \in R_i\})$
    Step 3: Assign labels
    if $|B_i - \mu_i| > \tau \cdot \sigma_i$ **then**
      Label x_i as **anomaly**
    else
      Label x_i as **normal**
    end if
  end for

6.3 Environmental setup

All experiments were implemented in Python 3.0 (15), an object-oriented, high-level programming language. We leveraged the key libraries like Pandas for data structures (36), Scikit-learn for statistical modeling (39), Geopandas for handling spatial data (17), Matplotlib (22) and Seaborn (56) for data visualization, SciPy for scientific computing (55), SHAP (34) for interpretation, NumPy for mathematical operations (20).

7 Results and discussion

This section presents key findings derived from the methodologies outlined earlier in this study. This research focuses on detecting contextual anomalies in cash withdrawal transactions while also validating the proposed framework on benchmark datasets. Additionally, the study examines the explainability of the obtained results and explores interpretability aspects in general. The results are presented with a strong focus on transparency and interpretability, supported by comprehensive tables summarizing quantitative findings. The discussion aggregates experimental methods and reveals key insights from anomaly detection.

7.1 Comparison of frameworks

This section compares the performance of the frameworks mentioned in Section 5 with respect to different datasets described in Section 6.1. As discussed before, we have 8 domain (socioeconomical + cash withdrawal) related datasets (Table 6) to work with and 16 additional benchmark datasets (Table 7) for verification purpose. All methodologies are evaluated with respect to the common evaluation measures mentioned in Section 3.4.

7.1.1 Performance of local approach (QCAD)

QCAD parameters setting:

As mentioned in Algorithm 4, QCAD algorithm requires to tune several parameters before proceeding further. First, we need to set up the number of neighbors (k) in order to generate the reference groups. Value selection for this parameter is somewhat trickier as, values too small might be insufficient to capture the proper context on the other hand, very large values of nearest neighbors might add unnecessary noise in the reference group.

Then we need to set the parameters specific to Quantile regression forest such as number and depth of trees, minimum samples required to split a node, and minimal sample size to split a node of trees. These parameters simply decide on complexity of the models. Highly complex model posses risk of overfitting where as very simple model might not capture the trends very well.

Finally, we need to specify the algorithm specific parameters such as the number of conditional quantiles (l) to estimate for an object in each behavioral feature as well as quantiles for upper and lower bounds beyond which we consider the observations as an anomaly.We specify these parameters as follows.

- **The number of nearest neighbors (k)**: The sensitivity check indicates that the approach is comparatively robust with respect to this parameter. By default, we set this parameter to 20 for the benchmarking dataset as the same number of neighbors are considered in providing the labels for the datasets. For the benchmark datasets, we have set it to 10% of dataset sample size.
- **Quantile regression specific parameters**: We experimented with models of different complexities since there is no single model performs optimally across all datasets. Specifically, we tested three distinct configurations: less complex, medium complex and higher complex (a bit more complex than medium). The parameter tuning for each of the model is as follows:

Param	**Purpose**	**Complexity Level**		
		Low	**Medium**	**High**
`n_estimators`	Number of trees in the model	80	100	120
`max_depth`	Maximum depth of each decision tree	7	8	10
`min_samples _split`	Minimum samples required to split a node	15	10	10

Table 8: Hyperparameter values based on model complexity level

- **Algorithm specific parameters**: While more conditional quantiles (n_q) should improve accuracy theoretically, our tests show diminishing returns beyond 100. We therefore set $n_q = 100$ as it provides optimal accuracy-runtime balance. Also as far as the lower and upper bounds of quantiles (Q_{low} and Q_{upper}) concerns, we set them to 5 and 95 quantiles respectively, leaving the margins for extreme values of behavioural attributes in reference group while training the model unlike original paper (28).

Performance evaluation:

After finding the most suitable set of hyperparameters for each dataset we trained QCAD models to detect the contextual anomalies. It was general observation that models with simpler to medium complexity levels are more suitable for most of the datasets except the *Household_Economics* dataset.

Figure 6 compares ROC curves (AUC scores) of the domain (socioeconomic + cash withdrawals) specific datasets. This ROC curve plot compares the effectiveness of various contextual feature sets in detecting anomalies using contextual anomaly detection

methods which is discussed in detail in Section 3.4. As we can see, the QCAD models perform most promising with respect to *'Vehicle and Crime stats'* ($AUC = 0.80$) followed closely by *Household_Economics* ($AUC = 0.78$) and *'Demography'* ($AUC = 0.77$). On the other hand, contextual domains such as *'Unemployment'* ($AUC = 0.58$) and *'Land Use'* ($AUC = 0.60$) are rather slightly better than random classification (dotted line).

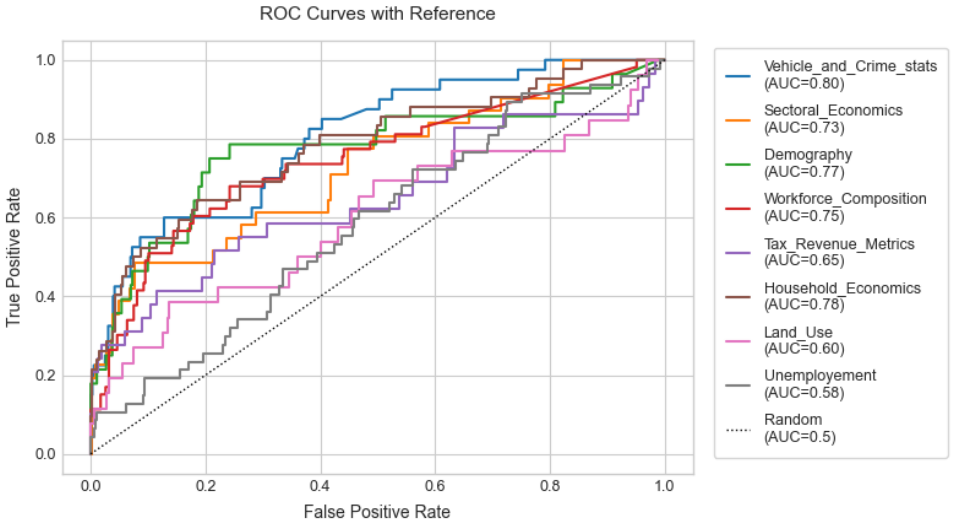


Figure 6: ROC curves for different socioeconomic context datasets

7.1.2 Performance of global approach (DepCAD)

As discussed in Section 5.2, the global approach involves identifying a minimal set of relevant parameters via Markov Blanket. This set is then used to estimate the expected values of the target variables. In this study, we employ a random forest estimator for the expected value computation.

For minimal feature selection, we use fast-IMAB (Fast Incremental Markov Blanket) method (58) to learn the Markov Blankets of variables. It is an efficient algorithm for discovering Markov Blankets in high-dimensional data. It uses incremental updates and heuristic search to quickly identify key variables while minimizing computations. This makes it ideal for real-time analytics and large-scale datasets where traditional methods are too slow. Since it is relatively recent and specialized algorithm, there isn't in built Python libraries for it yet. Therefore, we implemented the same in R (54) using bnlearn package (47) as it provides more statistical accuracy as compare to Python.

For the estimation part, we explored multiple approaches for random forest model training. First we tried Cross-Validated Hyperparameter Tuning (4) which selects optimal values of hyper parameters for model training to ensure the best model performance on unknown data. It accomplishes the same by means of grid or random search along with cross-validation to avoid overfitting and improve predictive stability. Then we tried

bootstrapping (14) approach, which involves creating multiple subsets of the training data through resampling. In this phase, we utilized an ensemble of 20 models, each trained on a different bootstrap sample. This method not only enhanced the robustness of the model but also improved its ability to generalize to new data. Notably, the bootstrapping ensemble approach yielded high Area Under the Curve (AUC) scores for anomaly detection, demonstrating its effectiveness in identifying anomalies with greater accuracy. The final scores obtained by both the methods for different contextual datasets are illustrated in Table 9. As we can see, both PR_AUC and ROC_AUC scores for models trained with Bootstrapping approach are significantly better then the Hyperparameter Tuning one.

Contexts	**DepCAD_Hyperparam _Tuning**		**DepCAD _Bootstraping**	
	PR_AUC	**ROC_AUC**	**PR_AUC**	**ROC_AUC**
Vehicle and Crime Statistics	0.4146	0.8238	**0.4420**	**0.8340**
Sectorial Economics	0.3065	0.7507	**0.3975**	**0.8434**
Demographics	0.3045	0.7513	**0.3316**	**0.8118**
Workforce Composition	0.5586	0.8490	**0.6319**	**0.8769**
Income Taxpayer and Tax Revenue Metrics	0.2969	0.7363	**0.3473**	**0.7631**
Household Economics	0.3073	0.7175	**0.3846**	**0.8033**
Unemployment	0.4571	0.8141	**0.4786**	**0.8603**
Land Use and Infrastructure	0.4546	0.8425	**0.5058**	**0.8848**

Table 9: Performance comparison for DepCAD Models trained with different approaches.

Looking at the results of Table 9, we can conclude that global approach is better at detecting anomalies with respect to *'Workforce Composition'* context. Irrespective of training approach, global anomaly detection framework exhibits exceptional performance when it comes to *'Land Use and Infrastructure'* and *'Unemployment'* unlike the local approach.

7.1.3 Performance after integrating additional Isolation Forest model in global approach (DepCAD + iForest)

As mentioned in Section 5.2.4, we introduced an additional phase of iForest on top of global approach in order to further enhance our results. This process employs a Two-Stage Hybrid Anomaly Detection method, which combines supervised and unsupervised learning. The global approach predicts the behavioral values (cash withdrawal parameters) based on contextual features (socioeconomic factors), helping to identify anomalies in behavioral variables relative to contextual features. By applying the iForest algorithm to the prediction residuals, we can isolate and detect anomalies more accurately. This phase uses individual anomaly scores from value-wise deviations, allowing the model to capture complex patterns that might be missed with aggregated scores.

Overall, integrating very basic iForest model on the final result of the global approach significantly improved the robustness of our anomaly detection mechanism and led to more reliable results. The results of the global approach before and after integration are compared in the Table 10 across the contexts.

Contexts	**DepCAD (Global Approach)**		**DepCAD + iForest (After integration)**	
	PR_AUC	**ROC_AUC**	**PR_AUC**	**ROC_AUC**
Vehicle and Crime Statistics	0.4420	0.8340	**0.89**	**0.9875**
Sectorial Economics	0.3975	0.8434	**0.8881**	**0.9901**
Demographics	0.3316	0.8118	**0.3526**	**0.8185**
Workforce Composition	**0.6319**	**0.8769**	0.6004	0.8743
Income Taxpayer and Tax Revenue Metrics	0.3473	0.7631	**0.4816**	**0.8105**
Household Economics	0.3846	0.8033	**0.464**	**0.8105**
Unemployment	0.4786	0.8603	**0.5431**	**0.8913**
Land Use and Infrastructure	0.5058	0.8848	**0.5648**	**0.9053**

Table 10: Result comparison for Global approaches before and after iForest integration

We can clearly notice the straight up increment in the scores of all contexts after iForest integration, except for the *'Workforce Composition'*. Among all the contexts, scores of *'Vehicle and Crime Statistics'* and *'Sectorial Economics'* increased significantly. The PR_AUC scores for both contexts increased by more than two folds of the scores recorded prior (from 0.442 to 0.8900 and 0.3975 to 0.8881 respectively) to the iForest integration.

