Anomaly Detection in Financial Data Using Isolation Forest

***Abstract*—This survey delves into the realm of anomaly detection within financial data, specifically exploring the Isolation Forest algorithm's application in this context. The Isolation Forest algorithm proves to be a robust tool, exhibiting efficiency in isolating anomalies, particularly within extensive and multidimensional datasets. Through rigorous empirical studies and comprehensive comparisons with conventional methodologies, this survey showcases the algorithm's considerable potential to fortify the security and integrity of financial systems. By adeptly identifying irregularities like fraud and market anomalies, the Isolation Forest algorithm emerges as a pivotal component in bolstering trust and stability within financial markets.**

***Index Terms*—outlier detection, anomaly detection, financial data, market anomalies, data integrity**

I. Introduction

The advent of digital technology has revolutionized the financial industry, leading to an explosion in the volume and variety of financial data. This data, while a valuable resource, also presents new challenges. One such challenge is the detection of anomalies, which are unusual patterns in the data that deviate from what is expected or normal. Anomalies can be indicative of significant events such as fraudulent activity, market manipulation, or sudden changes in price or volume. The early detection of these anomalies is crucial for financial institutions and investors to mitigate risks and protect their assets.

In the realm of financial data, anomaly detection is not just a challenging task but also a critical one. The stakes are high, as the implications of undetected anomalies can range from financial losses for businesses to impacts on the economy. Therefore, the focus of this short study is to evaluate few well-known number of mainstream anomaly identification techniques specifically on financial data.

The logic behind data acquisition plays a pivotal role in interpreting the results of these techniques. However, in certain cases, such as banking transaction data, confidentiality requirements necessitate the transformation of the original dataset into a different form. In these instances, deriving conclusions about why a specific data object is anomalous becomes challenging.

The significance of detecting outliers or anomalies extends beyond the financial sector and is a topic of interest in data mining and research. These outliers or anomalies, terms often used interchangeably, represent data points that significantly deviate from expected patterns. Their identification can be challenging due to their infrequency, but they are of considerable interest to data analysts. Further elaborated, outliers often result from data acquisition errors and resemble noise, while anomalies significantly deviate from the norm. Despite the occasional distinctions made in the literature, both terms are commonly treated as equivalent in practice. The detection of these anomalies has applications in various fields, such as fraud detection in finance, public health anomaly detection, and more.

Outlier detection methods can enhance model accuracy and reduce computational complexity. Outliers can be categorized as global, deviating significantly from the entire dataset, or local, deviating from specific neighborhoods. The identification of local outliers poses a greater challenge, further underscoring the importance of sophisticated detection techniques.

In conclusion, this study aims to provide a comprehensive review of anomaly detection techniques with a especial focus on financial datasets for evaluation purposes, acknowledging the challenges posed by confidentiality requirements and the distinction between global and local outliers. By doing so, it hopes to contribute to the ongoing efforts to safeguard financial institutions and investors from the potential risks posed by undetected anomalies.

II. Related work

Outlier detection techniques can typically be classified into eight main categories [1]–[4]: Statistical and probabilistic models; proximity models including methods based on distance, density-dependent techniques, and methods that rely on clustering; graph-centric methods; methods rooted in information theory; and isolation-centric methods. Following the examination of these categories, we will engage in a discussion to evaluate the capacity of these respective methods to handle data on different scales.

A. Statistical and probabilistic methods

The earliest outlier detection techniques, dating back to the 19th century, are based on statistical and probabilistic models. These methods were developed before the rise of computer technology, so they didn't emphasize practical aspects like data representation or computational efficiency. However, their foundational mathematical models have proven valuable and have been adapted for various computational contexts. A common approach in statistical modeling for outlier analysis involves identifying extreme univariate values [5].

Statistical methods: Extreme Value Analysis (EVA) is a statistical method used for outlier detection. This technique is based on the assumption that the data follows a specific probability density distribution. The outliers are then identified as the data points that reside at the extreme ends of this distribution. In the context of a normal distribution, for instance, most data points cluster around the mean value, and the frequency of the values decreases as we move away from the mean towards either end of the distribution. The outliers, in this case, would be the values that lie in the tails of the distribution, significantly distant from the mean.

