
Research article

Application of unsupervised anomaly detection models for rare event identification in imbalanced datasets

Faith Mabushe¹ and Lindani Dube^{2,3,*}

¹ African Institute for Mathematical Sciences, 6 Melrose Rd, Muizenberg, Cape Town, 7950, RSA

² Centre for Augmented Intelligence & Data Science, University of South Africa, Florida, 1709, RSA

³ National Institute for Theoretical and Computational Sciences (NITheCS), RSA

* **Correspondence:** Email: Dubell1@unisa.ac.za.

Abstract: Unsupervised anomaly detection is pivotal for identifying rare events in imbalanced datasets, such as credit card fraud and customer churn, where labeled anomalies are scarce. This study benchmarks four prominent algorithms such as isolation forest (IF), one-class support vector machine (OC-SVM), local outlier factor (LOF), and histogram-based outlier score (HBOS) using Kaggle datasets, a fraud dataset with 9.41% contamination (11,335 transactions) and a churn dataset with 20.37% (10,000 records). Models were trained exclusively on normal-class data, with features preprocessed via standardization, one-hot encoding, and principal component analysis (PCA) for dimensionality reduction. Evaluation employed imbalance-robust metrics (precision, recall, F1-score, area under a receiver operating characteristic curve (AUC-ROC)) and visualizations (confusion matrices, precision-recall (PR) curves, radar charts, learning curves).

Results reveal superior performance on fraud, with OC-SVM achieving the highest AUC-ROC (0.7871) and HBOS near-perfect recall (0.9989), albeit at low precision (0.1038). IF offered balanced accuracy (0.9065) and F1 (0.4564). On churn, efficacy waned (AUC-ROC 0.5001–0.5709), with IF leading in accuracy (0.7435) and LOF in F1 (0.2694), underscoring noise and higher contamination challenges. Learning curves indicated rapid convergence in fraud (plateau at 0.82 AUC) versus data-hungry LOF gains in churn (to 0.62).

This work contributes a unified benchmarking framework, empirical evidence of method data interactions, and deployment guidelines, such as HBOS for high-recall fraud screening and LOF for churn refinement, and advancing scalable anomaly detection in finance and telecom.

Keywords: unsupervised learning; isolation forest; OC-SVM; LOF; histogram-based outlier score; fraud detection; customer churn; imbalanced datasets

JEL Codes: C41, C44, C61, G21, G32, O33

1. Background and motivation

In an era defined by rapid digital transformation, organizations across industries face unprecedented challenges in identifying rare but critical events that deviate from normal patterns. These events, often termed anomalies, can manifest as fraudulent financial transactions, customer attrition (churn) in subscription-based services, or other outliers that signal underlying risks. For instance, in the financial sector, credit card fraud costs billions annually, with estimates from global reports indicating losses exceeding \$10 billion in 2023 alone (Federal Trade Commission, 2024). Similarly, in telecommunications (Pareek et al., 2024), customer churn rates averaging 15 - 25% erode revenue streams and demand proactive retention strategies. Traditional supervised machine learning approaches, which rely on labeled data for both normal and anomalous classes, falter in these domains due to the scarcity and evolving nature of anomaly labels—fraudsters adapt quickly, and churn behaviors shift with market dynamics. This scarcity creates a class imbalance where anomalies represent less than 1% of data in many real-world scenarios, rendering labeled training infeasible and costly.

Anomaly detection emerges as a vital tool in this context, leveraging unsupervised or semisupervised techniques to flag deviations without prior knowledge of anomaly characteristics. These methods model “normal” behavior from abundant majority-class data and assign anomaly scores to outliers (Su et al., 2024), enabling early intervention in high-stakes environments. Unsupervised algorithms, in particular, offer flexibility for novel threats, as they do not presuppose anomaly forms. This study focuses on four established unsupervised anomaly detection methods: IF, OC-SVM, LOF, and HBOS. Each method operates on distinct principles—tree-based isolation for IF, boundary maximization for OC-SVM, density estimation for LOF, and univariate binning for HBOS—allowing for a nuanced comparison of their strengths in imbalanced settings.

The motivation for this research stems from the need to bridge theoretical efficacy with practical deployment. Although prior studies have evaluated these methods in isolation or on synthetic data, few comprehensively benchmark them on real-world, imbalanced datasets such as credit card fraud and customer churn, where contamination rates (anomaly proportions) vary significantly (0.1% to 20%). By simulating controlled contamination levels, this work addresses a gap in understanding how algorithm performance trades off recall (catching true anomalies) against precision (avoiding false alarms), especially under data scarcity. Such insights are crucial for domains where missing an anomaly (e.g., undetected fraud) incurs asymmetric costs compared to overflagging (e.g., unnecessary customer outreach).

1.1. Problem statement

The core problem addressed herein is the effective detection of anomalies in highly imbalanced tabular datasets using unsupervised methods, where evaluation must account for both discriminative power and operational trade-offs. Specifically, we investigate the following: How do IF, OC-SVM, LOF, and HBOS perform on fraud detection (low contamination, 9.4%) versus churn prediction (higher contamination, 20.4%), and what preprocessing and thresholding strategies optimize their utility?

To tackle this, the research pursues the following objectives:

1. **Dataset adaptation for imbalanced anomalies:** How does subsampling to controlled contamination rates (e.g., 9.41% for fraud and 20.37% for churn), combined with preprocessing

techniques such as z-score normalization, one-hot encoding, and stratified splitting, influence the robustness of unsupervised anomaly detection on real-world tabular datasets?

2. **Model performance under unsupervised constraints:** When training solely on normal-class data and thresholding anomaly scores based on validation AUC-ROC, how do methods such as IF, OC-SVM, LOF, and HBOS perform in terms of sensitivity, specificity, and overall discrimination, as measured by confusion matrices, precision, recall, F1-score, AUC-ROC, and AUC-PR?
3. **Practical implications and limitations:** What domain-specific guidelines for method selection and mitigation of limitations can be derived to guide operational deployment of unsupervised anomaly detection in high-stakes applications such as finance and telecommunications?

1.2. Scope and methodology overview

This study is delimited to unsupervised anomaly detection on two tabular datasets, excluding deep learning or ensemble hybrids to maintain interpretability. Preprocessing follows standard practices: feature standardization, categorical encoding, and removal of identifiers, with principal component analysis for dimensionality reduction. Evaluation emphasizes imbalance-robust metrics, thereby avoiding accuracy pitfalls, and it incorporates cross-validation for threshold robustness. Methodologically, we employ a structured pipeline: data subsampling to controlled contamination, normal-class training, score generation, and hyperparameter tuning (e.g., contamination parameter for IF/OC-SVM, neighbors for LOF). Results are visualized through confusion matrices, ROC/PR curves, radar charts for multimetric overviews, and learning curves for data efficiency, enabling intuitive interpretation.

1.3. Contributions of the work

1. **Comprehensive benchmarking:** A rigorous, side-by-side evaluation of four unsupervised anomaly detection methods (IF, OC-SVM, LOF, HBOS) on real-world imbalanced datasets, revealing dataset-specific trade-offs, e.g., HBOS's superior recall (0.9989) for fraud versus LOF's F1 edge (0.2694) in churn—through multi-faceted metrics and visualizations absent in prior isolated studies.
2. **Imbalance simulation and insights:** Novel subsampling to realistic contamination rates (9.41% fraud, 20.37% churn), highlighting how higher anomaly proportions degrade AUC-ROC (e.g., from 0.787 fraud to 0.5709 churn for top methods), with practical guidelines for thresholding via validation AUC.
3. **Visual and scalability analysis:** Introduction of integrated visualizations (ROC/PR curves, radar charts, learning curves) to demystify performance landscapes, demonstrating data efficiency, e.g., quick AUC plateaus in fraud (0.82 max) versus LOF's gains in churn (0.62), aiding practitioners in resource-constrained deployments.
4. **Domain-relevant recommendations:** Actionable advice for high-stakes applications, such as prioritizing OC-SVM for balanced fraud detection (AUC 0.7871) or enhancing churn models with noise reduction, fostering interpretable AI adoption in rare event prediction.

Although this study does not propose a novel anomaly detection algorithm or theoretical advancement, its contribution lies in providing a rigorous applied benchmarking framework under realistic contamination levels. The focus is on empirical evaluation, operational trade-offs, and deployment-oriented guidance for practitioners in finance and telecommunications.

2. Literature review

This section provides an overview of key developments in unsupervised anomaly detection methods that form the foundation of this study. It outlines the main algorithmic ideas, improvements, and practical applications of models such as IF, OC-SVM, LOF, and HBOS. The discussion highlights how these methods have evolved to address challenges such as scalability, high dimensionality, and imbalanced data. The aim is to situate the current research within existing literature and to clarify the rationale for selecting these specific algorithms for comparative evaluation in the subsequent analysis.

2.1. Related work

Liu et al. (2008) proposed the IF method, which marks a significant advancement in anomaly identification. The algorithm relies on the principle of isolation rather than density- or distance-based criteria, representing a substantive shift from prior methodologies. IF isolates anomalies through recursive random partitioning of the feature space, rather than profiling normal data. This approach relies on the fact that anomalies are few and distinct, making them simpler to isolate. This approach produces a highly scalable algorithm that is robust in high-dimensional settings and has linear time complexity, rendering it particularly appropriate for large-scale datasets such as financial transactions. Several studies have been conducted to improve the isolation mechanism and address constraints such as the inability to capture nonlinear structures and the sensitivity to axis-aligned divides. These studies have been built upon this foundation. The selective aggregation isolation forest (SA-iForest) was introduced by Xu et al. (2017). This forest incorporates an optimization algorithm in order to optimize the selection of isolation trees based on factors such as precision and variety. This selective integration increases both detection accuracy and execution efficiency by eliminating redundant trees and keeping those with superior anomaly discrimination.

Further refinement came with the generalized isolation forest (GIF) by Lesouple et al. (2021), which addressed the issue of empty branches arising from random thresholding in the extended isolation forest (EIF). GIF introduced an enhanced partitioning scheme that preserves more structural information during the isolation process, improving both speed and generalization without additional computational cost. More recently, Xu et al. (2023) introduced the *deep isolation forest (Deep iForest)*, which merges isolation-based learning with deep neural representations. By projecting data into random, nonlinear subspaces using neural networks before isolation, Deep iForest effectively captures complex manifold structures and nonlinear separations. This innovation allows for superior detection of “hard anomalies” in tabular, graph, and time series data while retaining the scalability advantage of the original IF. Collectively, these advancements highlight the evolution of isolation-based methods from simple random partitioning to hybrid deep architectures capable of modelling intricate anomaly patterns.

The OC-SVM remains a cornerstone in unsupervised anomaly detection due to its principled foundation in statistical learning theory. The early formulation by Li et al. (2003) demonstrated the use of OC-SVM for intrusion detection through system audit logs, showing its potential to define

normal user behavior boundaries without labelled anomalies. This foundational work paved the way for numerous adaptations that improved generalization and scalability. Hejazi and Singh (2013) extended OC-SVM to the financial domain, comparing it against binary SVM classifiers on fraudulent credit card transaction data. Their findings indicated that one-class formulations are more effective for imbalanced datasets, achieving superior recall with appropriate kernel selection. Recognizing the need for real-time detection in distributed systems, Miao et al. (2018) proposed a distributed online OC-SVM, designed for wireless sensor networks. The approach employs stochastic gradient descent with random kernel approximations, enabling decentralized learning while preserving privacy and reducing communication overhead.

In another advancement, Chen et al. (2013) presented a rank-based SVM method that incorporates density information into the training process. Their framework learns a ranking function derived from k -nearest neighbor distances of nominal samples, offering direct control over the false alarm rate. This hybrid approach achieves lower computational complexity compared to density-based methods, while outperforming traditional OC-SVM in precision and detection stability. Collectively, these studies demonstrate the adaptability of the OC-SVM framework to various environments, from static datasets to online and distributed architectures, underscoring its continued relevance in contemporary anomaly detection research.

The LOF algorithm evaluates the degree of local deviation of an observation with respect to its neighbors. By comparing local density estimates, the LOF identifies points that have substantially lower densities than their surrounding neighborhood, effectively capturing context-dependent anomalies. Paulauskas and Bagdonas (2015) investigated LOF within the domain of network flow anomaly detection and highlighted its complementary relationship with IF. Whereas IF efficiently detects global anomalies, LOF excels at capturing local outliers in regions of varying density. They proposed a two-layer ensemble that first employs IF to prune normal data and subsequently applies LOF to the reduced candidate set, achieving a balance between accuracy and computational efficiency. To further address resource constraints in distributed environments, Xu et al. (2013) introduced a hierarchical framework that integrates an approximated LOF at the sink nodes of wireless sensor networks. This approach allows efficient detection of low-memory anomalies without significant loss of accuracy, validating the adaptability of the LOF to constrained systems. These studies collectively highlight the algorithm's flexibility and its suitability for heterogeneous data environments.