Another thing to notice here is the significant increase in PR_AUC scores compared to ROC_AUC when moving from DepCAD to DepCAD + iForest in Table 10. This can be attributed to the fundamental differences in what PR_AUC and ROC_AUC represents, especially in imbalanced classification problems. ROC_AUC measures ranking quality across all classes, where as PR_AUC rewards for detecting the rare positive class well. Integration of iForest makes the model better at that consequently giving the PR_AUC a bigger boost.

7.1.4 Performance of hybrid approach (ROCOD)

As mentioned in Section 5.3, this framework is the combination of local and global approaches mainly targeting the issue of contextual sparsity.

For modeling the values of Local Expected Behavior(LEB), we tried several approaches including simple averaging one mentioned in equation 24 which is also recommended in original paper (30). Apart from simply averaging the behavioral values of observations of contextual neighbors, we also explored slightly sophisticated techniques such as regression (linear or non linear). For linear regression, we used simple linear regression model and we implemented XGBoost as a non linear regressor.

However, one challenge needed to be addressed when implementing regression models for Local Expected Behavior modeling for smaller (but non-zero) Contextual Neighbor groups. In such scenarios, the models tend to overfit the limited available data, leading to inaccurate results. To solve this problem, we treated those cases separately. Instead of applying the complex algorithm, we simply averaged the target variables for observations with considerably smaller (less than 20 neighbors) context group. And for rest of the observations, which has comparatively higher amount of contextual neighbors, we applied the mentioned regression models in order to get more accurate than results than simple averaging.

Comparing the results, modeling with non-linear regression using XGBoost improved overall result significantly than other two (linear regression and averaging) as expected. The model using simple averaging performed reasonably well, while the one employing linear regression resulted in a decline in overall performance.

The original paper recommended using a simple linear regression model for Global Expected Behavior (GEB) values. However, we opted to leverage the expected values predicted by our global approach framework (predicted using set of MB parameters) to enhance both the efficiency and accuracy of the process.

Apart from training the closer to accurate models, the main concern while implementing this framework is to decide upon influence of local and global estimators on the final estimation of values. This trade off could be managed by tuning the similarity threshold. As

mentioned in Section 5.3.1, lower threshold focuses on a strong contextual relationships, while high threshold handles weaker correlations by including more neighbors. Different similarity threshold has been chosen for different contextual datasets depending upon the model performance as described in the Table 11.

Contexts	**PR_AUC**	**ROC_AUC**	**Optimal_Similarity _Threshold**
Vehicle and Crime Statistics	0.5092	0.7815	85
Sectorial Economics	0.3288	0.6324	99
Demographics	0.4062	0.7803	95
Workforce Composition	0.3605	0.7548	30
Income Taxpayer and Tax Revenue Metrics	0.3278	0.6909	5
Household Economics	0.4220	0.7044	40
Unemployment	0.3573	0.7720	50
Land Use and Infrastructure	0.4649	0.7622	60

Table 11: ROCOD results by optimal similarity thresholds

From the table, we can see that contexts like *'Income Taxpayer'*, *'Workforce Composition'*, and *'Household Economics'* use low thresholds, suggesting strong local contextual influence within datasets. In contrast, *'Demographics'*, *'Sectorial Economics'*, and *'Vehicle and Crime Statistics'* rely on high thresholds, implying weaker local context and a need for broader, global similarity to make reliable predictions. Finally *'Unemployment'* and *'Land Use and Infrastructure'* fall somewhere in the middle, showing a balanced blend of local and global behavioral influences. One thing to note here is results described in Table 11 are optimal results derived with non-linear regression as a local expectation prediction model.

7.2 Overall performance comparison

Based on our comprehensive evaluation, which included analyses of various contextual anomaly detection frameworks, our custom Global approach with integrated iForest model consistently outperformed rest of other approaches for most of the contexts, as demonstrated in Table 12.

Contexts	QCAD		DepCAD		DepCAD+ iForest		ROCOD	
	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC
Vehicle and Crime Statistics	0.4582	0.8030	0.4420	0.8340	**0.8901**	**0.9875**	0.5092	0.7815
Sectorial Economics	0.3101	0.7266	0.3975	0.8434	**0.8881**	**0.9901**	0.3288	0.6324
Demographics	0.3765	0.7708	0.3316	0.8118	0.3526	**0.8185**	**0.4062**	0.7803
Workforce Composition	0.4122	0.7641	**0.6319**	**0.8769**	0.6004	0.8743	0.3605	0.7548
Income Taxpayer and Tax Revenue Metrics	0.3148	0.6462	0.3473	0.7631	**0.4816**	**0.8105**	0.3278	0.6909
Household Economics	0.4519	0.7942	0.3846	0.8033	**0.4640**	**0.8105**	0.4220	0.7044
Unemployment	0.2050	0.5837	0.4786	0.8603	**0.5431**	**0.8913**	0.3573	0.7720
Land Use and Infrastructure	0.1981	0.5978	0.5058	0.8848	**0.5648**	**0.9053**	0.4649	0.7622

Table 12: Performance comparison across different frameworks

The comparative analysis indicates significant variations in performance across different frameworks and contexts. DepCAD + iForest turns out to be the strongest overall approach, achieving exceptional results in key domains like *'Vehicle and Crime Statistics'* (PR AUC: 0.8901, ROC AUC: 0.9875) and *'Sectorial Economics'* (PR AUC: 0.8881, ROC AUC: 0.9901). Even for the rest of the contexts, it continues to perform better than other models, suggesting that combining DepCAD's suprvised approach with iForest's robust unsupervised anomaly detection creates a particularly effective hybrid method for these contexts. The framework's consistent outperformance indicates its robustness in handling diverse anomaly patterns.

'Workforce Composition' is an exception where the basic DepCAD (PR AUC: 0.6319) outperforms its enhanced iForest version, this might be due to the presence of strong contextual dependencies, where DepCAD's targeted modeling captures structured relationships more effectively, while the addition of iForest might be introducing unnecessary generalization that slightly degrades performance. Another special case is *'Demography'*, where DepCAD+iForest achieves a higher ROC score of 0.8185, indicating better overall anomaly ranking. In contrast, ROCOD has a superior PR score of 0.4062, which reflects higher precision for identifying clear anomalies. Despite utilizing optimized global predic-

tions (similar to those used in DepCAD) and extensive optimizations related to similarity thresholds, ROCODs continues to lag behind, notably underperforming competing models, especially the global approach.

Comparison of contexts:

Context-specific analysis reveals interesting patterns as well. Across all frameworks, contexts like *'Vehicle and Crime Statistics'* and *'Workforce Composition'* consistently achieve strong performance, likely due to their well-structured relationships and clear anomaly patterns. *'Household Economics'* and *'Sectorial Economics'* (if results from DepCAD + iForest ignored) perform moderately, reflecting useful but less distinct contextual signals. Interestingly, *'Land Use and Infrastructure'* and *'Unemployment'* show superior performance with DepCAD's (and its enhanced version's) global modeling but perform poorly with QCAD's local neighborhood based approach. This contrast in performance is indication that these contexts lack strong localized patterns that neighborhood-based methods can reliably detect. On the other hand, *'Income Taxpayer and Tax Revenue Metrics'* and *'Demography'* consistently perform poorly across all frameworks, likely due to subtle or weak anomaly patterns and noisy context-behavior relationships inefficient for both local and global modeling approaches.

7.3 Comparison of performance on benchmark datasets

As discussed before, to further validate our methodology, we evaluate performance on standardized benchmark datasets with pre-labeled anomalies which provides a controlled baseline to assess detection accuracy across diverse scenarios alongside our domain-specific cash withdrawal analysis. Table 13 compares the performance on different benchmark datasets (described in Table 7) across frameworks.

As can be seen in Table 13, **DepCAD consistently outperforms** other methods, achieving the highest PR_AUC and ROC_AUC on most datasets. It demonstrates strong performance, particularly on large-scale datasets with low anomaly ratios (1-3%) and predominantly continuous features, such as *Abalone*, *GasEmission*, *Parkinson*, and *PowerPlant*. Although **DepCAD+iForest slightly underperforms** DepCAD on some datasets, it still **delivers strong results**. It frequently ranks as the **second-best method** across numerous datasets, achieving the **highest ROC_AUC** (0.9413) and the **second highest PR_AUC** (0.7776) overall scores.

Among other methods, ROCOD performs well on the comparitivly smaller-scale datasets with moderate anomaly ratios (5-10 %) , surpassing DepCAD in PR_AUC for cases like *BodyFat*, *ForestFires*, and *Toxicity*. It also achieves notable ROC_AUC scores on *GasE-*

mission (0.9978) and *HeartFailure* (0.9442). In contrast, QCAD shows weaker overall performance, with generally lower scores across most datasets.

Benchmark Datasets	**QCAD**		**DepCAD**		**DepCAD + iForest**		**ROCOD**	
	PR AUC	**ROC AUC**	**PR AUC**	**ROC AUC**	**PR AUC**	**ROC AUC**	**PR AUC**	**ROC AUC**
Abalone	0.6245	0.8166	**0.9797**	**0.9963**	0.9177	0.9917	0.8354	0.962
AirFoil	0.6440	0.8477	**0.8774**	**0.9785**	0.8591	0.9577	0.5734	0.8599
BodyFat	0.4832	0.8138	0.8100	0.9400	0.7822	0.9218	**0.9021**	**0.9794**
Boston	0.5822	0.7840	**0.7272**	**0.9315**	0.7209	0.9296	0.6173	0.9253
Concrete	0.5706	0.8298	**0.7601**	**0.9226**	0.7445	0.9095	0.6053	0.8398
Energy	0.7878	0.9387	**0.9202**	**0.9701**	0.8420	0.9600	0.8891	0.9774
FishWeight	0.5049	0.7829	**0.8300**	0.9602	0.8161	**0.9669**	0.7980	0.9371
ForestFires	0.3233	0.7787	0.2743	0.7703	0.2641	0.7973	**0.6277**	**0.9376**
GasEmission	0.7927	0.9660	**0.9853**	**0.9902**	0.9609	**0.9989**	0.9587	0.9978
HeartFailure	0.4256	0.8455	**0.7348**	0.9276	0.6001	0.9238	0.6941	**0.9442**
Hepatitis	0.6678	0.8703	**0.9398**	**0.9978**	0.8635	0.9878	0.8115	0.9767
LiverPatient	0.5222	0.9081	**0.9011**	0.9666	0.7585	**0.9771**	0.6644	0.9509
Parkinson	0.7668	0.8922	**0.9870**	**0.9979**	0.9741	0.9966	0.5062	0.7629
PowerPlant	0.6185	0.9158	**0.9155**	0.9600	0.9067	**0.9887**	0.8134	0.9594
Toxicity	0.3559	0.7580	0.6517	0.8928	0.6288	0.8675	**0.7091**	**0.9092**
Yacht	0.7616	0.8921	**0.9948**	**0.9956**	0.9491	0.9948	0.5883	0.7885
Average_Scores	0.5873	0.8525	**0.8214**	**0.9398**	**0.7776**	**0.9413**	0.7170	0.9071

Table 13: Performance comparison across benchmark datasets

The feature composition is also an important factor influencing performance of each dataset. For instance, datasets containing a higher number of **continuous features** (such as *Abalone*, *GasEmission*, *Parkinson*,etc.), see strong results with DepCAD. Meanwhile, datasets like *HeartFailure* and *Hepatitis*, which contain a larger proportion of **categorical features**, show competitive performance from ROCOD, which seems to adapt well at handling mixed-type attributes. Overall, **DepCAD and DepCAD+iforest maintain stable performance**, likely because they can learn complex relationships better.