EVA is particularly useful in detecting global outliers, which are data points that are significantly deviant from the rest of the data set. These outliers are not just anomalous within a certain subset or cluster of data, but they stand out within the entire data set. This makes EVA a powerful tool in fields like anomaly detection, fraud detection, and robust statistical estimation, where identifying these global outliers is of critical importance. However, it's important to note that the effectiveness of EVA is heavily dependent on the assumption that the data follows a specific distribution. If this assumption does not hold, the method may not be able to accurately identify outliers. Therefore, a thorough exploratory data analysis is often necessary before applying EVA to understand the underlying distribution of the data [1], [2].

The authors in [6] introduce COPOD, a novel outlier detection algorithm inspired by copulas, addressing issues of computational complexity, predictability, and interpretability in existing methods. COPOD constructs an empirical copula to assess the "extremeness" of each data point, akin to calculating an anomalous p-value, offering parameter-free, interpretative, and efficient outlier detection. The contributions include proposing COPOD as a high-performance and interpretable algorithm, extensive experiments demonstrating its superiority on 30 benchmark datasets, and providing an easy-to-use Python implementation for reproducibility.

Probabilistic methods: In the realm of probabilistic methodologies, the underlying assumption posits that data emanates from a composite of diverse distributions, conceptualized as a generative model. Subsequently, the same dataset is employed to iteratively estimate the parameters integral to the model. Once these specified parameters are delineated, objects designated as outliers are identified based on their diminished likelihood of conforming to the generated model [2]. Schölkopf et al. [7] propose an alternative approach rooted in supervision, wherein a probabilistic model is formulated concerning the input data, tailored to best encapsulate normal data patterns. The primary objective of this method is to discern the most confined region encompassing a majority of normal objects; any data points lying outside this delineated region are deemed outliers. Essentially, this method functions as an extended iteration of Support Vector Machines (SVM), specifically enhanced to grapple with imbalanced data. Termed as one-class SVM or OCSVM in the literature, it effectively treats a limited number of outliers as part of a rare class, contrasting with the majority of the data designated as normal objects.

Current unsupervised methods often grapple with issues such as high computational cost, intricate hyperparameter tuning, and limited interpretability, particularly when dealing with large, high-dimensional datasets. To tackle these challenges, in [8] a straightforward yet potent algorithm titled ECOD (Empirical-Cumulative-distribution-based Outlier Detection) is introduced. ECOD, inspired by the notion that outliers are typically the ”rare events” found in the tails of a distribution, initially estimates the input data’s underlying distribution nonparametrically by calculating the empirical cumulative distribution for each data dimension. It then uses these empirical distributions to estimate each data point’s tail probabilities per dimension. Ultimately, ECOD computes an outlier score for each data point by aggregating the estimated tail probabilities across dimensions. The key contributions in this work include the proposal of a novel, parameter-free, and easily interpretable outlier detection method, and the extensive testing of this technique on various benchmark datasets, where it outperformed many other cutting-edge baselines in accuracy, efficiency, and scalability.

The work in [9] presents ABOD (Angle-Based Outlier Detection), a novel approach for outlier detection in large datasets. ABOD and its variants assess the variance in angles between difference vectors, alleviating the challenges posed by the ”curse of dimensionality” in high-dimensional data. Unlike typical distance-based methods, ABOD does not require parameter selection, providing a significant advantage.

The authors in [10] examine a wrap-around version of the L2-discrepancy (WD) to compare random designs and Latin hypercube designs. Theoretical analysis reveals that Latin hypercube designs consistently exhibit lower expectations and variances compared to random designs. The study also explores the construction of uniform designs under WD, demonstrating the flexibility of one-dimensional uniform design. For high-dimensional uniform designs, a threshold-accepting heuristic is applied to discover low-discrepancy designs, confirming the conjecture proposed in certain design conditions.

B. Proximity-based models

Distance-based techniques: Distance-based outlier detection algorithms are methods that identify data points that deviate significantly from the rest of the data based on some distance measure. These methods typically use the nearest-neighbor relationships between data points to determine the outlier scores, which can be global or local depending on the reference set used for comparison [11]. Distance-based outlier detection algorithms can be applied to various types of data, such as tabular, image, or network data, and can use different distance metrics, such as Euclidean, cosine, or Hamming distance.