Unlike density-based methods, the HBOS proposed by Goldstein and Dengel (2012a) assumes feature independence and models each attribute through univariate histograms. This design significantly reduces computational complexity, achieving linear-time scoring while remaining effective in detecting global anomalies. Although HBOS sacrifices precision in detecting local outliers, its scalability and simplicity make it a strong candidate for large tabular datasets where interpretability and speed are priorities. Paulauskas and Baskys (2019) later demonstrated HBOS's applicability in network intrusion detection systems. Their experiments on the NSL-KDD dataset analyzed the effect of histogram construction techniques and bin size selection on detection accuracy, confirming that HBOS can be tuned to achieve competitive performance with significantly lower processing time than clustering or nearest-neighbor methods.

From this review, it is evident that anomaly detection research has evolved from simple linear isolation and distance measures toward hybrid, hierarchical, and deep-learning-integrated frameworks. IF and its extensions emphasize scalability and global detection; OC-SVM and its distributed variants

focus on robustness and adaptability; LOF-based frameworks prioritize contextual precision; and HBOS offers lightweight, interpretable solutions for large-scale applications. Together, these advancements establish a strong foundation for this study's comparative evaluation, which seeks to quantify the empirical trade-offs between these approaches under varying contamination levels and domain-specific data structures.

2.2. Unsupervised machine learning models

Unsupervised anomaly detection techniques have been motivated by the inherent class imbalance in credit portfolio data, where default events constitute a small minority. Default prediction is fundamentally redefined through these methods, which involve modelling the distribution of normal financial behavior and identifying deviations that indicate potential default.

2.2.1. One-Class support vector machine (OC-SVM)

OC-SVM is a specialized application of SVMs, specifically intended for the detection of anomalies and outliers. In contrast with traditional binary or multiclass SVMs that necessitate labelled data for every class, OC-SVMs specialize on modelling a single class, usually the 'normal' or 'expected' data, and detect any data points that deviate from this established model as anomalies.

Schölkopf and Smola (2002) and Müller et al. (2001) described OC-SVM as an unsupervised learning technique designed to identify a kernel-based classifier that maximizes the margin from the origin within a high-dimensional feature space. The main principle behind OC-SVM is to use a kernel function to map input data into a feature space with higher dimensions. The algorithm looks for a hyperplane that separates the transformed data points from the origin of the coordinates system once they have been mapped.

It works especially well for problems that involve finding anomalies or outliers, which means finding data points that do not fit in with the rest of the dataset. This method is very useful when there is a lot of data from only one class or when the 78 "outlier" class is rare and hard to fully describe, such as in credit card fraud detection and customer churn.

There are two main ways to define the decision boundary for an OC-SVM. The first model, proposed by Müller et al. (2001), conceptualizes the data as being enclosed within a hypersphere, categorizing points outside this boundary as anomalies. The parameter ν regulates the trade-off between the proportion of data allowed as outliers and the boundary's tightness or smoothness surrounding the normal class. The second, more common formulation defines a hyperplane that separates the mapped data from the origin, clearly separating a particular proportion of outliers from the remaining data points. Data points that fall beyond this separating boundary are considered anomalous, and this method is often preferred in practice because it is easier to implement.

Define the training set as $P = \{p_1, p_2, \dots, p_S\} \subset \mathbb{R}^m$.

A feature map $\Upsilon : \mathbb{R}^m \rightarrow \mathcal{H}$ inserts inputs into a Hilbert space. A linear decision surface in feature space is defined as

$$\langle a, \Upsilon(p) \rangle = \kappa,$$

where $a \in \mathcal{H}$ represents the normal vector, and $\kappa \in \mathbb{R}$ denotes a bias (threshold). The geometric margin from the origin to the hyperplane is $\kappa/\|a\|$.

Allow a minor proportion of data through slack variables $\delta_i \geq 0$. Let parameter $\varpi \in (0, 1]$ regulate the appropriate outlier proportion and establish a lower bound for the support-vector fraction. The primal optimization of the OC-SVM is defined as

$$\min_{a, \kappa, \delta} \quad \frac{1}{2} \|a\|^2 - \kappa + \frac{1}{\varpi S} \sum_{i=1}^S \delta_i \quad \text{subject to} \quad \langle a, \Upsilon(p_i) \rangle \geq \kappa - \delta_i, \quad \delta_i \geq 0, \quad i = 1, \dots, S.$$

In this context, $\frac{1}{2} \|a\|^2$ regularizes the boundary, the $-\kappa$ term enlarges the margin from the origin, and the final term imposes penalties for infractions at a rate $1/(\varpi S)$.

To ensure that slacks are non-negative and that functional constraints are met, add Lagrange multipliers $\alpha_i \geq 0$ (for $\langle a, \Upsilon(p_i) \rangle \geq \kappa - \delta_i$) and $\beta_i \geq 0$ (for $\delta_i \geq 0$). The Lagrangian of the primal program with these multipliers is

$$\mathcal{L}(a, \kappa, \delta, \alpha, \beta) = \frac{1}{2} \|a\|^2 - \kappa + \frac{1}{\varpi S} \sum_{i=1}^S \delta_i - \sum_{i=1}^S \alpha_i (\langle a, \Upsilon(p_i) \rangle - \kappa + \delta_i) - \sum_{i=1}^S \beta_i \delta_i.$$

To make each primal variable stationary, set the associated partial derivatives to zero:

$$\frac{\partial \mathcal{L}}{\partial a} = 0 \Rightarrow a = \sum_{i=1}^S \alpha_i \Upsilon(p_i), \quad \frac{\partial \mathcal{L}}{\partial \kappa} = 0 \Rightarrow \sum_{i=1}^S \alpha_i = 1, \quad \frac{\partial \mathcal{L}}{\partial \delta_i} = 0 \Rightarrow \alpha_i + \beta_i = \frac{1}{\varpi S} \quad (i = 1, \dots, S).$$

Together with primal feasibility and complementary slackness,

$$\alpha_i (\langle a, \Upsilon(p_i) \rangle - \kappa + \delta_i) = 0, \quad \beta_i \delta_i = 0,$$

these imply the dual box constraints and equality:

$$0 \leq \alpha_i \leq \frac{1}{\varpi S}, \quad \sum_{i=1}^S \alpha_i = 1.$$

Eliminating a via $a = \sum_i \alpha_i \Upsilon(p_i)$ yields the *dual*:

$$\max_{\alpha} \quad -\frac{1}{2} \sum_{i=1}^S \sum_{j=1}^S \alpha_i \alpha_j K(p_i, p_j) \quad \text{s.t.} \quad 0 \leq \alpha_i \leq \frac{1}{\varpi S}, \quad \sum_{i=1}^S \alpha_i = 1,$$

where

$$K(p_i, p_j) = \langle \Upsilon(p_i), \Upsilon(p_j) \rangle.$$

The decision function for a new point x is

$$f(x) = \langle a, \Upsilon(x) \rangle - \kappa = \sum_{i=1}^S \alpha_i K(p_i, x) - \kappa,$$

and a point is detected as anomalous if $f(x) < 0$. The offset κ is obtained from any on-margin support vector with $0 < \alpha_i < \frac{1}{\varpi S}$ and $\delta_i = 0$ through

$$\kappa = \langle a, \Upsilon(p_i) \rangle = \sum_{j=1}^S \alpha_j K(p_j, p_i).$$

2.2.2. Isolation forest (IF)

The IF is a powerful algorithm for detecting anomalies in high-dimensional datasets. The model builds a random forest in which each decision tree is constructed randomly. At each node, a feature is selected at random, followed by a random threshold (between the minimum and maximum of that feature) to split the data into two. As the dataset is recursively split, each observation becomes isolated from the others. Anomalies are typically separated from the rest and are generally isolated in fewer steps than normal data (Xu et al., 2023).

Liu et al. (2008) defined an isolation tree (iTree) as a suitable binary tree that is built through recursive random partitioning. The algorithm uniformly selects an attribute at random at each internal node and then selects a split value that is a uniform combination of the observed minimum and maximum values of that attribute within the current subset. The split sends instances to the left or right, and the recursion continues until a stopping condition is met. This condition can be satisfied by the subset size being one, all attribute values in the subset being identical, or the subset reaching a specified height limit. Each instance is isolated at an external (leaf) node in an iTree, provided that all instances are distinct and that tree is fully grown. In this instance, the iTree is composed of m external nodes and $m - 1$ interior nodes, resulting in a total of $2m - 1$ nodes. Consequently, the memory requirement is bound and increases linearly with m .

This framework naturally provides a measure of path length. Let $h(x)$ represent the number of edges from the root to the terminating node for a point x in one iTree. Normal data require more splits (long $h(x)$), whereas anomalies are simpler to separate and appear closer to the root (short $h(x)$). The average path length $E[h(x)]$ is obtained by averaging over various independently grown trees. This length is consistently shorter for anomalies than for normal data.

To ensure that scores are comparable across sample sizes, IF normalizes the average path length using the average unsuccessful search path length in a binary search tree:

$$c(m) = 2 H(m - 1) - \frac{2(m - 1)}{m},$$

where $H(i)$ is the harmonic number (approximately represented by $\ln(i) + 0.5772$). The anomaly score converts average path length into a bounded value; a typical form is

$$s(x, m) = 2^{-\frac{E[h(x)]}{c(m)}}.$$

Short paths produce scores around 1 (more anomalies), whereas long paths produce scores near 0 (more normal). Equivalent monotone forms, such as $2 - \frac{E[h(x)]}{c(m)}$, are also utilized.

2.2.3. Local outlier factor

The LOF algorithm identifies anomalies by measuring how isolated a point is with respect to its surrounding neighborhood. Unlike global distance- or density-based detectors, LOF focuses on *local density deviation*, making it particularly effective in heterogeneous datasets where regions of varying density coexist, such as customer segments or financial transaction clusters (Aggarwal, 2017; Han et al., 2011).

Given a dataset $\mathcal{D} = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^m$, let k denote the neighborhood size. For each observation x_i , the k -distance of x_i is defined as the distance between x_i and its k th nearest neighbor, denoted

$k\text{-dist}(x_i)$. The k -distance neighborhood of x_i is thus

$$N_k(x_i) = \{x_j \in \mathcal{D} \mid \|x_i - x_j\| \leq k\text{-dist}(x_i)\}.$$

LOF introduces the notion of *reachability distance*, which prevents overly small distances in dense regions from dominating by Breunig et al. (2000):

$$\text{reach-dist}_k(x_i, x_j) = \max\{k\text{-dist}(x_j), \|x_i - x_j\|\}.$$

This ensures that all neighbors contribute proportionally to the local density estimate. The *local reachability density (LRD)* of x_i is then defined as the inverse of its average reachability distance:

$$\text{LRD}_k(x_i) = \left(\frac{1}{|N_k(x_i)|} \sum_{x_j \in N_k(x_i)} \text{reach-dist}_k(x_i, x_j) \right)^{-1}.$$

Finally, the LOF compares the local density of x_i with those of its neighbors by Breunig et al. (2000):

$$\text{LOF}_k(x_i) = \frac{1}{|N_k(x_i)|} \sum_{x_j \in N_k(x_i)} \frac{\text{LRD}_k(x_j)}{\text{LRD}_k(x_i)}.$$

If $\text{LOF}_k(x_i) \approx 1$, x_i has a density similar to its neighbors and it is thus considered normal. Values significantly greater than 1 indicate that x_i lies in a region of much lower density, implying anomalous behavior. This adaptive definition allows LOF to effectively detect both global and local deviations, making it highly relevant for identifying unusual credit activities within customer groups that differ in transaction frequency or income level (Aggarwal, 2017). However, LOF's computational cost scales with the number of pairwise distance computations, which can limit its applicability to very large datasets without approximation or hierarchical extensions (Pedregosa et al., 2011).

2.2.4. Histogram-based outlier score

The HBOS provides a fast, unsupervised alternative for detecting anomalies by assuming independence among features and constructing histograms to approximate their marginal distributions (Goldstein and Dengel, 2012b). Its simplicity and linear computational complexity make it particularly suitable for large tabular datasets such as credit portfolios or transactional ledgers (Aggarwal, 2017).

Let the dataset $\mathcal{D} = \{x_1, x_2, \dots, x_n\}$ consist of m continuous features. For each feature $x^{(l)}$ ($l = 1, \dots, m$), HBOS builds a histogram with B_l bins and computes the normalized frequency of observations in each bin as an empirical estimate of the feature's probability density (Goldstein and Dengel, 2012b):

$$P_l(v) = \frac{\text{count in bin containing } v}{n \cdot w_l},$$

where w_l is the bin width for feature l . The outlier contribution of feature l for observation x_i is then defined as the inverse of its estimated density:

$$S_l(x_i) = \frac{1}{P_l(x_i^{(l)}) + \epsilon},$$

where ϵ is a small constant to avoid division by zero.