8 Addressing research questions from obtained results

While the previous section discusses the results from the perspective of framework performance metrics, this section analyzes the disparity in the distribution of cash withdrawals in Germany and relates the results to theoretical knowledge. Additionally, we will be addressing all the research questions mentioned in Section 1.1 in this section.

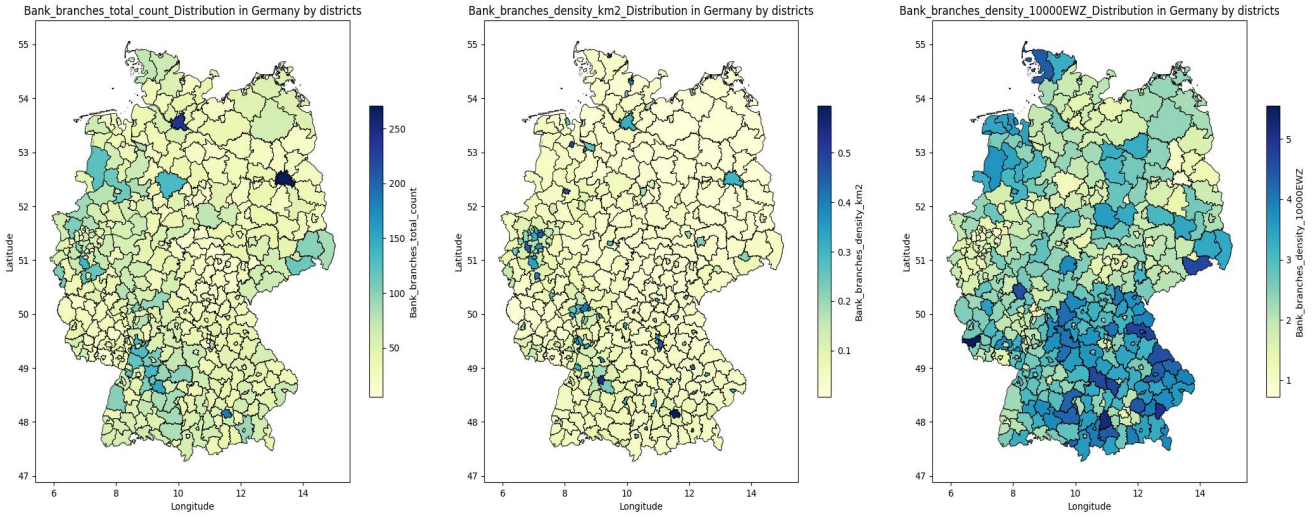
As mentioned in Section 6.1.1, the study mapped all cash withdrawal locations across Germany's 400 administrative regions, calculating both absolute counts and two density metrics (per km^2 and per 10,000 residents) for each service type (Banks branches, ATMs, cashbacks and ATMs+Cashbacks). In total, we have **12 target variables** to analyze.

The density measures complement raw counts by normalizing for geographic size and population, revealing disparities that simple counts might overlook such as, urban shortages despite high counts or rural adequacy despite low numbers. Together, these three variables provide a **complete spatial, geographic and demographic assessment** of service distribution. Additionally, since ATMs and Cashbacks both provide ways to access cash, this study covers all the options available for cash withdrawals by **jointly modeling** both entities together. Analyzing them together helps to avoid mistakenly thinking that there are big service gaps in places where cashback points make up for the lack of ATMs.

8.1 Analyzing actual distributions:

Considering the distribution of bank branches from Figure 7 (generated using Geopandas (17) and Matplotlib (22) packages), we can clearly notice that the major cities like Berlin, Hamburg, and Munich stand out with high bank branch counts (darkest shades of blue in Figure 7(a)) as well as density_km^2 (Figure 7(b)), but lag in population-adjusted availability (Figure 7(c))due to high population concentrations. In contrast, Southern rural areas offer strong per-head coverage, suggesting better access per resident, even with moderate infrastructure. Eastern regions display a mixed pattern, often appearing underserved by count or area but showing adequate service relative to population in some districts.

Figure 8 compares the joint distribution of ATMs and Cashback stores all across Germany. The count and density distributions show a similar pattern to that of bank branches (with major cities dominating access); however, when looking at the combined data for ATMs and cashbacks, the count heatmap appears to be more evenly spread across the country. However, when adjusted for the population (Figure 11(c)), many rural and southern regions, particularly in Bavaria and parts of the east, demonstrate high per capita access, highlighting a more equitable spread.

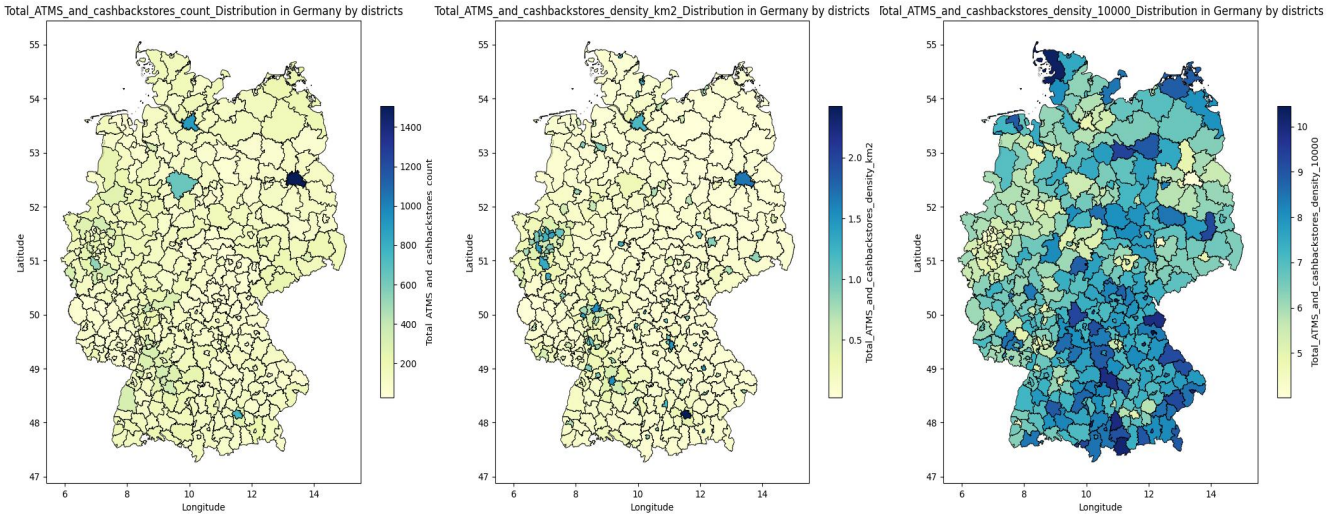


(a) Bank branches total count

(b) Bank branches density (km^2)

(c) Bank branches density (per 10,000 head)

Figure 7: Bank branches distribution across Germany



(a) ATMs and Cashbackstores total count

(b) ATMs and Cashbackstores density (km^2)

(c) ATMs and Cashbackstores density(per 10,000 head)

Figure 8: ATMs and Cashbackstores joint distribution across Germany

Additionally, the separate distribution of ATMs and cashbacks shows similar behavior with respect to count and density (by area) distribution when analyzed individually, re-

inforcing the overall trend observed when they are modeled together. However, when considering density distribution by population, ATMs are more concentrated in southern regions of Germany, while cashbacks are denser in the eastern states. Reference distribution heatmaps for both ATMs and cashbacks can be found in Appendix B.

8.2 Identifying and analyzing the blind spots:

We combined the final results from all five approaches, including the one discussed in Section 6.2, to detect contextual anomalies across all eight contexts. In total, we analyzed the results of 40 models (5 models for each of the 8 contexts) to generate a comprehensive list of cities identified as anomalous.

City	**Anomalous Context (out of 8)**	**Anomalous (as per total models)**	**Anomalous (% of models)**	**Anomalous w.r.t. context except for**
Berlin	8	39	97.5	none (Anomalous w.r.t. all contexts)
Hamburg	8	39	97.5	none (Anomalous w.r.t. all contexts)
München	8	39	97.5	none (Anomalous w.r.t. all contexts)
Region Hannover	8	37	92.5	none (Anomalous w.r.t. all contexts)
Rosenheim	8	35	87.5	none (Anomalous w.r.t. all contexts)
Esslingen	8	33	82.5	none (Anomalous w.r.t. all contexts)
Stuttgart	8	31	77.5	none (Anomalous w.r.t. all contexts)
Köln	7	25	62.5	Vehicle and Crime stats
Ludwigsburg	7	25	62.5	Workforce Composition
Kreis Rhein-Sieg-Kreis	6	21	52.5	Land Use and Infrastructure, Unemployment
Landkreis Ortenaukreis	6	21	52.5	Demography, Land Use and Infrastructure
Nürnberg	5	19	47.5	Demography, Sectorial Economics, Land Use and Infrastructure
Dortmund	5	17	42.5	Demography, Sectorial Economics, Land Use and Infrastructure
Landsberg am Lech	5	16	40	Household Economics, Income Taxpayer and Tax Revenue Metrics, Unemployment
Nordfriesland	5	16	40	Sectorial Economics, Income Taxpayer and Tax Revenue Metrics, Land Use

Table 14: List of anomalous cities across contextual domains

The Table 14 presents a list of top 15 cities ranked based on the number of models identified them as anomaly across eight contextual domains, highlighting the number of contexts in which each city appears as an outlier, the total number and proportion of models (out of 40) that flagged the respective city as anomaly. In this analysis, we

highlight only the contexts where cities are **not** anomalous, as most cities show anomalies across nearly all eight domains. This approach avoids repetition and offers a clearer, more concise summary.