In [12], a robust and efficient method for distance-based outlier detection is introduced. A multitude of outlier detection techniques have been introduced to date, with k-nearest neighbor (kNN) based methods being among the most notable. A significant challenge with k-nearest neighbor-based methods is their substantial reliance on the parameter k. However, the method proposed in this study mitigates the sensitivity to k while preserving the algorithm’s high precision.

In the context of statistics, [13] introduces the Multi-Granularity Deviation Factor (MDEF), a tool for detecting outliers in data. It identifies outliers when MDEF values deviate significantly from local averages. Several algorithms are proposed, including LOCI and aLOCI, which are efficient and deliver high-quality results. This method also provides a detailed LOCI plot for each point, offering more information than the typical outlierness score. This characteristic of the method allows users to better understand detected outliers. Sugiyama and Borgwardt [14] introduce a quick distance-based technique that uses the nearest neighbor distance from a small random sample of the dataset. It assigns an outlier score to each point based on its distance to its nearest neighbor in the sample. With linear time complexity and constant space complexity, it's well-suited for analyzing large datasets.

Bay and Schwabacher [15] propose ORCA which is an optimized algorithm using k Nearest Neighbors (kNN) distances. It processes data in blocks, updates a cut-off value for anomaly scores, and prunes data points below the cut-off. Its worst-case time complexity is quadratic. Angiulli and Fassetti [16] introduce DOLPHIN which is a distance-based outlier mining method for disk-resident data. It combines object selection, pruning strategies, and similarity inspection approaches for efficiency, and can be applied to both metric and non-metric data.

The work in [14] is a swift distance-based technique that relies on nearest neighbor distances computed from a small dataset sample. This method extracts a random subset, assigning an outlierness score to each point based on its distance to the nearest neighbor in the sample. With linear time complexity in relation to critical variables (object count, dimensions, and samples), and constant space complexity, this technique proves effective for analyzing large datasets.

Density-based techniques: Density-based methods calculate the local density of each object in a unique way to determine outlier scores. The lower an object's local density compared to its neighbors, the more likely it is to be an outlier. Many techniques, mostly distance-based, can calculate density around points. For instance, Breunig et al. [17] proposed the Local Outlier Factor (LOF) that uses the distance values of each object to its nearest neighbors to compute local densities. However, LOF's limitation is that the scores are not globally comparable across all objects in the same dataset or different datasets. An enhanced version of LOF, the Local Outlier Probability (LoOP), was introduced by the authors of [18]. LoOP assigns each object a score within the range [0,1], representing the probability of the object being an outlier, and this score is broadly interpretable in various contexts.

In light of the limitations of current outlier detection methods and the need for rapid processing of high-dimensional data sets, a novel unsupervised local outlier detection technique is introduced in [19]. This method leverages the Chebyshev inequality to define the neighborhood boundaries of data points. It establishes these boundaries using a deviation parameter related to the standard deviation of the data distribution, and identifies outliers by measuring their neighborhood densities.

The technique introduced in [20] is an innovative density estimation technique for anomaly detection, employing density matrices (a potent mathematical construct from quantum mechanics) and Fourier features. This method serves as an effective approximation of Kernel Density Estimation (KDE). The method is efficiently trained, utilizing optimization to determine data embedding parameters. The algorithm's prediction phase complexity remains constant relative to the size of the training data and exhibits strong performance across data sets with varying anomaly rates. Its design supports vectorization and is compatible with GPU/TPU hardware implementation as well.

Clustering-based techniques: Clustering-based approaches for outlier or anomaly detection are based on the premise that normal data objects belong to large and dense clusters, while outliers belong to small or sparse clusters, or do not belong to any clusters. These methods detect outliers by examining the relationship between objects and clusters [21].