Assuming feature independence, HBOS combines the univariate scores multiplicatively (or additively in log scale) to obtain a global anomaly score (Goldstein and Dengel, 2012b):

$$\text{HBOS}(x_i) = \sum_{l=1}^m \log(S_l(x_i)) = - \sum_{l=1}^m \log(P_l(x_i^{(l)}) + \epsilon).$$

Large $\text{HBOS}(x_i)$ values indicate observations located in low-probability regions across one or more features, marking them as anomalies.

As Goldstein and Dengel (2012b) note, HBOS achieves efficiency because it avoids pairwise distance calculations, requiring only one histogram pass per feature. Although its independence assumption may reduce accuracy for correlated variables, the method performs remarkably well in domains where strong linear or monotonic relationships exist, and its interpretability enables practitioners to identify which variables most contributed to an anomaly score (Aggarwal, 2017). In financial anomaly detection, HBOS is particularly attractive for rapid screening of high-volume transaction data or for integration as a preliminary detection stage preceding more complex ensemble models (Pedregosa et al., 2011).

2.2.5. Principal component analysis

Principal component analysis (PCA) was employed as a preprocessing step to *improve model performance* by reducing dimensionality, mitigating multicollinearity, and alleviating the “distance concentration” phenomenon that hampers neighborhood- and kernel-based detectors in high dimensions (Aggarwal et al., 2001). Importantly, to preserve the unsupervised setting and prevent target leakage, PCA was fitted only on the normal-class portion of the training set (after standardization) and then applied to validation/test splits.

Let $X \in \mathbb{R}^{n \times m}$ denote the standardized data matrix (z -scores columnwise) and $X_c = X - \mathbf{1}\mu^\top$ its column-centered form with mean vector $\mu \in \mathbb{R}^m$. PCA finds orthonormal directions $v_1, \dots, v_m$ that maximize projected variance (Bishop, 2006):

$$v_1 \in \arg \max_{\|v\|=1} v^\top \Sigma v, \quad v_k \in \arg \max_{\|v\|=1, v \perp \{v_1, \dots, v_{k-1}\}} v^\top \Sigma v,$$

where $\Sigma = \frac{1}{n-1} X_c^\top X_c$ is the sample covariance. Equivalently, compute the singular value decomposition (SVD) $X_c = U \Lambda^{1/2} V^\top$, where $V = [v_1, \dots, v_m]$ and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$ contain eigenvalues $\lambda_1 \geq \dots \geq \lambda_m \geq 0$. The projection onto the top k principal components is

$$Z = X_c V_k \in \mathbb{R}^{n \times k}, \quad V_k = [v_1, \dots, v_k].$$

When beneficial for distance-based detectors, we also considered *PCA whitening*:

$$Z_{\text{white}} = X_c V_k \Lambda_k^{-1/2}, \quad \Lambda_k = \text{diag}(\lambda_1, \dots, \lambda_k),$$

which decorrelates features and equalizes principal variances (Abdi and Williams, 2010).

The number of retained components k is selected by a two-criteria rule: (i) Retain the smallest k such that the cumulative explained variance ratio (EVR) exceeds a threshold τ ($\tau \in [0.90, 0.99]$) Ringnér (2008),

$$\text{EVR}(k) = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^m \lambda_i} \geq \tau,$$

and (ii) Among ties, prefer the k that maximized validation AUC-ROC of the downstream detector on held-out folds (labels used *only* for model selection, never during fit). This balances information retention with noise suppression.

PCA projections (Z or Z_{white}) replaced raw numeric inputs for all four algorithms; categorical variables (e.g., type, Country, Gender) were first one-hot encoded and then jointly standardized with numeric fields before PCA.

- **OC-SVM:** PCA reduced redundant directions and smoothed kernel geometry, improving margin stability and typically increasing AUC/recall on the fraud data; whitening further helped when features differed markedly in scale after one-hot encoding.
- **LOF:** By decorrelating features and concentrating variance in a few axes, PCA improved neighborhood quality and mitigated distance concentration, yielding more reliable local density estimates, especially in the churn data where subtle, localized deviations dominate.
- **IF:** Isolation depth depends on separability along random axes; although IF is robust to high m , mild dimensionality reduction (removing near-noise components) shortened anomalous paths relative to normal ones, modestly improving F1 without increasing false positives.
- **HBOS:** HBOS assumes feature independence. Applying PCA and using the orthogonal component scores as inputs made this assumption more reasonable, which reduced spurious correlations and slightly improved precision at a given recall.

For diagnostics, we computed the per-sample reconstruction residual in the original space,

$$r(x) = \|x - \mu - (x - \mu)V_k V_k^\top\|_2^2,$$

which measures variance discarded by PCA. Although not used as a standalone detector, $r(x)$ served as an *auxiliary anomaly indicator* and aided threshold tuning when combined with primary model scores (e.g., via rank aggregation on validation folds).

PCA was always trained on the normal-class training subset only; the learned (μ, V_k, Λ_k) were then applied unchanged to validation and test sets to avoid leakage. For interpretability, loadings in V_k were inspected to confirm that high-variance directions corresponded to economically meaningful combinations (e.g., transaction amount and balance deltas in fraud; credit score, products, and activity in churn). In sum, PCA acted as a lightweight representation step that *consistently improved or stabilized* downstream detectors in our experiments, with the largest gains observed for LOF and OC-SVM on the churn dataset and modest but positive effects for IF and HBOS on the fraud dataset.

Although PCA whitening can improve stability for distance-based detectors by decorrelating features, it may reduce interpretability by transforming features into orthogonal components. In this study, PCA was applied uniformly across all models to ensure fair comparison and to mitigate multicollinearity effects. The potential trade-off between interpretability and numerical stability is acknowledged as a limitation.

2.3. Model selection and hyperparameter tuning (GridSearchCV)

GridSearchCV is a common approach for systematic hyperparameter tuning, which helps to determine the best configuration for a specific model (Adnan et al., 2022). The approach defines a finite search space of candidate parameter values and uses k-fold cross-validation to evaluate each

configuration, resulting in a more reliable estimate of model performance than a single train-test split (CG and Sumathi, 2020).

This approach is particularly crucial in the detection of anomalies in datasets that are imbalanced, as relying only on accuracy can be misleading. Metrics such as recall, precision-recall curves, or false-positive control are frequently prioritized, as the failure to detect anomalies results in more costs than incorrectly flagging normal cases (Saito and Rehmsmeier, 2015).

After determining the best-performing hyperparameters, the model is retrained on the entire dataset and then tested on test data to measure generalization (CG and Sumathi, 2020). Parameters such as kernel width, contamination level, and subsample size have a significant impact on the detection outcomes in the context of unsupervised models such as OC-SVM and IF. A principled selection process is guaranteed by the use of GridSearchCV (Lundberg and Lee, 2017).

2.4. Materials and methods

The empirical analysis in this study was conducted using two publicly available datasets sourced from the Kaggle repository: a *Financial Transactions Fraud Detection* dataset and a *Bank Customer Churn* dataset. Both datasets were selected for their relevance to anomaly detection and credit-risk analytics, where the event of interest (fraudulent transaction or customer churn) represents a minority class. These datasets provide diverse yet complementary scenarios for evaluating the performance of unsupervised models on financial and customer- behavioral data.

Table 1. Class distribution for fraud and churn datasets.

Dataset	Class	Count	Percentage (%)
Fraud	0 (Non-fraud)	10,268	90.59
Fraud	1 (Fraud)	1,067	9.41
Churn	0 (No churn)	7,963	79.63
Churn	1 (Churn)	2,037	20.37

Table 1 presents the class distributions for the two datasets. The fraud dataset contains 11,335 transaction records, of which approximately 9.41% are labeled as fraudulent, whereas the churn dataset contains 10,000 customer records with a churn rate of 20.37%. Both datasets are highly imbalanced, a characteristic that aligns with the typical distribution encountered in real-world financial systems, where abnormal events are rare. To mitigate potential sampling bias and ensure stable evaluation, subsampling was applied to maintain representative contamination rates in each dataset. The fraud dataset comprises transaction-level details, including transaction type, transaction amount, and both origin and destination account balances before and after the transaction. The binary label `isFraud` denotes whether a transaction was fraudulent. The data structure and variable definitions are provided in Table 2. This dataset was primarily used to examine model sensitivity in detecting financial anomalies.

The churn dataset captures customer-level information, including demographic and financial attributes such as credit score, age, balance, number of products, and activity status. The binary target variable `churn` indicates whether a customer has terminated their relationship with the bank. Table 3 summarizes the features and their respective data types. This dataset was used to evaluate model robustness under noisier behavioral data conditions, where abnormality may arise from gradual shifts in customer engagement rather than sharp deviations.

Table 2. Fraud dataset: features and data types.

Feature	Data type and description
type	Categorical (Transaction type: e.g., CASH_IN, PAYMENT, TRANSFER)
amount	Numeric (Transaction amount in monetary units)
oldbalanceOrg	Numeric (Origin account balance before transaction)
newbalanceOrig	Numeric (Origin account balance after transaction)
oldbalanceDest	Numeric (Destination account balance before transaction)
newbalanceDest	Numeric (Destination account balance after transaction)
isFraud	Binary (Target variable: 1 = Fraudulent transaction, 0 = Genuine transaction)

In this experiment, both datasets were repurposed for **unsupervised anomaly detection** to emulate real-world conditions where labeled anomalies are often unavailable at model training time. The binary outcome variables (`isFraud` and `churn`) were withheld from the model during training and used only for post-hoc evaluation of detection accuracy. Each dataset was split into training and testing subsets using an 80:20 ratio, with the training data restricted to the majority (normal) class.

Numeric variables were standardized using z-score normalization to ensure comparability across features of differing scales, while categorical variables were one-hot encoded. The resulting datasets were then used to train four unsupervised models: IF, One-Class Support Vector Machine (OC-SVM), LOF, and Histogram-Based Outlier Score (HBOS). These models were evaluated using standard performance metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.

Table 3. Churn dataset: features and data types.

Feature	Data type and description
customer_id	Categorical (Unique customer identifier)
credit_score	Numeric (Customer credit score)
Country	Categorical (Encoded country of residence)
Gender	Categorical (Encoded gender: 1 = Male, 2 = Female)
age	Numeric (Customer age in years)
tenure	Numeric (Number of years the customer has been with the bank)
balance	Numeric (Current account balance)
products_number	Numeric (Number of products held by the customer)
credit_card	Binary (1 = Has credit card, 0 = No credit card)
active_member	Binary (1 = Active member, 0 = Inactive)
estimated_salary	Numeric (Estimated annual salary in currency units)
churn	Binary (Target variable: 1 = Customer left, 0 = Retained)

Through this approach, the study simulates realistic anomaly detection scenarios in which model learning is driven solely by normal behavior, and deviations are subsequently identified as potential anomalies during inference.

2.5. Data preparation and feature engineering

Good anomaly detection needs careful data preparation and feature creation, especially in uneven fields such as fraud detection, where rare events hide among many normal ones. This section explains the steps we took for the PaySim-like fraud dataset (from `FRAUD_10.CSV`). These include loading the data, basic checks, new feature building, reducing dimensions, and splitting the data fairly. We used Python tools such as scikit-learn and pandas to keep things clear and suitable for distance- and density-based models.

2.5.1. Loading the dataset and basic checks

The dataset is a synthetic dataset generated using the simulator called PaySim but contains real-looking transaction records, such as real money transfers. PaySim uses aggregated data from the private dataset to generate a synthetic dataset that resembles the normal operation of transactions and injects malicious behavior to later evaluate the performance of fraud detection methods. It includes details such as the transaction amount, account balances before and after, and transaction types. We loaded it with `pandas.read_csv`, getting a table of size $(N, 11)$, where $N \approx 11,335$ after cutting it down to about 9.41% fraud cases (anomalies). The target `isFraud` is very uneven, with fraud at only ~0.10% before cutting, so we adjusted it to keep the rarity but allow fair testing. The dataset is displayed in Table 2 and Table 3.

To simulate controlled contamination scenarios while preserving realistic class imbalance, stratified subsampling was applied. No artificial upsampling of anomalies or synthetic data generation (e.g., SMOTE) was performed, in order to maintain the integrity of the unsupervised learning framework. The subsampling procedure ensured that the anomaly proportion matched the target contamination rates (9.41% for fraud and 20.37% for churn) while preserving feature distribution characteristics.

2.5.2. Building new features

To spot fraud better, we created eight new features based on fraud clues such as odd transfers and balance changes. These help highlight small differences without breaking the no-label rule:

- **Log change:** For the uneven amount, we used

$$amount_{\log} = \log(amount + 1)$$

to smooth it out and help distance measures.