As we can see, the list is a well-balanced combination of bigger cities, mid-tier cities and smaller districts ranging from being identified as anomaly by almost 100% of models with respect to all 8 contexts (more than 50% cities from list) to be anomalous as per 40% of models with anomalies in only a subset of contexts. Interestingly, all models show strong agreement in detecting the same set of anomalous cities, regardless of their individual performance or sensitivity. While major cities like Berlin, Hamburg, and München are expected to stand out due to their scale and complexity, the consistent flagging of less prominent anomalies such as **Region Hannover, Rosenheim, and Esslingen** provides particularly valuable insights. Though not Germany's largest urban centers, repeatedly appear as outliers by **more than 80%** of models across all contextual domains, pointing to underlying structural or socioeconomic irregularities that might otherwise remain hidden.

These deviations could be traced to distinct local dynamics for instance, Hannover's role as a logistics and trade center (Hannover Messe) is likely to inflate cash demand beyond what resident population metrics would suggest. Meanwhile Esslingen's excessive cashwithdrawals might be driven by its integration with Stuttgart's automotive industry creating high daytime worker populations. In contrast, Rosenheim's outlier status possibly linked to a strong agricultural sector, which shows in its higher Gross Value Added (GVA) in primary sector, indicating a rural economy that relies heavily on cash.

8.3 Interpretablity:

As discussed in Section 1.1, apart from anomaly detection, achieving interpretablity is equally important objective of this study. In order to achieve the same we have used several methods described in the Section 5.2.3.

8.3.1 Interpretablity by Feature Importance

As mentioned in the Section 5.2.3, Random Forests from global approach by default provides a measure of feature importance, which quantifies the contribution of each feature in the model's predictions. The summary of overall feature importance analysis is as follow,

Cash withdrawal resources counts:
Population is the top driver across all resources when it comes to count prediction, with the highest importance for cashback stores (0.96) and significant weight for branches

(0.73) and ATMs (0.85). Cashback stores thrive in commercial areas, with strong ties to registered businesses (0.75) and female professionals (0.63). Where as, Branches depend on economic health (GDP/GVA) and land availability (0.35), serving diverse employment needs. Finally, ATMs counts are linked to mobility (car ownership(0.48)), and accessibility (disability rates at 0.58).

Cash withdrawal resources density by area:
The proportion of road traffic area is a consistently strong predictor of density (with importance scores of 0.74 for branches, 0.79 for cashback, and 0.81 for ATMs), highlighting the critical role of infrastructure and accessibility. The urban status (Kreisfreiestadt_flag) also appears frequently, especially for cashback services where it shows up multiple times with high importance (e.g., 0.62, 0.59, 0.53), indicating a strong urban concentration of these services.

Branch density is strongly associated with areas generating high Property Tax A revenue (0.93), indicating a concentration in agriculturally active rural regions. Cashback density, on the other hand, shows a clear urban focus, heavily influenced by city status. ATMs density reflects a more balanced pattern, influenced by both rural economic indicators (e.g., Property Tax A: 0.88) and residential factors like marital status (0.41) and living space (0.42).

Cash withdrawal resources density by population:
Branch density by population is most influenced by recreational land use (importance: 0.82) and unemployment rates (0.61), indicating a connection between community infrastructure and banking presence in less economically vibrant areas. Cashback store density appears driven by workforce characteristics particularly, foreign unemployed individuals (0.53) and qualified female employees (0.52), suggesting urban and socially diverse areas are key locations. ATM density by population is shaped strongly by burglary rates (0.56) and unemployment (0.55), pointing toward a reactive placement strategy in areas where both cash demand and perceived security concerns may be higher.

8.3.2 Interpretablity by SHAP

As described in Section 3.5.1, the Shapley value indicates the marginal contribution of a given feature by considering all possible coalitions or combinations of features in the model. In this section, we will be analyzing the contribution of different features in anomaly scores of different cities with respect to our optimal DepCAD + iForest model. Here we have applied the SHAP calculations in both the phase, during global resource value prediction and while applying the iForest model on prediction errors. In order to interprete the same, we are using bottom up approch. We first analyse the SHAP calculation for iForest phase which reveals the contribution of individual Cash withdrawal

resource entitities behind the anomaly score of the given city followed by, the contribution of Socioeconomic features for prediction error of the Cash withdrawal resources by SHAP calculations of prediction phase. The described methodology is demonstrated in Figure 9. The SHAP values and corresponding contribution waterfall graphs were generated using the SHAP package (34) in Python.

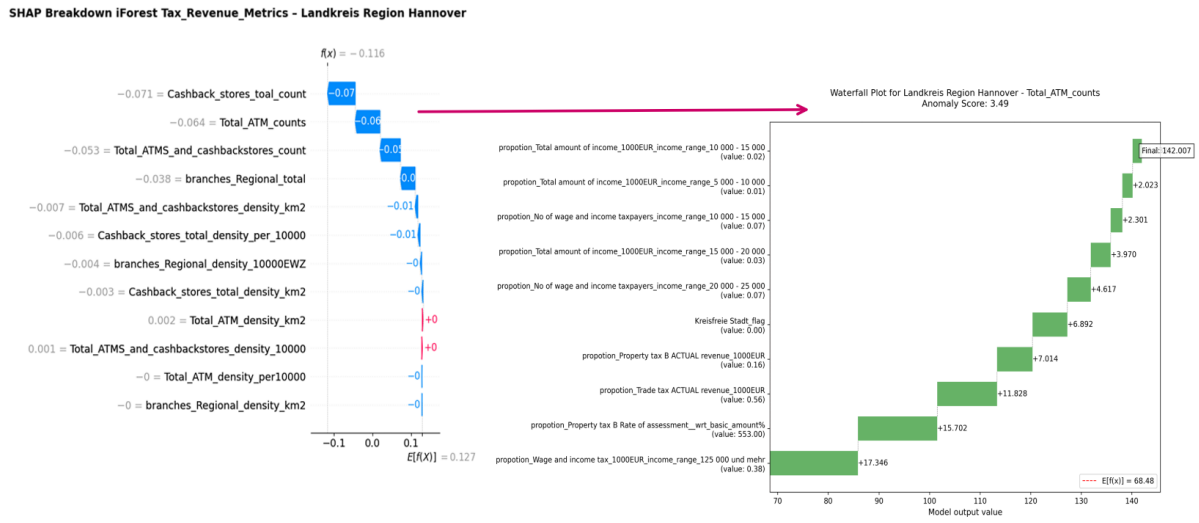


Figure 9: Interpretation of SHAP values (iForest on left and Random Forest prediction on right) of Tax Revenue matrix for Hannover

The left hand side of Figure 9 illustrates the SHAP breakdown of different Cash withdrawal resources in iForest applied on the prediction errors with respect to *Tax revenue matrix* context. As we can see from the graph the value of Anomaly Score $f(x) = -0.116$ which is significantly lesser than the Base Anomaly Score $E[f(x)] = 0.127$. It's important to note that, according to our model, a lower Anomaly Score indicates that the observation is more anomalous. The figure clearly shows that variables such as *'Cashback_stores_total_count'* and *'Total_ATM_counts'* are among the most significant contributors to the anomaly score (-0.071 and -0.064 respectively) for the Region Hannover with respect to *Tax revenue matrix*.

Now, we will be analyzing the SHAP contributions of the contextual features from *Tax revenue matrix* in prediction of *'Total_ATM_counts'* on the right hand side of the Figure 9. The model is predicting around 142 ATMs in Hannover Region based on it's *Tax revenue matrix* characteristics values which is much higher than base level prediction value ($E[f(x)] = 68.48$),indicating significantly above average ATMs. We can clearly notice that, Income tax contribution of significant proportion of taxpayers with more than 125,000€ annual income contributes upto 17 additional ATMS (on top of baseline) in the city. The second major contributor is the rate of assessment for Property Tax B (levied houses, apartments, and commercial buildings), which adds around 16 ATMS to the pre-

diction. The third key driver is the actual revenue from trade tax, with a contribution of around 12 additional ATMs. Together, these factors push the model's output from a baseline of 68.48 to 142.01, resulting in an anomalous level of ATM availability in the region.

8.4 Ideal Distribution Projection:

Now, the obvious question that arises is: Given the evidence of blind spots in cash withdrawal distribution, how should the ideal distribution look like as per current socioeconomic indicators? In order to address this question, we derived the expected values of individual Cash withdrawal entity variable by calculating the weighted average of their predictions with respect to all 8 contexts. The weights are assigned based on model's ability to accurately capture and explain the observed data (also known as Goodness of fit).

Goodness of Fit or Coefficient of Determination (R^2)
Evaluates how well a statistical model replicates observed data by quantifying the proportion of variance in the dependent variable explained by the model:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{31}$$

where, y_i represents Observed value, $\hat{y}_i$ is Predicted value and $\bar{y}$ is Mean of observed data.

The weights for the individual contexts are assigned in such a way that their sum equals 1 for each given behavioral variable. The final results, which are derived based on weighted average of predicted values with respect to all contexts are then expected value ($\hat{y_{exp^{(i)}}}$) of given behavioral (Cash withdrawal) attribute for the given region. We then calculate the projections by considering the deviation between actual ($y^{(i)}$) and expected values ($\hat{y_{exp^{(i)}}}$) of respective cashwithdrawal quantity followed by scaling the deviation by expected values to project the Expectation vs. Reality gap in percentage.

for i^{th} region and j^{th} context for given Cash withdrawal variable the expected value, $\hat{y_{exp^{(i)}}}$ (the ideal value of cashwithrawal entity as per overall socioeconomical indicators in i^{th} region) is defined as follows,

$$\hat{y_{exp^{(i)}}} = \sum_{j=1}^{8}(w_j * \hat{y}_{ij}) \qquad \text{Where, } \sum_{j=1}^{8} w_j = 1 \tag{32}$$

Here, ($\hat{y}_{ij}$) represents the predicted cash withdrawal value for the (i^{th}) region based on the socioeconomic indicators of the (j^{th}) context using the global model. Finally the projection ($\hat{y}_{\text{proj}^{(i)}}$) of the same is defined as,

$$\hat{y}_{\text{proj}^{(i)}} = \frac{y^{(i)} - \hat{y_{exp^{(i)}}}}{\hat{y_{exp^{(i)}}}} \times 100 \tag{33}$$

Table 15 demonstrates the goodness of fit scores for the Random Forest models (from global approach) trained for different Cash withdrawal variables across all 8 contexts, with the highest score for each variable emphasized in bold font. Overall we can see that model is fitting very well to majority of variables with respect to most of the contexts, especially *'Sectorial Economics'* is exceptionally good at describing the *Total_ATM_count* distribution (0.9395) where as *'Demography'* is fitting very well on *Branches_total_count* and *Cashbackstores_total_count* with R^2 score of 0.8733 and 0.9708 respectively.