Jobe and Pokojovy [22] introduce a data-intensive approach based on cluster analysis, using a reweighted version of Rousseeuw's minimum covariance determinant method [23]. Despite the reduced detection capability with increasing outliers, a multi-stage algorithm helps distinguish potential outliers from true inliers. Huang et al. [24] present ROCF, an outlier cluster detection method that doesn't require a count of top-N outliers. ROCF builds a neighborhood graph, MUNG, using the mutual neighbors concept, and identifies both singular outliers and small anomalous clusters by estimating the outlier ratio of the dataset.

The work in [25] introduces a novel outlier detection method that calculates the outlierness degree for each instance in a dataset. The model uses a mass-based dissimilarity measure to overcome the limitations of neighbor-based models and detect local outliers across various data point densities. It employs a hierarchical partitioning technique to create a nested structure for the dataset, defines a mass-based dissimilarity measure to quantify dissimilarity between two instances, and gathers neighbors with the k lowest mass dissimilarities to form a context set for each instance. A mass-based local outlier score model is then used to compute the outlierness for each instance. This method revolutionizes the outlier model perspective by using mass-based measurements instead of the distance-based functions commonly used in neighbor-based methods.

C. Graph-centric models

Graph-based outlier detection methods are crucial in machine learning, with applications such as identifying social network spammers and detecting sensor faults. However, these methods face challenges due to the complex data structure of graphs, the difficulty of training more complex models, increased memory consumption, and the lack of a comprehensive benchmark for performance evaluation. To address these issues, researchers in [26] have introduced the Unsupervised Node Outlier Detection (UNOD) benchmark. UNOD evaluates fourteen methods and benchmarks their performance on real-world datasets. It also compares algorithm efficiency and scalability. This benchmark aims to advance the field of graph outlier detection and guide future research.

The study in [27] introduces a graph-based method that enables unsupervised outlier detection methods to consider a few labeled data. Despite the challenges of obtaining labeled data, particularly for outliers, there are opportunities to obtain a few labeled data, such as feedback from a human analyst reviewing the results of an unsupervised method. However, widely used unsupervised methods for outlier detection often fail to properly consider or use the labeled data. To address this, the proposed method first endows the unsupervised method with the ability to consider a few labeled data. It then extends this semi-supervised method to active outlier detection by incorporating a query strategy that selects top-ranked outliers.

D. Information-theoretic models

Information-theoretic methods are generally considered on par with distance-based and other deviation-based models. The unique aspect of these methods is that they first define a specific concept of deviation, and then anomaly scores are calculated by assessing the model size for this particular type of deviation. This differs from typical approaches where a fixed model is initially established, and anomaly scores are derived by measuring the magnitude of deviation from this model.

In [28] a novel approach is introduced for the purpose of event log anomaly detection in process event streams. It defines a general framework where various anomaly detection methods can be incorporated, and proposes a method based on statistical leverage. This measure, widely used in statistics to identify outliers, is adapted to the specific scenario of event streams. The approach is evaluated on artificial and real event streams, including those characterized by concept drift, demonstrating its potential for early anomaly detection and adaptability to changing processes.

The study in [29] revisits a direct objective function for anomaly detection with information theory, addressing the limitations of surrogate task-based methods in unsupervised image anomaly detection. The proposed framework maximizes the distance between normal and anomalous data in terms of the joint distribution of images and their representation. It finds a lower bound for this objective function that balances mutual information and entropy, leading to a novel information theoretic framework for unsupervised image anomaly detection. Extensive experiments demonstrate that this framework significantly outperforms several state-of-the-art methods.

E. Isolation-centric methods

Isolation-based methods, first introduced in [30], [31] and exemplified as the inspiring method entitled Isolation Forest (iForest), draw inspiration from the well-known ensemble method, Random Forests [32]. The approach involves constructing an ensemble of isolation trees (iTrees) for the input dataset, with outliers identified as objects with a short average path length on the corresponding trees. Each iTree is created by recursively partitioning a random sub-sample of the entire data, based on a randomly chosen attribute and a split value selected from the range of minimum and maximum values of that attribute. Outliers, which deviate significantly from normal data behavior, are located on tree branches with notably lower depth.