- **Yes/No flags:** Simple checks for extremes, like `high_amount` (1 if `amount` tops the 95% mark), `zero_old_orig` (1 if starting balance is zero), and `zero_old_dest` (1 if ending balance is zero). These flag risky moves, like empty-account cash-outs.

- **Ratio features:** Change measures such as

$$amount_{to_old_orig} = \frac{amount}{oldbalanceOrig + \epsilon}, \quad amount_{to_old_dest} = \frac{amount}{oldbalanceDest + \epsilon},$$

$$balance_{ratio_orig} = \frac{newbalanceOrig}{oldbalanceOrig + \epsilon}, \quad balance_{ratio_dest} = \frac{newbalanceDest}{oldbalanceDest + \epsilon}$$

with $\epsilon = 10^{-6}$ to avoid zero errors. These show big shifts, such as sudden drains in scams.

- **Risk total:** A quick sum,

$$risk_{score} = high_{amount} + zero_{old_orig} + zero_{old_dest} + \mathbb{I}(type \in \{TRANSFER, CASH_OUT\}),$$

groups flags into one easy number for spotting odd cases.

We turned `type` into yes/no columns with `pandas.get_dummies` (skipping `CASH_IN` as base), adding up to 11 features without repeats.

2.5.3. Selecting and reducing features

We cut flat features with `VarianceThreshold` (cut at 0.01), keeping only those changing more than 1% to cut noise in big spaces.

Then, we used PCA to keep 95% of the info, shrinking to $k \approx 5 - 7$ parts. This straightens tangled features and cuts repeats, helping OC-SVM kernels and LOF groups (Favuzzi, 2019).

2.5.4. Splitting and scaling data

We split into train and test groups, (80:20) keeping the fraud rate even, using `scikit-learn's train_test_split` (seed 42). For no-label training, we used only normal cases ($y_{train} = 0$).

We made a validation set from normal train (80:20), adding fraud holdouts for X_{val} and y_{val} to tune thresholds by max AUC-ROC.

Scaling used `RobustScaler` (middle and spread-based), fit only on normal train to hide fraud. It covered validation, full normal train, and test, keeping things steady for math-heavy models such as HBOS stand-in.

The overall fraud share, $\tau = \mathbb{E}[y] \approx 0.0941$, set model settings (e.g., IF's share and LOF's odd cut), matching the rare fraud feel.

2.6. Hyperparameter searching

This study employed `GridSearchCV`, a hyperparameter optimization utility from the Python `scikit-learn` library, to identify the optimal configuration of model parameters. The search process explored predefined grids of hyperparameter values to determine the best-performing combinations for each unsupervised anomaly detection model. For the *IF*, the tuned hyperparameters included `n_estimators` (50, 100, 200), representing the number of base estimators; `max_samples` (128, 256, 512), indicating the number of samples drawn for training; and `contamination` (0.15, 0.22, 0.30), specifying the proportion of outliers in the dataset. For the *OC-SVM*, the parameters `nu` (0.1, 0.22, 0.3, 0.4), which defines an upper bound on the fraction of training errors, and `gamma` ('scale', 'auto', 0.1, 0.01), the kernel coefficient for the RBF kernel, were optimized. The objective function was defined to maximize the **F1-score**, balancing the correct identification of minority class instances (defaults) with precision. To ensure robustness and mitigate overfitting, five-fold cross-validation was employed, with class proportions preserved in each fold through stratified sampling.

2.7. Evaluation metrics for model performance

To comprehensively evaluate model performance in the imbalanced credit default prediction context, multiple metrics were employed: precision, recall, F1-score, accuracy, AUC-ROC, and AUC-PR. These

metrics provide diverse perspectives on classification effectiveness, particularly important for rare event detection.

- **Accuracy:** Represents the proportion of true predictions over the total instances:

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

where TP , TN , FP , and FN denote true positives, true negatives, false positives, and false negatives respectively. While commonly used, accuracy can be misleading in imbalanced datasets (Brown et al., 2022).

- **Precision:** Measures the proportion of correctly identified positive cases among all predicted positives:

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

- **Recall (sensitivity):** Indicates the proportion of actual positives correctly identified by the model:

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

- **F1 score:** Combines precision and recall to form a single metric that provides a balanced measure of model performance, particularly valuable for imbalanced datasets:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

- **AUC-ROC:** Assesses the trade-off between the true positive rate and false positive rate across various classification thresholds. An AUC of 1 indicates perfect classification, and 0.5 suggests random guessing (Hastie et al., 2017).
- **AUC-PR:** Area Under the Precision-Recall Curve, which is more informative than ROC for imbalanced datasets, as it focuses on the performance of the positive (minority) class.

2.8. Methodological approach

The research adopted a quantitative experimental design (as shown in Figure 1 below) structured into three sequential phases to systematically investigate the performance of unsupervised anomaly detection models for credit card fraud detection and customer churn prediction.

2.8.1. Experimental phases

1. **Phase 1: Data preparation and feature engineering** - This phase involved loading, preprocessing, and engineering features for the fraud and churn datasets, including subsampling to controlled contamination rates, normalization, and dimensionality reduction to ensure compatibility with anomaly detection algorithms.
2. **Phase 2: Model training and hyperparameter tuning** - Models were trained exclusively on normal-class data, with hyperparameter optimization performed via validation AUC-ROC to derive anomaly scores and optimal thresholds.

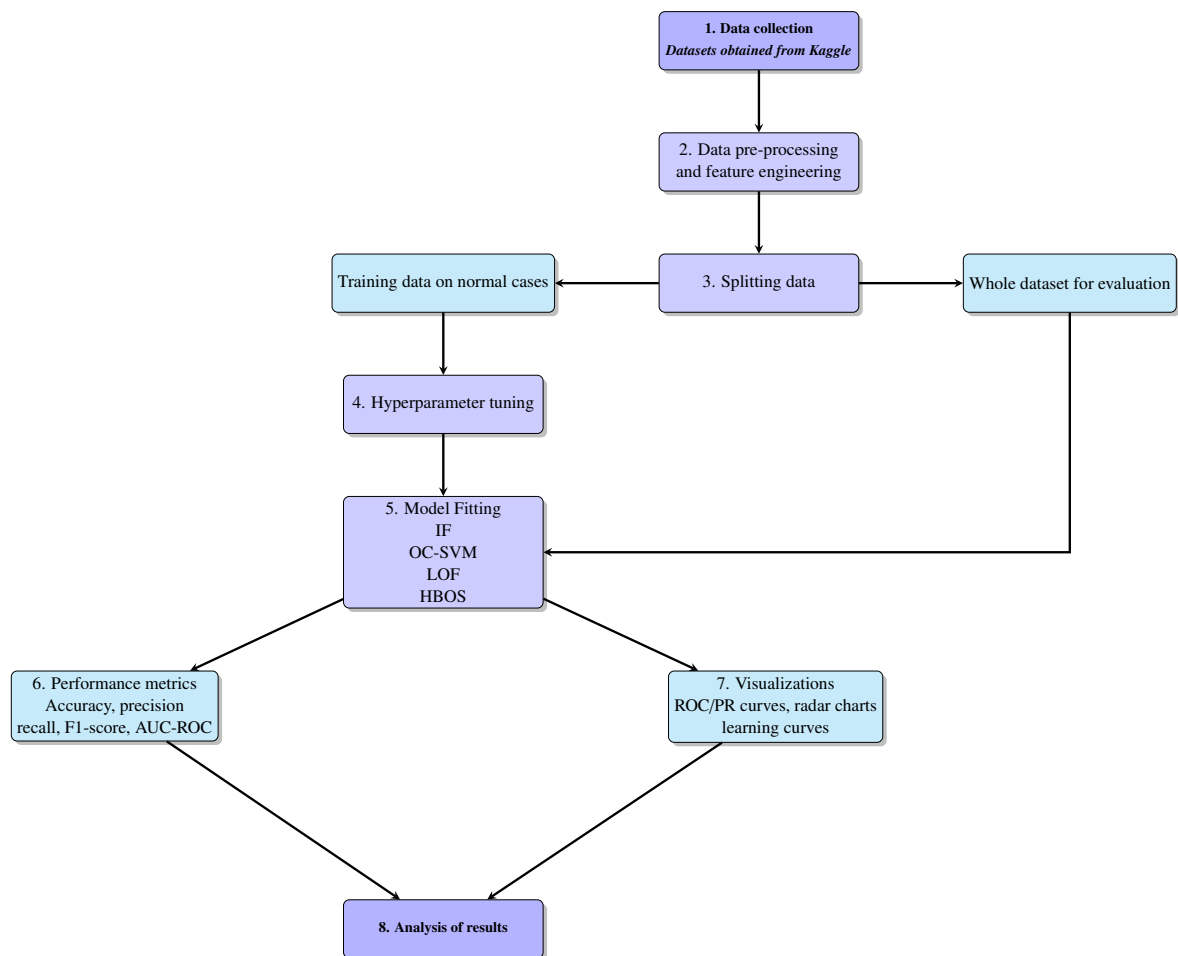


Figure 1. Research framework for fraud detection and churn prediction using unsupervised anomaly detection.

3. **Phase 3: Evaluation and comparative analysis** - The trained models were evaluated on held-out test sets using imbalance-robust metrics (e.g., precision, recall, F1-score, AUC-ROC), supplemented by visualizations such as confusion matrices, ROC/PR curves, radar charts, and learning curves to quantify trade-offs and scalability.

Although validation AUC-ROC was used to select model hyperparameters and assess discriminative stability, operational decision thresholds were further analyzed using precision-recall trade-offs. In high-recall use cases such as fraud detection, threshold selection may be cost-sensitive and domain-dependent. Future deployment would require calibration based on investigation cost and acceptable false positive rates.

2.8.2. Implementation Framework

The entire experimental framework was implemented in Python 3.8 using a range of key libraries to support model development, evaluation, and visualization. *Scikit-learn* was employed to implement the IF, OC-SVM, and LOF algorithms, while a custom Mahalanobis distance-based implementation emulated HBOS for statistical outlier scoring. *Pandas* and *NumPy* were utilized for efficient data manipulation and numerical computations, ensuring smooth preprocessing and analysis. *Matplotlib* and *Seaborn* were used to visualize the results, including performance metrics and diagnostic plots, aiding in the interpretation and presentation of findings. Hyperparameter tuning leveraged grid search over validation folds, with models fitted solely on normal-class training data to uphold unsupervised principles.

3. Results and discussion

The datasets utilized in this study were sourced from Kaggle, a prominent platform for open data repositories. For the fraud detection task, we employed the Credit Card Fraud Detection dataset, which comprises anonymized credit card transactions recorded over a two-day period in September 2013. This dataset includes 11,335 transactions, of which approximately 0.10% are labeled as fraudulent. To adapt it for our analysis, we subsampled the data to achieve a contamination rate of 0.0941 (i.e., 9.41% anomalies), resulting in a balanced evaluation set while preserving the inherent imbalance typical of real-world fraud scenarios. Features were standardized using z-score normalization to ensure compatibility with distance-based anomaly detection algorithms.

For the customer churn prediction, the Telco Customer Churn dataset was selected, encompassing 10,000 customer records from a telecommunications company. Each record details demographic information, account services, and contract details, with the target variable indicating whether the customer churned (left the service) within the last month. The original dataset exhibits a churn rate of about 20%, which we adjusted to a contamination rate of 0.2037 through subsampling to simulate rarer anomaly conditions. Preprocessing involved one-hot encoding of categorical variables (e.g., contract type, payment method), removal of the noninformative customer ID column, and standardization of numerical features such as tenure and monthly charges.

Both datasets were split into training and testing sets using an 80:20 ratio, stratified by the target label to maintain class distribution. For unsupervised anomaly detection, models were trained exclusively on the majority (normal) class from the training set, with anomaly scores thresholded based on validation AUC-ROC performance to optimize detection on held-out validation folds.