(R^2 Scores)	Vehicle and Crime Stats	Demography	Workforce Composition	Sectoral Economics	Household Economics	Tax Revenue Metrics	Unemployment	Land Use
Branches _total_count	0.5699	**0.8733**	0.5400	0.8776	0.8449	0.5847	0.6913	0.6693
Branches _density_km2	0.8532	0.8814	0.7625	0.8573	0.8594	0.8684	0.6637	**0.9336**
Branches _density _10000EWZ	0.6576	0.6581	0.6745	**0.6966**	0.7147	0.6756	0.6091	0.6577
Cashbackstores _total_count	0.7204	**0.9708**	0.7568	0.9532	0.6574	0.5705	0.8600	0.7591
Cashbackstores _density_km2	0.9122	0.9143	0.7769	0.9200	0.9098	0.8828	0.6749	**0.9700**
Cashbackstores _density _per_10000EWZ	0.5687	**0.6177**	0.5527	0.5435	0.5282	0.5712	0.5071	0.5269
Total_ATM _counts	0.6533	0.9338	0.6115	**0.9395**	0.6658	0.6564	0.7646	0.7393
Total_ATM _density_km2	0.9065	0.9060	0.7803	0.9060	0.9100	**0.9207**	0.6778	0.9478
Total_ATM _density _per_10000EWZ	0.6651	0.6665	0.6528	**0.7266**	0.6813	0.6688	0.6375	0.6675
Total_ATMS and _cashbackstores_count	0.7131	**0.9670**	0.6690	0.9558	0.6351	0.5484	0.8360	0.7781
Total_ATMS and _cashbackstores _density_km2	0.9208	0.9235	0.7624	0.9189	0.9078	0.8770	0.6842	**0.9669**
Total_ATMS and_cashback stores_density _10000EWZ	0.6957	0.6955	0.6852	**0.7013**	0.6821	0.6508	0.6482	0.6070

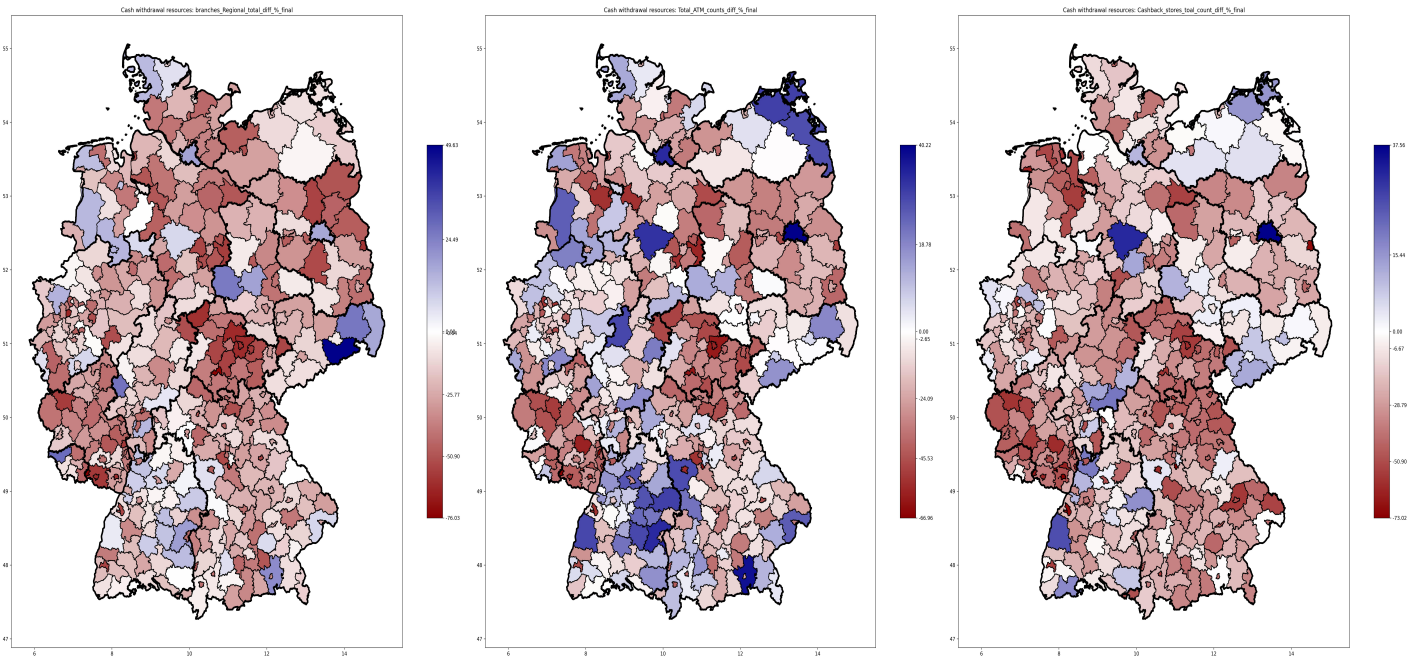
Table 15: Goodness of fit (R^2) scores for global approach across contextual domains

8.4.1 Projection analysis:

Based on the goodness of fit values in above given table, weights and expected values has been derived as per Equations 32 which further serves as basis of projection calculation as per Equation 33. Figure 10 represents side by side comparisons of the derived projections for all Cash withdrawal count distribution separated by the states (borders highlighted

in bold) with their respective scales on right side. Red tone on the scale indicate lower than expected values, blue tones indicate higher actual counts than expected values, and white regions represent actual counts similar to expected values, reflecting a close to 0% difference. The scale is imbalanced, with the minimum percentage (indicating the highest shortage) marked at the lower extreme and the maximum percentage (indicating the highest over-served regions) marked at the upper extreme.

In the Figure 10(a) depicting bank branches projections across the regions spanning from -76% to +50%, a general trend of under-supply is evident in central and southern Germany.All regional and state-level mappings are provided in the Appendix B for reference. States like Thüringen, Brandenburg and Rheinland-Pfalz are prominently red, indicating that these areas have significantly fewer branches than expected. In contrast, Saxony (Sachsen) especially, the area around Dresden shows a dark blue zone (approching +50%), pointing to a branch count much higher than expected. Frequent blue clusters also appear in Baden-Wüttemburg and parts of Niedersachsen, hinting at localized over service.



(a) Bank branches count (b) ATM counts (c) Cashback stores count

Figure 10: Projections: Cash withdrawals resources counts with highlighted state borders (generated using Geopandas (17) and Matplotlib (22) packages)

The ATM distribution shown in the second Figure 10(b) with the range extending from -67% to +40% across the regions, illustrates a more mixed picture. Baden-Wüttemburg and Mecklenburg-Vorpommern display a notable surplus of ATMs. Apart from the bigger cities (Berlin, Hamburg, Hannover,etc.), rural regions like Waldereck-Frankenberg (in Hessen), Emsland and Steinfurt (in Niedersacshen) show higher-than-expected ATMs.

Similar to bank branches, states like Thüringen, Brandenburg and Rheinland-Pfalz are prominently red, indicating significant shortage of ATMs than expected.

Cashback resources in Figure 10(c) appears somewhat more balanced between red and white areas ranging from -73% to +38%, with fewer dark blue hotspots compared to the ATM map, suggesting cashback services are more evenly distributed relative to expectations in some regions. Regions in eastern Germany, excluding Brandenburg and Thüringen, show a fairly balanced distribution with values that align closely with expectations, particularly in Mecklenburg-Vorpommern and Saxony. Where as western and southern areas (Rheinland-Pfalz and Bayern) consistently show moderate to severe underprovision of cashback services.

Overall, German banking infrastructure shows persistent regional disparities, with some states such as **Thüringen, Brandenburg, Rheinland-Pfalz** experiencing significant **under-provision across all cash access points**, while eastern regions around Dresden and parts of Baden-Württemberg display isolated service surpluses. Out of all, **Mecklenburg-Vorpommern** maintains relatively **balanced access**, reflecting systematic regional patterns in cash withdrawal distribution.

9 Conclusion and Future Scope

Contextual anomaly detection is a type of anomaly detection that identifies data points that are considered anomalous within specific contexts, while appearing normal in general. This thesis presents a comprehensive study on **contextual anomaly detection (CAD)** applied to the distribution of cash withdrawal resources in Germany, including bank branches, ATMs, and cashback stores. Here, we are considering the set of different **socioeconomic indicators** as contexts. The motivation originates from the need to understand and address potential inequities in financial service accessibility, with a focus on how resource distribution correlates with socioeconomic indicators.

For that purpose, three major methodological frameworks were proposed and evaluated. The **local QCAD** approach leverages **Quantile Regression Forests (QRF)** on contextually similar instances (via Gower's distance), effectively capturing nuanced, context-specific anomalies but struggling in sparse data scenarios. In contrast, the **global DepCAD** approach employs **Markov Blanket** feature selection with **Random Forest** regression though it may overlook localized patterns. Bridging these extremes, the **hybrid ROCOD** framework dynamically combines local and global expected behaviors, adapting to data density variations to deliver robust performance where individual methods lake, thus offering a balanced solution for diverse anomaly detection scenarios. Additionally, we proposed to integrate an **Isolation Forest with the global approach (DepCAD+iForest)** for refined anomaly scoring.

The models were rigorously evaluated using both real-world (Cash withdrawals locations mapped across **400 regions** w.r.t. 8 Socioeconomic contexts) and benchmark datasets. The performances were evaluated using different metrics (PR-AUC and ROC-AUC). The detailed assessment concludes, **DepCAD+iForest (method proposed by us)** as the **superior framework**, delivering top performance in **approximately 75% of cases across domain-specific** and second best (or very close to the best) in case of **around 70% benchmark datasets**. While **DepCAD** alone achieves excellent results with structured numerical information (espacially in case of benchmark datasets), combining it with iForest yields a better performance when dealing with sophisticated detection tasks or infrequent anomalies. ROCOD shows limited but valuable effectiveness for most of the datasets, whereas **QCAD consistently underperforms**. These results clearly demonstrate that the global approach (DepCAD) and its enhanced iForest version (DepCAD+iForest) offer the most robust solution (most likely due to advantage of feature selection above all other approaches)for diverse anomaly detection tasks.

The study also translates obtained analytical results (from framework implementations) into the real-world implications of cash withdrawal resource accessibility in Germany. The goal was to systematically answer the four core research questions posed at the beginning

of the study. Initially, the actual spatial distribution of Bank branches, ATMs, and Cashback stores, both in terms of actual count and with respect to basic indicators such as area and population, was analyzed using heatmaps. The analysis reveals disparities in the distribution of cash withdrawal resources, with **major cities** having **high counts and density by area but lower rates of density by population**. Additionally, **rural regions** (especially southern rural areas) showing **exact opposite** pattern than their urban counterparts.

Next, regions with **service gaps (blind spots)** in cash withdrawal access were identified by aggregating results from all five anomaly detection methods across eight contextual scenarios. This comprehensive analysis generated **up to 40 (5 methods $\times$ 8 contexts)** distinct model outputs per region, ensuring multi-perspective identification of access disparities. The analysis revealed while major cities (Berlin, Hamburg, Munich) exhibited significant disparities, mid-sized regions like **Hannover, Rosenheim, and Esslingen** offered unexpected insights into cash access patterns **beyond metropolitan centers**.