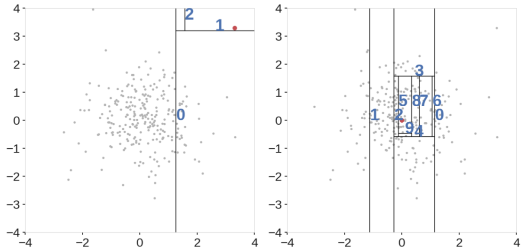
However, as the number of dimensions increases, an incorrect attribute choice for splitting at higher levels of the iTree can potentially mislead detection outcomes. Despite this, iForest's advantage lies in its ability to use sub-sampling more sophisticatedly than existing methods, offering an algorithm with low linear time complexity and memory requirement. An enhanced version of iForest, named iNNE, is proposed in [33], [34] to address some limitations of the primary method, including the inability to efficiently detect local anomalies, anomalies with a low number of relevant attributes, global anomalies disguised through axis-parallel projections, and anomalies in a dataset containing various modals.

III. Elaboration on Isolation Forest

The Isolation Forest algorithm stands out as a formidable technique for identifying outliers in diverse datasets, particularly excelling in large and high-dimensional financial datasets. Leveraging binary trees, this algorithm ensures linear time complexity and minimal memory usage, rendering it exceptionally suitable for processing vast datasets efficiently.

A. Methodology

Isolation Forest, analogous to Random Forests, is constructed based on an ensemble of binary decision trees each known as an Isolation Tree (iTree), constituting an unsupervised model devoid of predefined labels. In this approach, randomly sub-sampled data undergoes a tree structure process based on randomly selected features. The depth to which samples traverse the tree signifies their likelihood of being anomalies. The amalgamation of multiple Isolation Trees forms the Isolation Forest, culminating in the completion of the model training phase.



(a) Anomaly point (b) Nominal point

Fig. 1: (a) Isolating an anomalous point; (b) Isolating a normal point [35].

Here's a step-by-step breakdown of the algorithm:

1) A random sub-sample of the dataset is selected and assigned to a binary tree.
2) The tree's branching begins by first selecting a random feature from all N features. Branching then occurs based on a random threshold, which is any value within the range of the selected feature's minimum and maximum values.
3) If a data point's value is less than the selected threshold, it's directed to the left branch; otherwise, it goes to the right. This results in a node being split into left and right branches.
4) The process from step 2 is repeated recursively until each data point is completely isolated or until a defined maximum depth is reached.
5) Steps 1-4 are repeated to construct random binary trees, forming an ensemble of iTrees, or an Isolation Forest.

Once the Isolation Forest is generated, the model training is complete. However, as in Fig. 2 representing an iTree, it is not mandatory for such binary trees to be complete; i.e., the isolation tree building procedure may stop at any arbitrary branch due to the data sample inherent structure and the present randomness in the process as well.

Furthermore, during scoring, each data point is traversed through all the previously trained trees. An "anomaly score" is then assigned to each data point, based on the tree depth required to reach

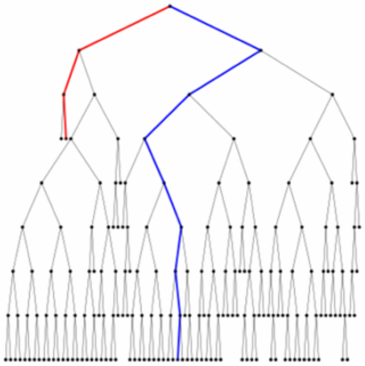


Fig. 2: Illustration of a single iTree in an iForest [35]. The red path is related to an anomaly while the blue path with a larger length is pertaining to an inlier.

that point. This score is an aggregate of the depths obtained from each iTree. Anomalies are assigned an anomaly score of -1, and normal points are assigned 1, based on the contamination parameter, which represents the percentage of anomalies in the data.