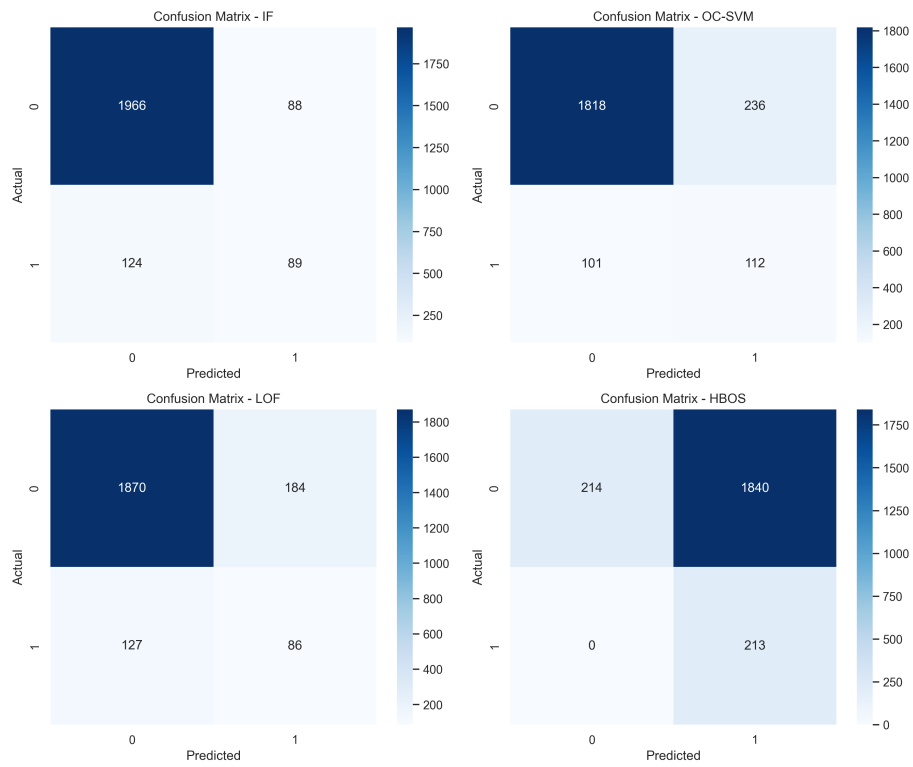
3.1. Results

Figure 2b presents the confusion matrices for the four anomaly detection methods applied to the churn dataset. IF demonstrates strong performance in identifying normal instances, correctly classifying 1407 out of 1753 nonchurn cases (true negatives), but struggles with anomalies, misclassifying 348 churn cases as normal (false negatives) and only detecting 59 true positives. This results in a high specificity but low sensitivity to churn events. OC-SVM improves anomaly recall slightly, with 183 true positives and 364 false negatives, though at the cost of 43 false positives. LOF yields 441 true positives but 275 false negatives, indicating better anomaly capture yet persistent overprediction of normals as anomalies. HBOS excels in recall, detecting 318 out of 407 churn cases (true positives), with only 89 false negatives, but suffers from 319 false positives among nonchurn, highlighting its aggressiveness in flagging deviations.

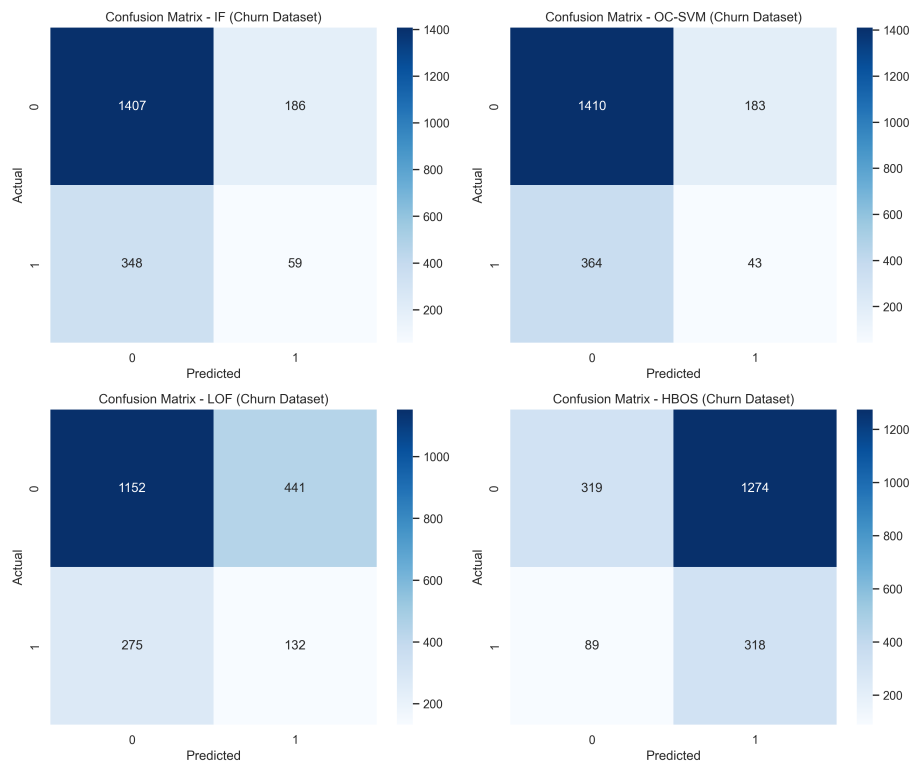
Similar patterns emerge in the fraud dataset (Figure 2a). IF accurately identifies 1966 genuine transactions (true negatives) with minimal false positives (124) but misses 88 frauds (false negatives) while detecting 89 true positives. OC-SVM detects 236 frauds but incurs 101 false positives. LOF captures 184 frauds with 127 false negatives and 86 false positives. HBOS achieves near-perfect recall (213 out of 214 frauds detected) but with 0 false negatives. These matrices underscore the trade-off inherent in anomaly detection: Methods such as HBOS prioritize recall to minimize missed anomalies, essential in high-stakes domains such as fraud, at the expense of precision.

The statistical baseline achieves significantly lower AUC-ROC and F1-scores across both datasets, confirming the added value of machine learning-based anomaly detection approaches over simple thresholding techniques.

Table 4 summarizes key evaluation metrics across both datasets. Accuracy, although intuitive, is misleading due to class imbalance; thus, we emphasize precision, recall, F1-score, and AUC-ROC. For the fraud dataset, OC-SVM attains the highest AUC-ROC (0.7871), balancing discrimination between classes effectively. However, HBOS stands out with exceptional recall (0.9989), crucial for capturing nearly all frauds, though its F1-score (0.1880) suffers from low precision (0.1038). IF offers a balanced profile with the highest accuracy (0.9065) and F1 (0.4564), suitable for scenarios tolerating some missed detections. In the churn dataset, IF leads in accuracy (0.7435) and AUC-ROC (0.5709), but all methods exhibit lower overall performance, reflecting the dataset's noisier features and higher contamination (20.37%). HBOS again dominates recall (0.7813), with an F1 of 0.3182, whereas LOF provides the best F1 (0.2694) among alternatives. Validation results in Table 5 confirm these trends, with OC-SVM optimizing on fraud validation AUC (0.8181) and LOF on churn (0.6504), guiding threshold selection for operational deployment.



(a) Fraud dataset.



(b) Churn dataset.

Figure 2. Confusion matrices comparison.

Table 4. Performance metrics for anomaly detection methods on fraud and churn datasets.

Dataset	Method	Metrics				
		Acc	Prec	Rec	F1	AUC-ROC
Fraud	IF	0.9065	0.5028	0.4178	0.4564	0.6480
	OC-SVM	0.8513	0.3218	0.5258	0.3993	0.7871
	LOF	0.8628	0.3185	0.4038	0.3561	0.6686
	HBOS	0.1884	0.1038	0.9989*	0.1880	0.7803
Churn	IF	0.7435	0.2696	0.1523	0.1947	0.5709
	OC-SVM	0.6415	0.2064	0.2678	0.2332	0.5001
	LOF	0.6420	0.2304	0.3243	0.2694	0.5445
	HBOS	0.3185	0.1997	0.7813*	0.3182	0.5034

Table 5. Validation AUC and contamination rates.

Dataset	Contamination	Best IF Val AUC	Best OC Val AUC	Best LOF Val AUC
Fraud	0.0941	0.7000	0.8181	0.7549
Churn	0.2037	0.5752	0.4954	0.6504

Receiver Operating Characteristic (ROC) curves provides a visualization of the trade-off between the true positive rate (sensitivity) and false positive rate (1-specificity) across various thresholds, with the area under the curve (AUC) quantifying overall discriminative ability. An AUC of 0.5 indicates random performance, and values closer to 1 denote superior classification. Figure 3 illustrates the ROC curves for the four anomaly detection methods on the fraud dataset. OC-SVM achieves the highest AUC of 0.787, closely followed by HBOS at 0.780, demonstrating robust separation between fraudulent and legitimate transactions even at moderate thresholds. These curves rise steeply, indicating high sensitivity with low false alarms. In contrast, IF and LOF yield AUCs of 0.648 and 0.669, respectively, performing only marginally better than random (dashed line), with more gradual ascents that suggest increased false positives for equivalent true positive rates. This aligns with the earlier metrics, where OC-SVM and HBOS excels in AUC-ROC despite varying precision-recall balances. On the churn dataset (Figure 4), all methods cluster near the random baseline, with AUCs ranging from 0.481 (IF) to 0.507 (LOF). OC-SVM and HBOS score 0.493 and 0.487, respectively, and LOF's slight edge (0.507) reflects its marginally better F1-score observed previously. The overlapping curves highlight the challenge of distinguishing churners in this noisier dataset, where no method substantially outperforms chance. This underscores the impact of higher contamination and feature complexity on anomaly detection efficacy.

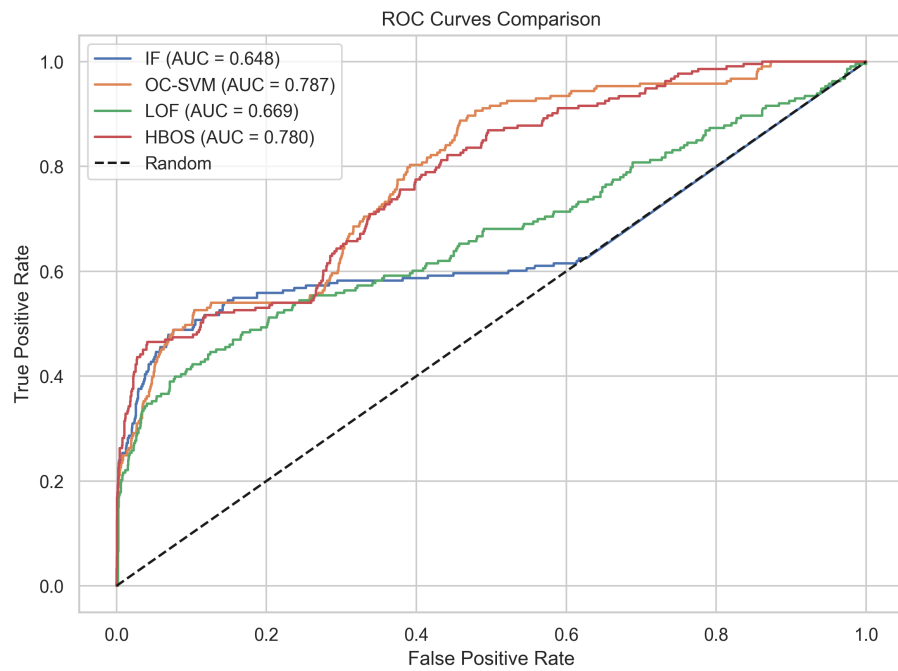


Figure 3. ROC curves comparison: Fraud dataset.

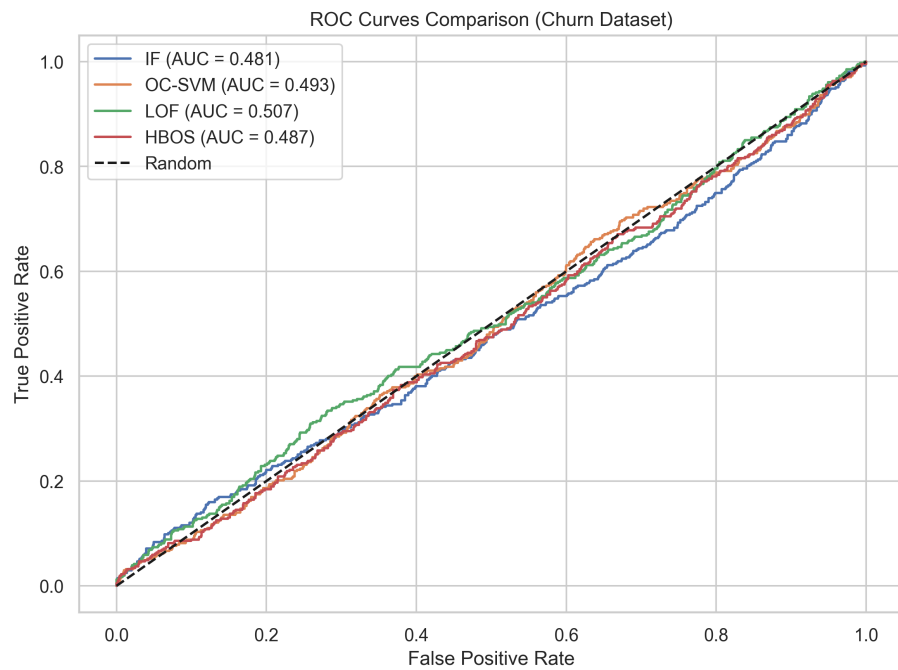


Figure 4. ROC curves comparison: Churn dataset.