Model **interpretability** is achieved through SHAP-based feature attribution, decomposing anomaly scores into contributions from cash service availability and socioeconomic variables. **Population metrics, economic indices, and land use** emerge as **dominant** factors, explaining regional variations in cash access adequacy. Finally, ideal distribution projection of cash withdrawal resources were modeled by calculating expected values through socioeconomic-weighted prediction averages. The projection highlights **structural imbalances**, particularly acute **shortfalls** across the states like **Thüringen, Brandenburg and Rheinland-Pfalz** against theoretically optimal service levels.

While This study establishes a robust framework for contextual anomaly detection in cash withdrawal infrastructure, there are several promising directions for future research. First, the validity of the identified anomalies and distribution patterns could be **further confirmed** by the **updated** financial resource dataset, ensuring that the findings remain relevant to changing real-world conditions. Second, the methodology could be expanded to a more **detailed regional hierarchy**, specifically focusing on the **municipal level** (approximately **10,753 administrative units** in Germany). This would enable more localized insights, particularly in identifying micro-level blind spots that may be overlooked at higher administrative levels. Finally, the effectiveness of local anomaly detection approaches could be improved by integrating more **advanced models**, such as **Contextual Isolation Forests** (35) or **neighbor-proximity-based model training** techniques(27). These models could better capture subtle contextual dependencies and might enhance robustness.

Bibliography

[1] Charu C Aggarwal. Outlier analysis. *Data Mining*, pages 237–263, 2015.

[2] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley & Sons, 1994.

[3] Clément Bénard, Gérard Biau, Sébastien Da Veiga, and Erwan Scornet. Interpretable random forests via rule extraction. *arXiv preprint arXiv:2004.14841*, 2021.

[4] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[5] Gérard Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13(1):1063–1095, 2012.

[6] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[7] Leo Breiman, Jerome H Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[8] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 93–104. ACM, 2000.

[9] Bundesinstitut für Bau-, Stadt- und Raumforschung (BBSR). Inkar – indikatoren und karten zur raum- und stadtentwicklung, 2024. Interactive online atlas of socioeconomic indicators for Germany and Europe.

[10] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 51(3):1–36, 2019.

[11] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):1–58, 2009.

[12] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240, 2006.

[13] Scrapy Developers. Scrapy: A fast and powerful scraping and web crawling framework, 2008. Accessed: 2023-10-01.

[14] Bradley Efron and Robert J. Tibshirani. An introduction to the bootstrap. *Monographs on Statistics and Applied Probability*, 57, 1994.

[15] Python Software Foundation. Python language reference, 2023.

[16] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Quantile regression for dynamic treatment regimes. *Journal of the American Statistical Association*, 115(531):1230–1240, 2020.

[17] GeoPandas Development Team. Geopandas: Python tools for geographic data.

[18] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871, 1971.

[19] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2011.

[20] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with numpy. *Nature*, 585:357–362, 2020.

[21] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, 2003.

[22] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[23] Boris Iglewicz and David C. Hoaglin. *How to Detect and Handle Outliers*. SAGE Publications, 1993.

[24] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[25] George Karypis, Eui-Hong Han, and Vipin Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer Society*, 32(8):68–75, 1999.

[26] Kasper Knudsen and Vilhelm Söderström. Interpretable outlier detection in financial data: Implementation of isolation forest and model-specific feature importance, 2023. Accessed: 2024-03-15.

[27] Mingshu Li, Bhaskarjit Sarmah, Dhruv Desai, Joshua Rosaler, Snigdha Bhagat, Philip Sommer, and Dhagash Mehta. Quantile regression using random forest proximities, 2024.

[28] Zhong Li and Matthijs van Leeuwen. Explainable contextual anomaly detection using quantile regression forests. 2023.

[29] Zhong Li, Yuxuan Zhu, and Matthijs van Leeuwen. A survey on explainable anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, 18(1):23:1–23:54, 2024.

[30] Jiongqian Liang and Srinivasan Parthasarathy. Robust contextual outlier detection: Where context meets sparsity. 2016.

[31] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.

[32] Sha Lu, Lin Liu, Jiuyong Li, Thuc Duy Le, and Jixue Liu. Lopad: A local prediction approach to anomaly detection. 2020.

[33] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

[34] Scott M Lundberg and Su-In Lee. Shap (shapley additive explanations). `https://github.com/shap/shap`, 2021. Accessed: 2025-04-29.

[35] Meghanath Macha, Leman Akoglu, and Christos Faloutsos. ConOut: Contextual outlier detection with multiple contexts: Application to ad fraud. In *Advances in Knowledge Discovery and Data Mining - 23rd Pacific-Asia Conference, PAKDD 2019*, volume 11439 of *Lecture Notes in Computer Science*, pages 136–151. Springer, 2019.

[36] Wes McKinney. Data structures for statistical computing in python. pages 56–61, 2010.

[37] Geoffrey McLachlan and David Peel. Finite mixture models. *Wiley Series in Probability and Statistics*, 2000.

[38] Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(Jun):983–999, 2006.

[39] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[40] Tomáš Pevný. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, 2016.

[41] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[42] Sunil Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 427–438. ACM, 2000.

[43] Kenneth Reitz. Requests: Http for humans, 2011. Accessed: 2023-10-01.

[44] Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

[45] Masashi Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014.

[46] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *Information Processing in Medical Imaging (IPMI)*, pages 146–157, 2017.

[47] Marco Scutari. bnlearn: Bayesian network structure learning. *Journal of Statistical Software*, 35(3):1–22, 2010.

[48] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[49] John Smith. Understanding the curse of dimensionality. *Journal of Data Science*, 15(2):123–145, 2023.

[50] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing Management*, 45(4):427–437, 2009.

[51] Xiuya Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):631–645, 2007.

[52] Peter Spirtes, Clark Glymour, and Richard Scheines. Causation, prediction, and search. *MIT Press*, 2000.

[53] Statistische Ämter des Bundes und der Länder. Regionalstatistik (genesis-online), 2024. Official German regional statistics database.

[54] R Core Team. R: A language and environment for statistical computing. 2023.

[55] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020.

[56] Michael Waskom. Seaborn: Statistical data visualization. Journal of Open Source Software, 2021.

[57] Y. Xia and J. Wu. A survey on semi-supervised learning. *Journal of Computer Science and Technology*, 30(3):1–20, 2015.

[58] Sandeep Yaramakala and Dimitris Margaritis. Speculative markov blanket discovery for optimal feature selection. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4 pp.–, Houston, TX, USA, 2005. IEEE.

Appendix

A Additional tables

Socioeconomical features list:

Table 16: Population Demographics (Features 1–24)

Overall Count	Contexts	Specific Count	Features
1	**Population Demographics** (Gender, Age, Marital Status, Nationality, Immigration)	1	Total_population
2		2	Proportion_male
3		3	Proportion_female
4		4	Age_<18
5		5	Age_19-39
6		6	Age_40-59
7		7	Age_60-74
8		8	Age_75+
9		9	Single
10		10	Married/Partnered
11		11	Widowed
12		12	Divorced
13		13	Status_unknown
14		14	German_nationals
15		15	Immigrants
16		16	Immigration_history
17		17	First_gen_immigrants
18		18	Immigrant_descendants
19		19	One-sided_descendants
20		20	No_immigration_hist
21		21	Kreisfreie_Stadt
22		22	Kreis
23		23	Landkreis
24		24	Stadtkreis

Table 17: Household Economics (Features 25–53)

Overall Count	**Contexts**	**Specific Count**	**Features**
25	**Household Economics** (GDP, GVA, Income, Purchasing Power, Living Standards)	1	Total_GDP_10k_EUR
26		2	GDP_per_worker_EUR
27		3	GDP_per_capita_EUR
28		4	Total_GVA_1000_EUR
29		5	Avg_gross_salary
30		6	Median_gross_salary
31		7	Female_salary_ratio
32		8	Male_salary_ratio
33		9	Salary_ratio_25_55
34		10	Salary_ratio_55_65
35		11	Salary_ratio_65-55/25-55
36		12	Salary_ratio_professionals
37		13	Salary_ratio_manufacturing
38		14	Salary_ratio_academics
39		15	Lower_income_households
40		16	Higher_income_households
41		17	Middle_income_households
42		18	Household_income_EUR
43		19	Disposable_income
44		20	Purchasing_power_per_capita
45		21	Retail_purchasing_power
46		22	Debtor_ratio
47		23	Labor_volume
48		24	Rent_per_m^2
49		25	Living_space_per_person
50		26	Kreisfreie_Stadt_flag
51		27	Kreis_flag
52		28	Landkreis_flag
53		29	Stadtkreis_flag

Table 18: Sectorial Economics (Features 54-147)

Overall Count	Contexts	Specific Count	Features
54	**Sectorial Economics** (GDP, GVA, Employment, Registered Businesses Stats, (by sectors)	1	GDP_10k_EUR
55		2	GDP_per_worker_EUR
56		3	GDP_per_capita_EUR
57		4	GVA_1k_EUR
58		5	Employees_primary_sector
59		6	Employees_secondary_sector
60		7	Employees_tertiary_sector
61		8	GVA_primary_sector
62		9	GVA_secondary_sector
63		10	GVA_tertiary_sector
64		11	Employees_industry
65		12	Employees_service
66		13	Employees_industry_construction
67		14	Employees_industry_manufacturing
68		15	Employees_industry_creative
69		16	Employees_service_business
70		17	Employees_service_IT_sci
71		18	Employees_service_financial
72		19	Employees_service_craft
73		20	Micro_enterprises
74		21	Small_enterprises
75		22	Mid_enterprises
76		23	Large_enterprises
77		24	GVA_agriculture_1kEUR
78		25	GVA_manufacturing_1kEUR
79		26	GVA_manufacturing_total_1kEUR
80		27	GVA_construction_1kEUR
81		28	GVA_trade_transport_1kEUR
82		29	GVA_finance_realestate_1kEUR
83		30	GVA_public_services_1kEUR
84		31	Registered_businesses
85		32	Businesses_mining

Overall Count	Contexts	Specific Count	Features
86		33	Businesses_manufacturing
87		34	Businesses_energy
88		35	Businesses_water
89		36	Businesses_construction
90		37	Businesses_retail
91		38	Businesses_transport
92		39	Businesses_hospitality
93		40	Businesses_IT
94		41	Businesses_finance
95		42	Businesses_realestate
96		43	Businesses_professional
97		44	Businesses_other
98		45	Businesses_education
99		46	Businesses_healthcare
100		47	Businesses_arts
101		48	Businesses_other_services
102		49	New_businesses_2023
103		50	Startups_2023
104		51	Business_closures_2023
105		52	Salaried_employed_total
106		53	Employed_agriculture
107		54	Employed_manufacturing
108		55	Employed_manufacturing_total
109		56	Employed_construction
110		57	Employed_trade_transport
111		58	Employed_finance_services
112		59	Employed_public_services
113		60	Self_employed_total
114		61	Self_employed_agriculture
115		62	Self_employed_manufacturing
116		63	Self_employed_manufacturing_total
117		64	Self_employed_construction
118		65	Self_employed_trade_transport