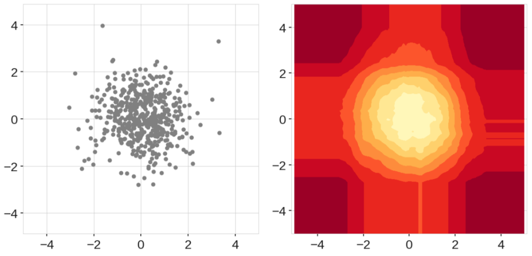
Fig. 1 illustrates an example of how isolating procedure happens for an anomaly and a normal data element. Taking the account of the red query point whether anomalous/outlier or normal/nominal/inlier, every axis-parallel (horizontal/vertical) line represents a randomly chosen value in the respective attribute. The numbers on the lines show the branching order while building the isolation trees. Finally, it is clear that many more branching or the same cut operations are required to isolate an inlier compared to an outlying point. However, it should be stipulated as well that due to the depth limit over the iTrees, our inlier here is not comprehensively isolated, yet the acquired isolation path length is sufficient to presume it as an inlying point.

B. *Limitations and Extensions*

While Isolation Forest offers computational efficiency and proven efficacy in anomaly detection, they are not without limitations. The final anomaly score's dependence on the contamination parameter implies a prerequisite understanding of the percentage of anomalous data for accurate predictions. Nevertheless, one can expect from data practitioners to come up with a reasonable number for top-N outliers and this cutoff value could be employed to induce candidate outliers out of scored data. The model also exhibits bias due to the branching mechanism, leading to potential inaccuracies in anomaly detection. An extension to Isolation Forest, known as "Extended Isolation Forest" [35] has been introduced to mitigate this bias by replacing horizontal and vertical branch cuts with cuts featuring random slopes.

Further elaborated, let us consider Fig. 3a showing a sample Gaussian cluster with anomaly score map out of iForest illustrated in Fig. 3b. In the Forest, evenly sampled points (apart from the original distribution and just for visualization purposes) are harnessed within the plot's range and the corresponding scores are allocated to these points according to the established iTrees. As a result, one will essentially anticipate an anomaly score map that displays a nearly circular and symmetrical pattern for the normal cluster in Fig. 3, with values escalating as we progress radially outwards.

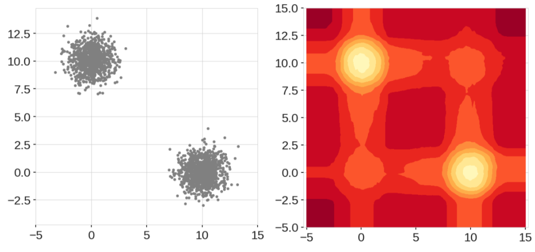
However, even by witnessing this expectation happening to some extent, we also notice areas of lower anomaly scores in the x and y directions, compared to other points at a similar radial distance from the center. Given our knowledge of the data distribution, the score map should ideally maintain a



(a) Normally Distributed Data

(b) Anomaly Score Map

Fig. 3: Axis-parallel cuts issue in iForest that causes many outlying candidates parallel to the dense center of inliers to get lower scores [35].

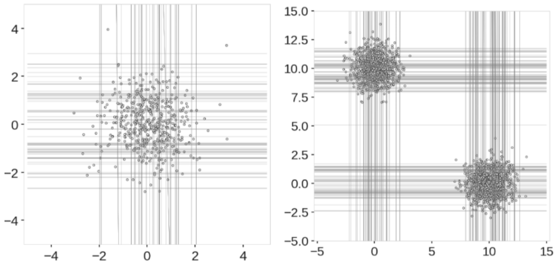


(a) Two normally distributed clusters (b) Anomaly Score Map

Fig. 4: Axis-parallel cuts issue in iForest for a two Gaussian cluster case [35].

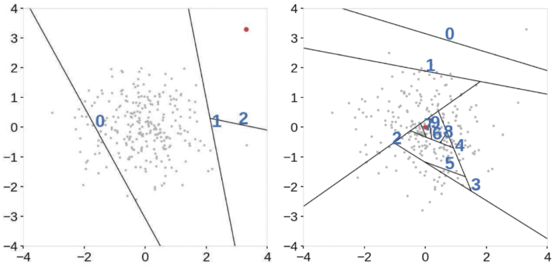
roughly circular shape for all radii, meaning similar score values for fixed distances from the origin. Nevertheless, this anticipation does not completely take place due to the axis-parallel cuts employed in the conventional iForest technique.