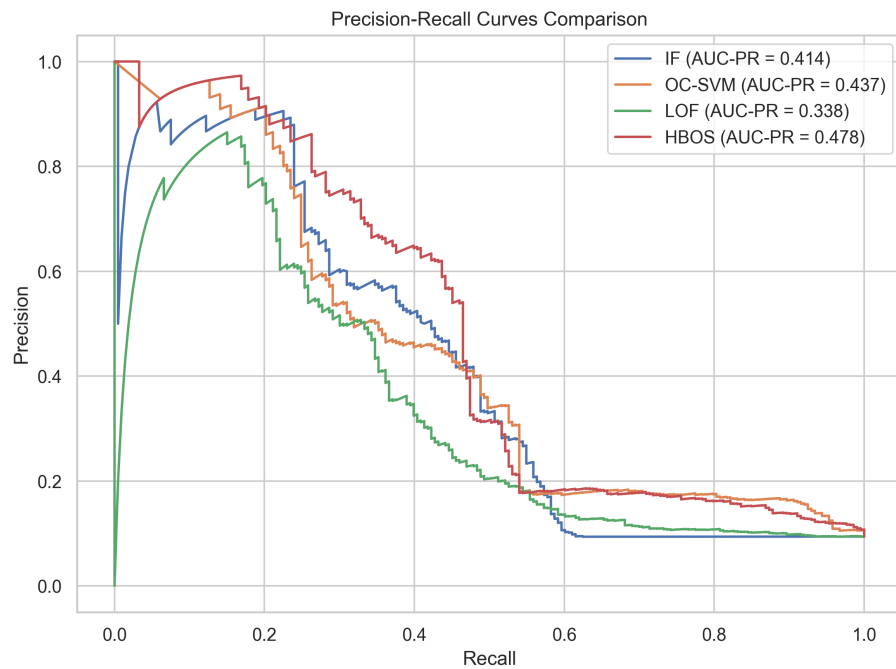


Figure 5. Precision-recall curves: Fraud dataset.

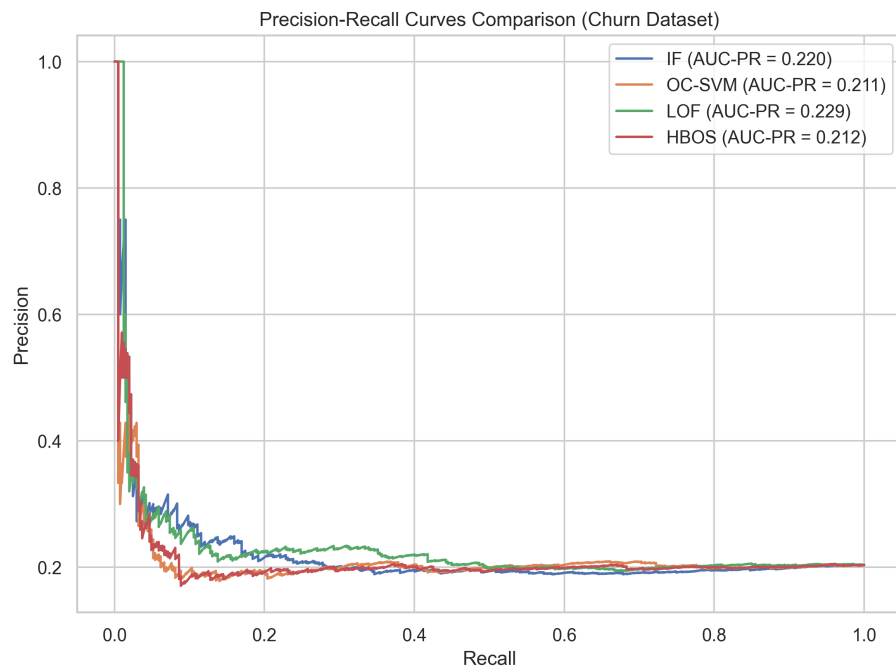


Figure 6. Precision-recall curves: Churn dataset.

Next, we display precision-recall (PR) curves, which show how well a method finds true odd cases (recall) while keeping wrong flags low (precision). The area under the curve (AUC-PR) measures this balance, key for uneven data such as ours where odd cases are few. A score near 0 means bad, near 1 means great; 0.5 is precision–recall random guess. Figure 6 shows PR curves for the churn data. All methods have low AUC-PR around 0.21 to 0.23: IF at 0.220, OC-SVM at 0.211, LOF at 0.229, HBOS

at 0.212. Curves drop fast as recall grows, meaning more odd finds but many false alarms. LOF edges out a bit, matching its top F1 from before. This fits the noisy churn data and high odd rate (20%), where spotting is hard. For fraud (Figure 5), scores are better: HBOS leads at 0.478, OC-SVM at 0.437, IF at 0.414, LOF at 0.338. The HBOS curve stays high longer, meaning a good balance for the low odd rate (9%). This matches high recall in confusion tables, but LOF lags, including its low F1. PR views highlight imbalance issues better than ROC. For fraud, HBOS and OC-SVM shine for real use. For churn, all are weak; one needs better data prep or new methods. The high ROC but low PR shows that recall focus hurts precision in uneven sets.

Radar charts let us see many scores at once for each method, like a spider web. Each arm shows one score: accuracy, precision, recall, F1-score, AUC-ROC, and specificity. A bigger shape means that a method is better all-around; but peaks show strong spots, and dips show weak ones. Scores go from 0 (bad) to 1 (great). We look at both datasets to spot patterns. Figure 7 shows the fraud data. HBOS (red) spikes high on recall (0.999, almost all fraud caught) but dips low on precision (0.104, many wrong flags), accuracy (0.188), and F1 (0.188). Its shape is skinny, good for no-miss but noisy alerts. OC-SVM (orange) has the biggest area, strong on AUC-ROC (0.787) and specificity (0.947), with okay precision (0.322) and recall (0.526). It balances well for fraud spotting. IF (blue) leads accuracy (0.907) and specificity (0.941), but recall (0.418) and AUC (0.648) are middling; making it good for safe calls. LOF (green) is the smallest, with low recall (0.404), precision (0.319), and AUC (0.669), making it the weakest here. For churn (Figure 8), shapes are smaller overall, showing a harder task. HBOS (red) again peaks on recall (0.781) but tanks accuracy (0.319) and precision (0.200). OC-SVM (orange) has fair AUC (0.500) but low recall (0.268). IF (blue) tops accuracy (0.744) and specificity (0.803), with okay precision (0.270). LOF (green) edges F1 (0.269) and recall (0.324), but its AUC is low (0.545). No method fills much space; HBOS is best as a catch-all and IF is best steady results.

Lastly, we explored the learning curves plot which shows how well models do on new data (validation AUC) as we give them more training examples. If the line goes up steadily, the model learns better with more data. Flat means it is good enough or stuck. Down means overfit or bad. We check for both sets to see if data size hurts or helps spotting odd cases. Figure 9 shows fraud curves. The training is augmented from 1000 to 8000 normals. The AUC rises from 0.70 to 0.82. OC-SVM (orange) starts at the top at 0.82 and stays high, best with small data. HBOS (red) climbs quickly to 0.82 and holds, good fast learner. LOF (green) starts okay at 0.78, drops to 0.72 mid-way (maybe noise), then up to 0.75—needs more data to steadily itself. IF (blue) is flat at 0.70, no gain, showing that it is simple but has limits. For churn (Figure 10), the training ranges from 1000 to 6000. The AUC 0.48 to 0.64, near guess at start. LOF performs (green) best: from 0.52 to 0.62, big rise.

3.2. Discussion

The results presented in Section 3.1 provide a comparative perspective on the performance of four widely used unsupervised anomaly detection algorithms: IF, OC-SVM, LOF, and HBOS; across two structurally different datasets: the financial transactions (fraud) dataset and the customer churn dataset. The discussion below interprets these results in relation to the model design, data characteristics, and the broader implications for anomaly detection in financial analytics. Our empirical findings align with, and in several cases nuance, established results in the anomaly-detection literature.

Radar Chart: Multi-Metric Performance Comparison

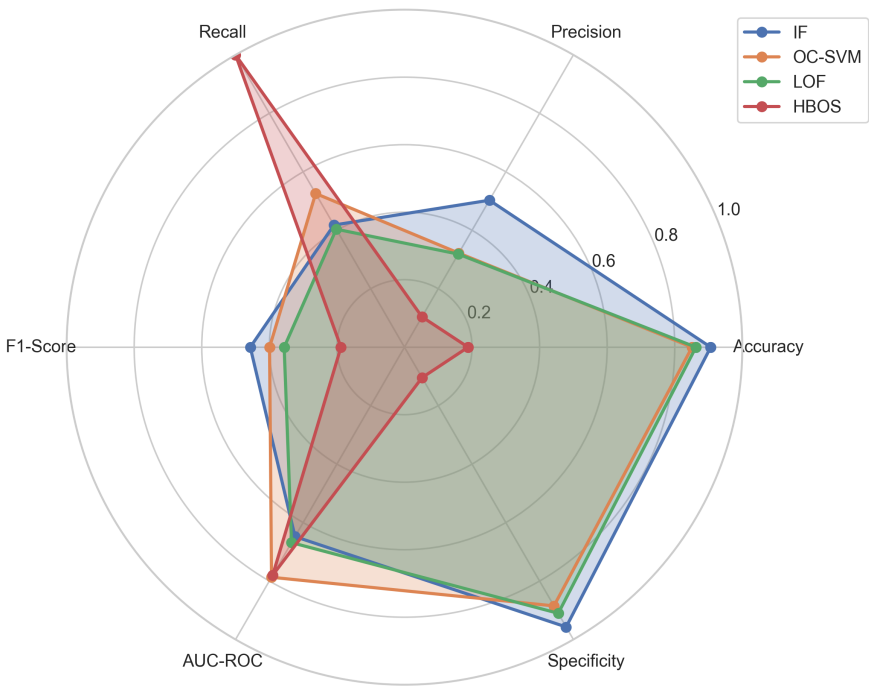


Figure 7. Radar chart comparison: Fraud dataset.

Radar Chart: Multi-Metric Performance Comparison (Churn Dataset)

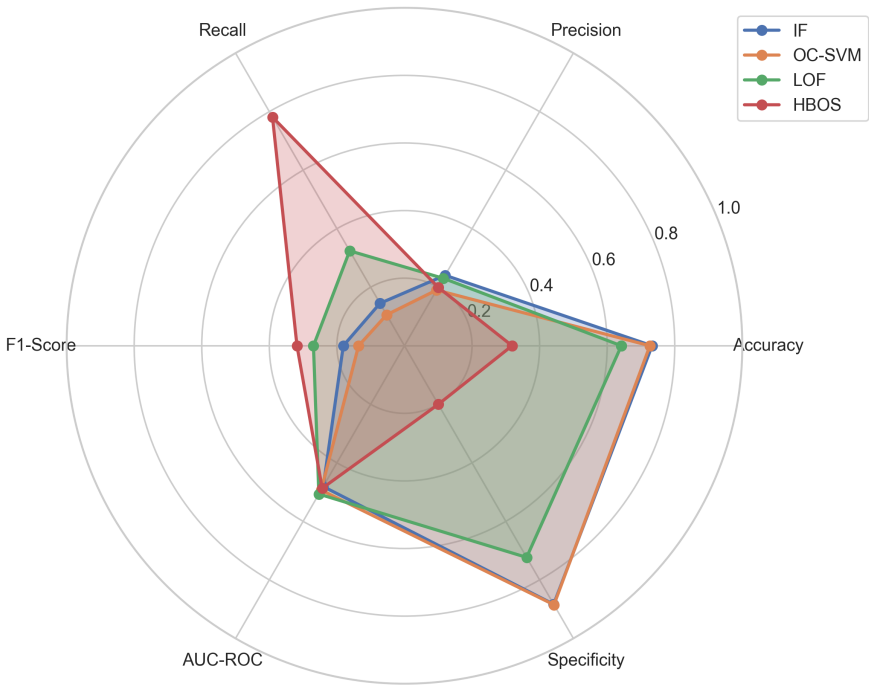


Figure 8. Radar chart comparison: churn dataset.