Overall Count	Contexts	Specific Count	Features
119		66	Self_employed_finance_services
120		67	Self_employed_public_services
121		68	Hours_worked_salaried
122		69	Hours_agriculture
123		70	Hours_manufacturing
124		71	Hours_manufacturing_total
125		72	Hours_construction
126		73	Hours_trade_transport
127		74	Hours_finance_services
128		75	Hours_public_services
129		76	Hours_worked_self_employed
130		77	Self_emp_hours_agriculture
131		78	Self_emp_hours_manufacturing
132		79	Self_emp_hours_manufacturing_total
133		80	Self_emp_hours_construction
134		81	Self_emp_hours_trade_transport
135		82	Self_emp_hours_finance
136		83	Self_emp_hours_finance_total
137		84	Self_emp_hours_public_services
138		85	Self_emp_hours_public_services_total
139		86	Self_emp_hours_services_total
140		87	Kreisschluessel_Stadt_Flag
141		88	Kreis_Flag
142		89	Landreis

Table 19: Workforce Composition and Demographics (Features 148–191)

Overall Count	**Contexts**	**Specific Count**	**Features**
148	**Workforce Composition** and Demographics (Breakdown by Qualifications, Nationality and Gender)	1	Total_population
149		2	Total_employees
150		3	Native_employees_ratio
151		4	Foreign_employees_ratio
152		5	Native_male_employees_ratio
153		6	Native_female_employees_ratio
154		7	Foreign_male_employees_ratio
155		8	Foreign_female_employees_ratio
156		9	Native_with_academic_qual_ratio
157		10	Overall_with_academic_qual_ratio
158		11	Native_with_prof_qual_ratio
159		12	Overall_with_prof_qual_ratio
160		13	Native_with_no_qual_ratio
161		14	Overall_with_no_qual_ratio
162		15	Male_with_academic_ratio_total
163		16	Male_with_academic_ratio_males
164		17	Male_with_prof_qual_ratio_total
165		18	Male_with_prof_qual_ratio_males
166		19	Male_with_no_qual_ratio_total
167		20	Male_with_no_qual_ratio_males
168		21	Female_with_academic_ratio_total
169		22	Female_with_academic_ratio_females
170		23	Female_with_prof_qual_ratio_total
171		24	Female_with_prof_qual_ratio_females
172		25	Female_with_no_qual_ratio_total
173		26	Female_with_no_qual_ratio_females
174		27	Foreign_with_academic_ratio_total
175		28	Foreign_academic_ratio_foreign
176		29	Foreign_prof_qual_ratio_total
177		30	Foreign_prof_qual_ratio_foreign
178		31	Foreign_emp_with _no_qual_ratio_total

Overall Count	Contexts	Specific Count	Features
179		32	Foreign_emp_with _no_qual_ratio_foreign
180		33	Foreign_male_emp_with _academic_ratio_total
181		34	Foreign_male_emp_with _academic_ratio_foreign_males
182		35	Foreign_male_emp_with _prof_qual_ratio_total
183		36	Foreign_male_emp_with _prof_qual_ratio_foreign_males
184		37	Foreign_male_emp_with _no_qual_ratio_total
185		38	Foreign_male_emp_with_no _qual_ratio_foreign_males
186		39	Foreign_female_emp_with _academic_ratio_total
187		40	Foreign_female_emp_with _academic_ratio_foreign_females
188		41	Foreign_female_emp_with _prof_qual_ratio_total
189		42	Foreign_female_emp_with_prof_qual _ratio_foreign_females
190		43	Foreign_female_emp_with _no_qual_ratio_total
191		44	Foreign_female_emp_with_no_qual _ratio_foreign_females

Table 20: Workforce Participation and Unemployment (Features 192–216)

Overall Count	**Contexts**	**Specific Count**	**Features**
192	**Unemployment** (Labor force participation and unemployment rates)	1	Total_population
193		2	Employment_per_1000
194		3	Job_density
195		4	Female_employment_rate
196		5	Male_employment_rate
197		6	Foreigner_employment_rate
198		7	Young_employees_ratio
199		8	Senior_employees_ratio
200		9	Labor_participation_rate
201		10	Women_participation_rate
202		11	Men_participation_rate
203		12	Self_employment_rate
204		13	Unemployment_rate
205		14	Immigrant_unemployment_rate
206		15	Foreigner_share_unemployed
207		16	Disabled_unemployed_share
208		17	Disabled_unemployment_rate
209		18	Youth_15_20_unemployment
210		19	Youth_15_20_share_unemployed
211		20	Youth_15_25_unemployment
212		21	Youth_15_25_share_unemployed
213		22	Senior_55_65_unemployment
214		23	Senior_55_65_share_unemployed
215		24	Longterm_unemployment_rate
216		25	Longterm_share_unemployed

Table 21: Vehicle and Crime Statistics (Features 217–232)

Overall Count	**Contexts**	**Specific Count**	**Features**
217	**Vehicle and Crime Statistics** (Crime rates, vehicle types, and charging infrastructure)	0	Crime_rate_per_100k
218		1	Burglary_rate_per_100k
219		2	Cars_per_1000
220		3	Petrol_cars_percentage
221		4	Diesel_cars_percentage
222		5	Gas_cars_percentage
223		6	Hybrid_cars_percentage
224		7	Plugin_hybrid_percentage
225		8	Electric_cars_percentage
226		9	Other_fuel_percentage
227		10	Chargers_per_100k
228		11	Chargers_per_100_EVs
229		12	Kreisfreie_Stadt_flag
230		13	Kreis_flag
231		14	Landkreis_flag
232		15	Stadtkreis_flag

Table 22: Income Taxpayer and Tax Revenue Metrics (Features 233–291)

Overall Count	Contexts	Specific Count	Features
233	**Income Taxpayer and Tax Revenue Metrics** (Counts and taxes across income ranges, tax revenue, property/trade taxes)	1	Taxpayers_income_0
234		2	Taxpayers_income_1_5k
235		3	Taxpayers_income_10k_15k
236		4	Taxpayers_income_125k_plus
237		5	Taxpayers_income_15k_20k
238		6	Taxpayers_income_20k_25k
239		7	Taxpayers_income_25k_30k
240		8	Taxpayers_income_30k_35k
241		9	Taxpayers_income_35k_50k
242		10	Taxpayers_income_5k_10k
243		11	Taxpayers_income_50k_125k
244		12	Total_taxpayers
245		13	Income_amount_1k_5k
246		14	Income_amount_10k_15k
247		15	Income_amount_125k_plus
248		16	Income_amount_15k_20k
249		17	Income_amount_20k_25k
250		18	Income_amount_25k_30k
251		19	Income_amount_30k_35k
252		20	Income_amount_35k_50k
253		21	Income_amount_5k_10k
254		22	Income_amount_50k_125k
255		23	Total_income_amount
256		24	Tax_income_range_0
257		25	Tax_income_range_1k_5k
258		26	Tax_income_range_10k_15k
259		27	Tax_income_range_125k_plus
260		28	Tax_income_range_15k_20k
261		29	Tax_income_range_20k_25k
262		30	Tax_income_range_25k_30k
263		31	Tax_income_range_30k_35k
264		32	Tax_income_range_35k_50k

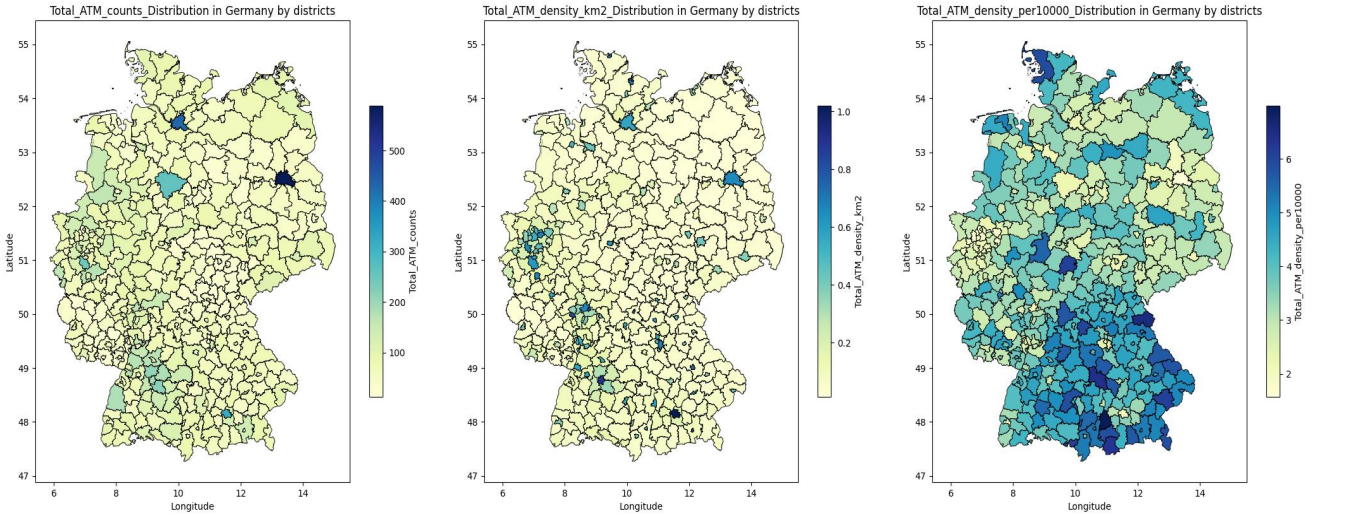
Overall Count	Contexts	Specific Count	Features
265		33	Tax_income_range_5k_10k
266		34	Tax_income_range_50k_125k
267		35	Total_income_tax
268		36	Property_tax_A_revenue
269		37	Property_tax_A_revenue_ratio
270		38	Property_tax_B_revenue
271		39	Property_tax_B_revenue_ratio
272		40	Trade_tax_revenue
273		41	Trade_tax_revenue_ratio
274		42	Property_tax_A_base
275		43	Property_tax_B_base
276		44	Trade_tax_base
277		45	Property_tax_A_assessment_rate
278		46	Property_tax_B_assessment_rate
279		47	Trade_tax_assessment_rate
280		48	Real_tax_collection_power
281		49	Trade_tax_levy
282		50	Net_trade_tax
283		51	Municipal_income_tax_share
284		52	Municipal_sales_tax_share
285		53	Total_tax_revenue
286		54	Tax_collection_efficiency
287		55	Trade_tax_levy_percent
288		56	Kreisfreie_Stadt_flag
289		57	Kreis_flag
290		58	Landkreis_flag
291		59	Stadtkreis_flag