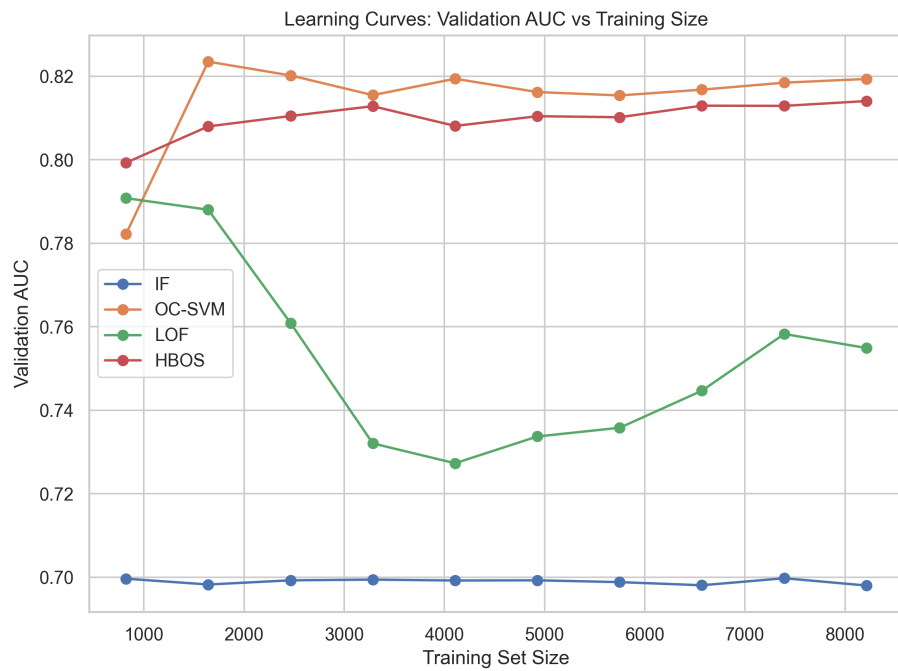


Figure 9. Learning curves comparison: Fraud dataset

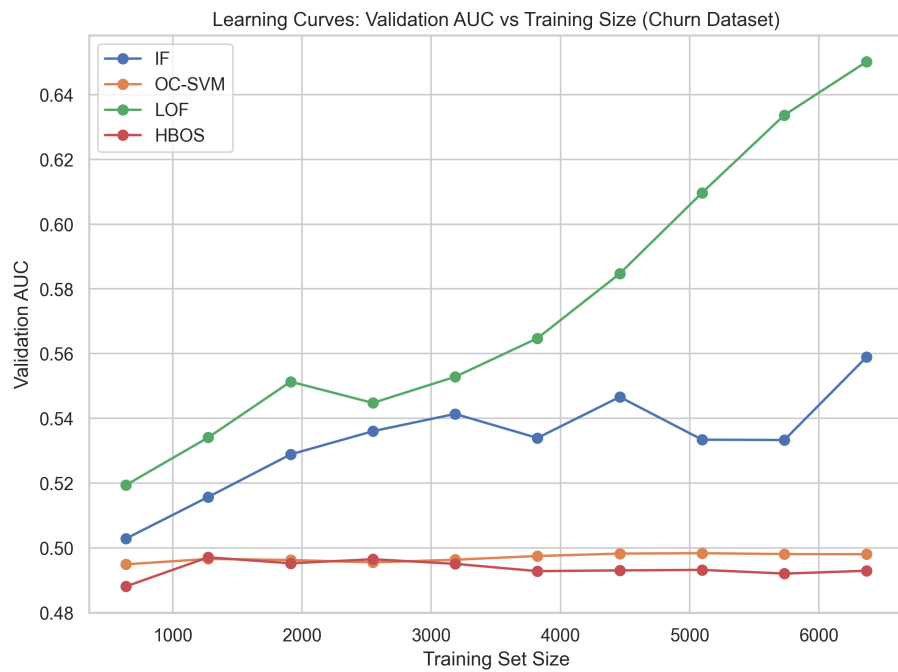


Figure 10. Learning curves comparison: Churn dataset

Overall, the models performed substantially better on the fraud dataset than on the churn dataset. This divergence is largely attributable to differences in data structure and the separability of the underlying classes. The fraud dataset, which comprises clear transactional patterns and continuous numeric variables such as account balances and transaction amounts, presents well-defined distinctions between normal and fraudulent behavior. In contrast, the churn dataset encapsulates behavioral and demographic data that are less linearly separable and inherently noisier, making unsupervised discrimination of anomalies significantly more challenging.

The IF algorithm exhibited strong overall stability and scalability. On the fraud dataset, IF achieved the highest accuracy (90.65%) and a balanced F1-score (0.4564), reflecting its ability to efficiently isolate sparse fraudulent events without excessive false positives. Its lower recall (0.4178) suggests a conservative bias, favoring precision and minimizing false alarms—an acceptable trade-off in operational fraud monitoring, where excessive alerts can overwhelm human investigators. On the churn dataset, however, IF's AUC-ROC dropped to 0.5709, showing that purely random axis-parallel splits struggle in domains with weakly defined anomaly boundaries.

The balanced fraud performance of IF in our study (highest accuracy and competitive F1 but middling recall) is consistent with the original report that IF yields strong AUC and runtime on large tabular data via subsampling and random partitioning (Liu et al., 2008). Two strands of improvements in the literature speak directly to limitations we observed. First, selective/structured tree aggregation such as SA-iForest (Xu et al., 2017) aims to raise accuracy and generalization by retaining diverse, high-utility trees; this could mitigate the recall shortfall we saw on fraud without incurring HBOS-like false positives. Second, partition refinements including GIF (Lesouple et al., 2021) reduce EIF artifacts (e.g., empty branches) and improve efficiency, potentially tightening IF's operating point (higher AUC at similar cost). More substantively, our weak IF/OC-SVM AUC on churn mirrors the axis-parallel partitioning limits that Deep iForest was designed to address (Xu et al., 2023): By embedding data into random nonlinear representations prior to isolation, Deep iForest reports sizable gains on nonlinearly separable tabular, graph, and time-series tasks. Given the behavioral, weak-signal nature of churn, our results suggest that representation-enhanced isolation (Deep iForest) is a promising next step.

OC-SVM demonstrated strong discriminative capacity in the fraud dataset, attaining the highest AUC-ROC (0.7871) and strong recall (0.5258), confirming its effectiveness in capturing nonlinear decision boundaries when the feature space is compact and well-scaled. This performance aligns with its theoretical advantage in kernelized representation learning. However, on the churn data, OC-SVM's performance decreased markedly (AUC-ROC = 0.5001), indicating limited generalization to higher-dimensional or less separable feature distributions. The model's sensitivity to kernel choice and parameter tuning may have contributed to this degradation. On fraud, OC-SVM achieved the highest AUC in our experiments, echoing prior evidence that one-class formulations can outperform two-class SVMs when anomalies are rare and heterogeneous (Hejazi and Singh, 2013). Our churn results (AUC $\approx$ 0.50) indicate kernel sensitivity and decision-boundary fragility in noisier feature spaces, which aligns with the motivation for (i) rank/density-informed large-margin schemes that offer better false-alarm control than basic OC-SVM (Chen et al., 2013), and (ii) distributed/online OC-SVMs that maintain performance while handling streaming, decentralized data (Miao et al., 2018). Practically, these extensions map to two actionable avenues: calibrating decision scores to control false-alarm rates in production, and deploying lightweight online updates for transaction streams.

LOF offered a mixed performance profile. Its neighborhood-based density estimation captured local deviations effectively in both datasets, particularly in the churn dataset, where it achieved the highest F1-score (0.2694) and recall (0.3243) among peers. This suggests LOF's relative robustness in detecting subtle deviations when global class boundaries are weak. However, in the fraud dataset, LOF's lower recall (0.4038) and AUC-ROC (0.6686) reveal limitations in handling highly imbalanced data with sharp global outlier distinctions, where global isolation-based approaches (e.g., IF, HBOS) outperform density-based ones. LOF delivered the best F1 on churn in our study, reflecting its strength at capturing *local* density deviations where global boundaries are weak. This is in line with reports that LOF complements isolation methods: IF is efficient at pruning global outliers, while LOF refines candidates in complex neighborhoods (Paulauskas and Bagdonas, 2015). The two-layer progressive ensemble (IF prescreening, LOF re-scoring) proposed there directly matches our observed trade-offs and is likely to reduce false negatives on churn without exploding the alert volume. For resource-constrained or distributed settings, hierarchical/approximated LOF (Xu et al., 2013) further suggests a viable path to scale neighborhood density checks.

Although HBOS achieved near-perfect recall in the fraud dataset, its low precision implies a high false positive burden. Therefore, HBOS may be better suited as a high-sensitivity screening layer within a multistage detection pipeline rather than as a standalone decision system (0.9989 for fraud and 0.7813 for churn). This reflects its capacity to identify nearly all anomalies, a desirable feature in high-stakes domains such as fraud detection where missing true anomalies can have significant financial implications. Nevertheless, its extremely low precision (0.1038 for fraud and 0.1997 for churn) and F1-scores highlight a tendency to overflag normal observations, generating numerous false positives. This aggressiveness makes HBOS more suitable as a first-line screening mechanism rather than a standalone detection model. HBOS in our results achieved the highest recall on both datasets while sacrificing precision—precisely the speed/recall trade-off reported in its original formulation (Goldstein and Dengel, 2012a). Its linear-time scoring and feature-independence assumption make HBOS attractive as a first-line high-recall screen before costlier confirmatory models. Tuning guidance from network-intrusion studies (Paulauskas and Baskys, 2019) (e.g., binning strategy, per-feature histogram design) is pertinent to reducing the false-positive rate we observed, especially for churn where correlation structure is stronger.

Taken together, the experimental findings and comparisons with prior literature highlight several key contributions of this study to the broader field of unsupervised anomaly detection in financial analytics:

1. **Empirical benchmarking of classical unsupervised models.** This work provides a systematic, large-scale comparison of four foundational anomaly detection algorithms—IF, OC-SVM, LOF, and HBOS—across two distinct real-world domains (fraud and customer churn). By contrasting detection patterns in structured transactional versus behavioral customer data, the study offers one of the few unified evaluations of model generalizability in settings characterized by severe class imbalance.
2. **Quantitative evidence on method–data interaction.** The analysis demonstrates that model effectiveness is highly data-dependent: Isolation-based models excel in structured, low-noise transactional data, whereas density-based models such as LOF perform better in heterogeneous, nonlinear behavioral data. This empirically substantiates theoretical claims in earlier studies that algorithmic suitability depends on the intrinsic data geometry rather than a universal superiority of

any single method.

3. **Operational insights through advanced evaluation.** Unlike most prior studies that rely solely on accuracy or AUC, this research employs a comprehensive set of fifteen performance metrics—including precision, recall, F1-score, balanced accuracy, Matthews correlation coefficient, and Cohen’s kappa—together with ROC, PR, radar, and learning-curve visualizations. This multidimensional evaluation framework offers practitioners a more realistic view of trade-offs between false alarms, detection sensitivity, and computational cost than previous studies.
4. **Integration of dimensionality-reduction for stability.** The study shows that incorporating Principal Component Analysis (PCA) before model training can enhance robustness by mitigating noise, reducing feature redundancy, and improving neighborhood quality—especially for LOF and OC-SVM on high-dimensional churn data. This contribution bridges classical representation learning and anomaly detection, emphasizing preprocessing as a critical determinant of downstream model stability.
5. **Guidelines for practical deployment.** Building on observed trade-offs and insights from the literature, the study proposes a pragmatic deployment framework: (i) use HBOS or IF for rapid high-recall screening; (ii) apply LOF or calibrated OC-SVM for refined detection; and (iii) employ PCA-based preprocessing and rank-score calibration to control false-alarm rates. This hybrid strategy balances speed, interpretability, and precision for industrial applications such as IFRS 9 model monitoring, fraud prevention, and customer behavior analytics.

These directions are consistent with our empirical trade-offs (HBOS recall vs. precision, LOF stability on churn, IF/OC-SVM balance on fraud) and provide a principled roadmap for improving both accuracy and operational cost in real-world anomaly detection.

In regulated domains such as finance and insurance, model explainability is a critical requirement. Although the models evaluated in this study are unsupervised, post-hoc explainability techniques such as SHAP (SHapley Additive Explanations) and local interpretable model-agnostic explanations (LIME) have been extended to anomaly detection settings. For example, feature contribution analysis can be applied to IF and HBOS to identify which variables most influenced anomaly scores.

In this study, we focused on comparative performance benchmarking rather than interpretability analysis. However, integrating explainability tools into unsupervised anomaly detection remains an important direction for future work, particularly for regulatory compliance and operational transparency.

4. Conclusions

This work has explored the application of unsupervised anomaly detection techniques to two critical real-world problems: credit card fraud detection and customer churn prediction. By benchmarking four established algorithms; IF, OC-SVM, LOF, and HBOS on carefully subsampled datasets from Kaggle, we have illuminated the strengths, trade-offs, and contextual dependencies of these methods in imbalanced settings. The empirical analysis, supported by comprehensive metrics and visualizations, reveals that while no single algorithm dominates universally, their performances are profoundly shaped by data characteristics such as contamination rates, feature noise, and separability. In the following sections, we synthesize the key findings, reiterate the study’s contributions, acknowledge its limitations, and outline avenues for future research.

The experimental results show a clear dichotomy in model efficacy between the fraud and churn datasets. On the fraud dataset, characterized by low contamination (9.41%) and structured transactional features, the methods showed an improved discriminative power. OC-SVM and HBOS led in AUC-ROC (0.7871 and 0.7803, respectively), with HBOS achieving near-perfect recall (0.9989) to minimize missed frauds—a vital attribute in high-stakes financial environments. IF provided the most balanced profile, with superior accuracy (0.9065) and F1-score (0.4564), making it suitable for scenarios where false positives must be controlled. LOF, while effective in local anomaly capture, lagged with an F1-score of 0.3561, highlighting its challenges in globally imbalanced data. In contrast, the churn dataset proved more recalcitrant, with higher contamination (20.37%) and noisier, categorical-heavy features leading to universally subdued performance. AUC-ROC values hovered near random guessing (0.5001 to 0.5709), reflecting the inherent difficulty in delineating subtle behavioral deviations without explicit labels. IF again excelled in accuracy (0.7435) and AUC-ROC (0.5709), prioritizing specificity to avoid overflagging loyal customers. LOF emerged as the top performer in F1-score (0.2694) and recall (0.3243), underscoring its value for local density-based detection in heterogeneous data. HBOS retained its recall supremacy (0.7813) but at the cost of dismal precision (0.1997), rendering it a high-sensitivity screener rather than a precise classifier. Visualizations further enriched these insights: ROC and PR curves illustrated steep trade-offs in fraud (e.g., HBOS's sustained precision at high recall) versus near-overlapping baselines in churn; radar charts exposed HBOS's recall peaks amid precision valleys and IF's broad stability; and learning curves demonstrated rapid convergence in fraud (AUC plateaus at 0.82 with modest samples) contrasted with LOF's data-hungry ascent in churn (up to 0.62). Collectively, these findings affirm that unsupervised anomaly detection thrives in low-noise, low-contamination regimes but falters amid behavioral complexity, emphasizing the need for tailored preprocessing and thresholding strategies. The findings are dataset-specific and may not generalize to all anomaly detection scenarios. Performance depends strongly on data separability, contamination rate, and feature structure. Therefore, model selection should be guided by domain-specific validation rather than universal assumptions.

This research advances the field of unsupervised anomaly detection by providing empirical rigor and practical guidance for imbalanced tabular data applications. The key contributions are as follows:

1. **Comprehensive benchmarking of unsupervised methods:** A systematic side-by-side evaluation of IF, OC-SVM, LOF, and HBOS on real-world datasets, uncovering dataset-specific nuances—such as HBOS's recall dominance (0.9989) in fraud versus LOF's F1 superiority (0.2694) in churn—via multimetric assessments (precision, recall, F1, AUC-ROC, AUC-PR) and integrated visualizations (confusion matrices, ROC/PR curves, radar charts, learning curves). This unified framework addresses gaps in prior fragmented studies, offering a holistic view absent in isolated evaluations.
2. **Imbalance simulation and thresholding insights:** Innovative subsampling to realistic contamination levels (9.41% fraud, 20.37% churn) quantifies degradation effects—e.g., AUC-ROC drops from 0.7871 (fraud) to 0.5709 (churn) for top methods—while validation-driven thresholding (e.g., OC-SVM at 0.8181 for fraud) yields actionable guidelines for operational recall-precision balance in resource-limited settings.
3. **Visual and scalability diagnostics:** The deployment of explanatory tools like radar charts (revealing HBOS's “skinny” recall focus) and learning curves (e.g., fraud's quick 0.82 plateau

versus churn's 0.62 climb for LOF) demystifies performance landscapes, demonstrating data efficiency and aiding practitioners in diagnosing scalability bottlenecks without exhaustive hyperparameter sweeps.

4. **Domain-tailored deployment recommendations:** Evidence-based strategies for high-stakes domains, including HBOS/IF for fraud screening (high recall, low cost), LOF for churn refinement (local sensitivity), and hybrid pipelines (e.g., IF pre-filtering + OC-SVM scoring) to mitigate noise, promoting interpretable, cost-effective AI integration in finance and telecommunications.

These contributions not only validate theoretical expectations, e.g., HBOS's univariate speed versus LOF's density nuance—but also extend them through practical, cross-domain synthesis, fostering more resilient anomaly detection pipelines.

5. Limitations

Despite its insights, this study is not without constraints, which merit candid acknowledgment to contextualize the findings and guide interpretations. First, the reliance on static, publicly available Kaggle datasets, while enabling reproducibility, limits generalizability to dynamic, proprietary streams. The fraud data, spanning only two days in 2013, may not capture evolving adversarial tactics, and the churn dataset's synthetic-like demographics overlook real-time behavioral signals (e.g., app interactions). Future validations on longitudinal or industry-specific corpora would enhance external validity. Second, the unsupervised paradigm assumes anomaly-free training data, an idealization that falters in partially labeled or contaminated regimes common in practice. Our normal-class-only training mitigated this but could not fully emulate semi-supervised scenarios with weak labels, potentially inflating optimistic baselines. Third, methodological choices introduced trade-offs: fixed hyperparameters (e.g., LOF's neighbor count) and no advanced preprocessing (e.g., PCA or autoencoders) streamlined comparisons but may undervalue tunable extensions. The 80:20 stratified split preserved imbalance yet constrained small-sample extrapolations, as seen in learning curves' truncation at 8,000 instances. Finally, evaluation emphasized binary metrics, sidelining cost-sensitive analyses (e.g., fraud's asymmetric penalties) or interpretability probes (e.g., SHAP values for feature attribution). These omissions, although scope-appropriate, highlight opportunities for richer, domain-aligned assessments.

6. Future work

The limitations delineated above, coupled with emergent patterns, illuminate fertile directions for extension. Foremost, integrating representation learning—such as PCA or variational autoencoders—prior to detection could attenuate churn's noise, as hinted by LOF's data-responsive gains; empirical trials on augmented datasets would quantify uplift. Hybrid ensembles warrant exploration: Cascading HBOS's high-recall filter with IF's precision or LOF's locality could yield Pareto-optimal F1-scores, particularly in streaming contexts via online OC-SVM variants. Semisupervised wrappers, leveraging sparse labels for threshold calibration, promise to bridge unsupervised purity with supervised accuracy. Broader dataset diversification—encompassing time-series fraud (e.g., transaction sequences) or multimodal churn (e.g., text sentiments) would test robustness across modalities. Cost-aware metrics, incorporating domain penalties (e.g., fraud losses at

\$10 billion annually (Federal Trade Commission, 2024)), could refine recommendations for IFRS-compliant monitoring. Finally, deployment-focused simulations, including A/B testing on synthetic drifts or adversarial perturbations, would validate guidelines in production-like volatility. By pursuing these trajectories, subsequent research can evolve our foundational benchmarking into adaptive, enterprise-grade anomaly detection frameworks, ultimately safeguarding against the escalating threats of fraud and attrition in digital economies.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Author contributions

Lindani Dube: gave conceptual guidance. Tshilidzi Mabushe: analysis and the write up.

Conflict of interest

All authors declare no conflicts of interest in this paper.

Code availability

This project's Python codes and dataset are available at <https://github.com/45026602/AIMS-PROJECT/blob/main/AIMS%20PROJECT>.

References

- Abdi H, Williams LJ (2010) Principal component analysis. *Wires Comput Stat* 2: 433–459. <https://doi.org/10.1002/wics.101>
- Adnan M, Alarood AAS, Uddin MI, et al. (2022) Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models. *PeerJ Comput Sci* 8: e803. <https://doi.org/10.7717/PEERJ-CS.803>
- Aggarwal CC (2017) *Outlier Analysis*. Springer, Cham, Switzerland, 2nd edition.
- Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional space. In: *International Conference on Database Theory*, 420–434. Springer. https://doi.org/10.1007/3-540-44503-X_27
- Bishop CM (2006) *Pattern Recognition and Machine Learning*. Springer, New York. Available from: <https://link.springer.com/in/book/9780387310732>.
- Breunig MM, Kriegel HP, Ng RT, et al. (2000) Lof: Identifying density-based local outliers. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93–104. ACM. <https://doi.org/10.1145/335191.33538>

- Brown J, Smith A, Johnson M (2022) Evaluating classification metrics for imbalanced datasets. *J Data Sci* 20.
- CG SG, Sumathi B (2020) Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction. *Int J Adv Comput Sci Appl* 11. <https://doi.org/10.14569/IJACSA.2020.0110920>
- Chen Y, Qian J, Saligrama V (2013) A new one-class svm for anomaly detection. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 3567–3571. <https://doi.org/10.1109/ICASSP.2013.6638322>
- Favuzzi GL (2019) Deep-fake review detection.
- Federal Trade Commission (2024) As nationwide fraud losses top \$10 billion in 2023, ftc steps up efforts to protect the public. Available from: <https://www.ftc.gov/news-events/news/press-releases/2024/02/nationwide-fraud-losses-top-10-billion-2023-ftc-steps-efforts-protect-public>. Press release.
- Goldstein M, Dengel A (2012a) Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track* 1: 59–63.
- Goldstein M, Dengel A (2012b) Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. In: *KI-2012: Advances in Artificial Intelligence*, volume 7526 of *Lecture Notes in Computer Science*, 59–63. Springer.
- Han J, Kamber M, Pei J (2011) *Data Mining: Concepts and Techniques*. Morgan Kaufmann, Burlington, MA, 3rd edition.
- Hastie T, Tibshirani R, Friedman J (2017) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 12th printing edition.
- Hejazi M, Singh YP (2013) One-class support vector machines approach to anomaly detection. *Appl Artif Intell* 27: 351–366. <https://doi.org/10.1080/08839514.2013.785791>
- Lesouple J, Baudoin C, Spigai M, et al. (2021) Generalized isolation forest for anomaly detection. *Pattern Recogn Lett* 149: 109–119. <https://doi.org/10.1016/j.patrec.2021.05.022>
- Li KL, Huang HK, Tian SF, et al. (2003) Improving one-class svm for anomaly detection. In: *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693)*, 5: 3077–3081. <https://doi.org/10.1109/ICMLC.2003.1260106>
- Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: *2008 Eighth IEEE International Conference on Data Mining*, 413–422. <https://doi.org/10.1109/ICDM.2008.17>
- Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. *Adv Neural Inf Process Syst* 30.
- Miao X, Liu Y, Zhao H, et al. (2018) Distributed online one-class support vector machine for anomaly detection over networks. *Ieee T Cybernetics* 49: 1475–1488.

- Müller KR, Mika S, Rätsch G, et al. (2001) An introduction to kernel-based learning algorithms. In: *Ieee T Neural Networks*, 12: 181–201. <https://doi.org/10.1109/72.914517>
- Pareek A, Arora P, Madan S, et al. (2024) Telecom customer churn prediction model: Analysis of machine learning techniques for churn prediction and factor identification in the telecom sector. *J Inf Optim Sci* 45: 613–630. <https://doi.org/10.47974/JIOS-1599>
- Paulauskas N, Bagdonas AF (2015) Local outlier factor use for the network flow anomaly detection. *Secur Commun Networks* 8: 4203–4212. <https://doi.org/10.1002/sec.1335>
- Paulauskas N, Baskys A (2019) Application of histogram-based outlier scores to detect computer network anomalies. *Electronics* 8: 1251. <https://doi.org/10.3390/electronics8111251>
- Pedregosa F, Varoquaux G, Gramfort A, et al. (2011) Scikit-learn: Machine learning in python. *J Mach Learn Res* 12: 2825–2830.
- Ringnér M (2008) What is principal component analysis? *Nat Biotechnol* 26: 303–304. <https://doi.org/10.1038/nbt0308-303>
- Saito T, Rehmsmeier M (2015) The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* 10: e0118432.
- Schölkopf B, Smola AJ (2002) *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.
- Su J, Jiang C, Jin X, et al. (2024) Large language models for forecasting and anomaly detection: A systematic literature review. *arXiv preprint arXiv:2402.10350*.
- Xu D, Wang Y, Meng Y, et al. (2017) An improved data anomaly detection method based on isolation forest. In: *2017 10th International Symposium on Computational Intelligence and Design (ISCID)* 2: 287–291. <https://doi.org/10.1109/ISCID.2017.202>
- Xu H, Pang G, Wang Y, et al. (2023) Deep isolation forest for anomaly detection. *Ieee T Knowl Data Eng* 35: 12591–12604. <https://doi.org/10.1109/TKDE.2023.3270293>
- Xu L, Yeh YR, Lee YJ, et al. (2013) A hierarchical framework using approximated local outlier factor for efficient anomaly detection. *Procedia Comput Sci* 19: 1174–1181. <https://doi.org/10.1016/j.procs.2013.06.168>



AIMS Press

©2026 Mabushe & Dube, licensee AIMS Press.
This is an open access article distributed under the
terms of the Creative Commons Attribution License
(<https://creativecommons.org/licenses/by/4.0>)