Table 23: Land Use and Infrastructure (Features 292–313)

Overall Count	**Contexts**	**Specific Count**	**Features**
292	**Land Use and Infrastructure** (Land area, traffic areas, and settlement breakdowns)	1	Total_land_area_ha
293		2	Traffic_area_percentage
294		3	Road_area_of_traffic
295		4	Road_area_of_total
296		5	Path_area_of_traffic
297		6	Path_area_of_total
298		7	Square_area_of_traffic
299		8	Square_area_of_total
300		9	Railway_area_of_traffic
301		10	Railway_area_of_total
302		11	Settlement_area_percentage
303		12	Residential_of_settlement
304		13	Residential_of_total
305		14	Industrial_commercial_of_settlement
306		15	Industrial_commercial_of_total
307		16	Industry_commerce_of_settlement
308		17	Industry_commerce_of_total
309		18	Mixed_use_of_total
310		19	Special_function_of_settlement
311		20	Special_function_of_total
312		21	Sports_leisure_of_settlement
313		22	Sports_leisure_of_total

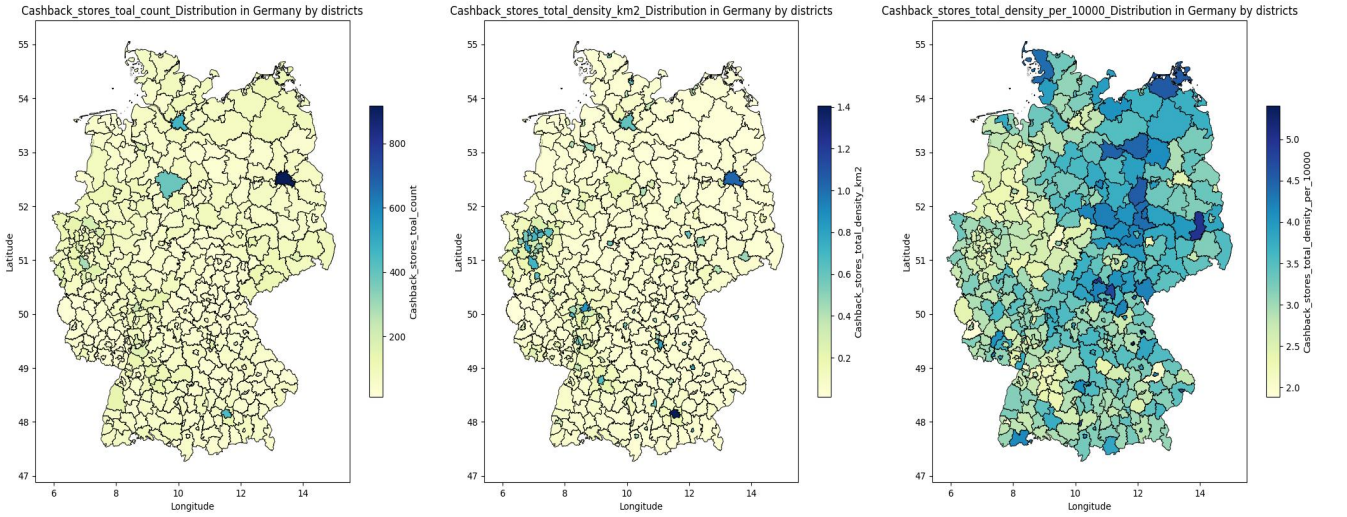
B Additional figures

All maps in this section has been generated using Geopandas (17) and Matplotlib (22).



(a) ATMs total count (b) ATMs density (km^2) (c) ATMs density(per 10,000 head)

Figure 11: ATMs distribution across Germany



(a) Cashbackstores total count (b) Cashbackstore density (km^2) (c) Cashbackstores density (per 10,000 head)

Figure 12: Cashbackstores distribution across Germany

Figure 13: Germany 400 Administrative Districts Map with labels

C Benchmark datasets description

All the datasets with pre-lebeled contextual anomalies are obtained from the research work of another paper themed on contextual anomaly detection (28). The labels were derived by simply perturbing the random observations from the observations with similar contexts as per the authors.

- **Abalone:** The dataset contains 4177 abalone physical measurement records. Especially, we use the features Sex, Length, Diameter and Height as contextual features. Accordingly, we use the features Whole Weight, Shucked Weight, Viscera Weight, Shell Weight and Rings as behavioral features.
- **Airfoil Self-Noise Dataset**: The dataset comprises 1503 experimental records from aerodynamic and acoustic tests performed on airfoil blade sections. The contextual attributes include frequency, angle of attack, chord length, free-stream velocity, and suction side displacement thickness (denoted as f, α, c, U_∞, and δ respectively). The target attribute is the scaled sound pressure level (SSPL).The dataset contains complete records with no missing values.
- **Bodyfat Dataset**: Consisting of measurements from 252 male subjects, this dataset estimates body density and body fat percentage (behavioral attributes) using various body measurements (contextual attributes). These include Age, Weight, Height, Neck circumference, Chest circumference, Abdomen circumference, Hip circumference, Thigh circumference, Knee circumference, Ankle circumference, Biceps circumference, Forearm circumference, and Wrist circumference. The original data contains no categorical variables or missing values.
- **Boston Housing Dataset**: With 506 entries after cleaning, this collection contains information about Boston area housing. Contextual attributes describe neighborhood characteristics and property features (CRIM, ZN, INDUS, CHAS, NOX, RM, AGE, DIS, RAD, TAX, PTRATIO, B, LSTAT), while the behavioral attribute (MED) represents median home values.
- **Concrete Compressive Strength Dataset**: This dataset contains 1030 records documenting concrete composition and strength. The contextual attributes (C1, C2, C3, C4, C5, C6, C7, and Age) describe mixture components and curing time, while Strength serves as the behavioral attribute. All records are complete with no missing values.
- **Energy Efficiency Dataset**: This collection comprises 768 records evaluating building energy performance based on architectural parameters. The contextual attributes include eight building characteristics (X1, X2, ...,X8), while the behav-

ioral attributes measure heating load (Y1) and cooling load (Y2). The dataset contains complete records without missing values.

- **Fish Weight Dataset**: Containing 157 entries of market fish measurements, this dataset uses species information and dimensional measurements (Species, Length1, Length2, Length3, Height, Width) as contextual attributes, with Weight as the behavioral attribute. All records are complete with no missing data.
- **Forest Fires Dataset**: This collection documents 517 instances of wildfire occurrences in northeastern Portugal. Spatiotemporal coordinates (X, Y, month, day) serve as contextual attributes, while meteorological and fire severity indices (FFMC, DMC, DC, ISI, temp, RH, wind, rain, area) function as behavioral attributes contains no missing values.
- **Gas Turbine Emission Dataset**: Originally containing 36,733 sensor readings from a Turkish power plant (from 2011 to 2015), this dataset uses turbine operational parameters (AT, AP, AH, AFDP, GTEP, TIT, TAT, CDP) as contextual attributes. Energy output and emissions (TEY, CO, NOX) constitute the behavioral attributes. For computational efficiency, experiments used only the 7,383 records from 2015. The dataset contains complete records.
- **Heart Failure Dataset**: This clinical dataset contains records of 299 heart failure patients. Contextual attributes include demographic and health indicators (age, sex, smoking, diabetes, high_blood_pressure, anemia), while behavioral attributes comprise clinical measurements (creatinine_phosphokinase, ejection_fraction, platelets, serum_creatinine, serum_sodium, time). The death_event feature was excluded. The dataset contains no missing values.
- **Hepatitis Dataset**: With 615 records of blood donor and patient laboratory values, this dataset uses demographic information (sex, age, donor Category) as contextual attributes and blood test results (ALB, ALP, ALT, AST, BIL, CHE, CHOL, CREA, GGT, PROT) as behavioral attributes. After removing 26 incomplete records from the original repository, 589 complete cases remain.
- **Indian Liver Patient Dataset**: This collection contains 583 medical records (after removing 4 incomplete cases) of Indian liver patients. Contextual attributes include Age, Gender and Selector, while behavioral attributes measure liver function through various biomarkers (Total_Bilirubin through Albumin_and_Globulin_Ratio).
- **Parkinson's Telemonitoring Dataset**: This biomedical dataset contains 5,875 voice measurement records from 42 Parkinson's patients. Contextual attributes include patient metadata and voice characteristics (subject through PPE), while behavioral attributes measure symptom severity (motor_UPDRS, total_UPDRS), with no missing values.

- **Power Plant Dataset**: Featuring 9,568 operational records from a combined cycle plant (2006-2011), this dataset uses ambient conditions (T, AP, RH, EP) as contextual attributes and energy output (EP) as behavioral attribute. Available with complete records.
- **Toxicity Dataset**: With 908 chemical compounds, this QSAR dataset uses molecular descriptors as contextual attributes and acute aquatic toxicity measurements as behavioral attribute, with complete records.
- **Yacht Hydrodynamics Dataset**: Containing 308 yacht design records, this dataset uses dimensional and hydrodynamic parameters (Longitudinal_position through Froude_number) as contextual attributes and resistance as behavioral attribute. Available with complete records.

Eidesstattliche Versicherung

(Affidavit)

Patel, Nidhi Kiritbhai — 230199

Name, Vorname (surname, first name) — Matrikelnummer (student ID number)

☐ Bachelorarbeit (Bachelor's thesis) ☒ Masterarbeit (Master's thesis)

Titel (Title)

Context Matters: Contextual Anomaly Detection in Cash Withdrawal Resource Distribution

Ich versichere hiermit an Eides statt, dass ich die vorliegende Abschlussarbeit mit dem oben genannten Titel selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

I declare in lieu of oath that I have completed the present thesis with the above-mentioned title independently and without any unauthorized assistance. I have not used any other sources or aids than the ones listed and have documented quotations and paraphrases as such. The thesis in its current or similar version has not been submitted to an auditing institution before.

Dortmund, 04/05/2025

Ort, Datum (place, date) — Unterschrift (signature)

Belehrung:
Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -).

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin") zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Official notification:
Any person who intentionally breaches any regulation of university examination regulations relating to deception in examination performance is acting improperly. This offense can be punished with a fine of up to EUR 50,000.00. The competent administrative authority for the pursuit and prosecution of offenses of this type is the Chancellor of TU Dortmund University. In the case of multiple or other serious attempts at deception, the examinee can also be unenrolled, Section 63 (5) North Rhine-Westphalia Higher Education Act (*Hochschulgesetz, HG*).

The submission of a false affidavit will be punished with a prison sentence of up to three years or a fine.

As may be necessary, TU Dortmund University will make use of electronic plagiarism-prevention tools (e.g. the "turnitin" service) in order to monitor violations during the examination procedures.

I have taken note of the above official notification:*

Dortmund, 04/05/2025

Ort, Datum (place, date) — Unterschrift (signature)

***Please be aware that solely the German version of the affidavit ("Eidesstattliche Versicherung") for the Bachelor's/ Master's thesis is the official and legally binding version.**