Fig. 4 shows a similar case to Fig. 3 but with two normal clusters. It is clear that the model has produced two extra blobs inside the anomaly score map (on the top right and bottom left) that were not originally present in the input data. Such axis-parallel cuts incurring this phenomenon are better illustrated in Fig. 5. By considering evenly sampled points (which are not part of the original distribution) to build the anomaly score map, it becomes more evident why the purely horizontal/vertical cuts are causing such artifacts in the resulting anomaly score map.



(a) Single blob (b) Multiple Blobs

Fig. 5: During the training phase of the standard Isolation Forest, branch cuts are produced. At each step, a random value is chosen from a random feature (dimension). The training data points are then directed to either the left or right branches of the tree, depending on which side of the line they fall on [35].



(a) Anomaly (b) Nominal

Fig. 6: The Extended Isolation Forest's branching process in which for an anomalous data point, which is isolated after just three random cuts. It is depicted for the same process for a nominal point near the data's center, requiring numerous cuts for isolation. However, the tree's depth limit was reached before the point was isolated [35].

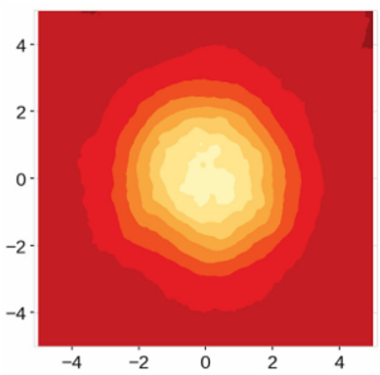


Fig. 7: Anomaly score map out of the Extended Isolation Forest [35] in which there are no artifacts out of the original distribution region.

As mentioned earlier, an enhancement to Isolation Forest, termed "Extended Isolation Forest" [35], has been proposed to alleviate this bias. It does so by substituting horizontal and vertical branch cuts with cuts that have random slopes. Fig. 6 demonstrates such a resolution to the conventional iForest's issue in which cutting lines with random slopes are employed to generate isolation trees. The outcome of such a cutting process results in an anomaly score map for our single normal cluster case shown in Fig. 7 which is appealingly devoid of any artifact like in Fig. 3.

C. Wrap-up

Finally, we can assert why Isolation Forest emerges as a powerful tool for efficiently detecting anomalies in diverse datasets, particularly in our specific cases here, the financial domain. Moreover, while there are multiple potentials for this technique, users must be cognizant of inherent limitations such as sensitivity to data dimensionality, i.e., the presence of irrelevant features, and the potential bias towards smaller clusters. Despite these considerations, Isolation Forest remains widely utilized in various fields and also by very authentic and large companies/institutions like LinkedIn[1] for anomaly detection, exemplifying their enduring relevance and applicability.

IV. Experimental evaluation

In this section, we rigorously assess the effectiveness, precision, and efficiency of several competing methods employed for outlier detection within the realm of synthetic and real-life multivariate financial data. The objective is to provide a fairly comprehensive understanding of the comparative performance of these outlier detection techniques, shedding light on their capabilities and limitations in handling the intricacies inherent in financial datasets.

The employed techniques here are as follows. ABOD: Angle-Based Outlier Detection [9], COPOD: Copula-Based Outlier Detection [6], ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions [8], Isolation Forest [30], LOF [17], QMCD: Quasi-Monte Carlo Discrepancy Quasi-Monte Carlo Discrepancy outlier detection [10], and Rapid D: Rapid distance-based outlier detection via sampling [14]. All of these methods have been reviewed shortly in Section II.

A. Evaluation metrics

The evaluation metrics for anomaly detection we are focusing on here are mostly concerned with imbalanced datasets, including F1-score, Area Under Precision-Recall Curve (AUC-PRC), and Area Under ROC Curve (AUC-ROC). However, the last one is not generally suitable for outlier detection cases as the false alarm rate in it is not influential enough. These metrics are explained in detail in the following:

1) F1-Score: The F1-score is a metric that combines precision and recall into a single value. For outlier detection, precision represents the ability of the model to correctly identify true outliers among the instances predicted as outliers, and recall represents the ability of the model to capture all actual outliers in the dataset.

The F1-score is calculated using the following formula:

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

where Precision is the ratio of true positives to the sum of true positives and false positives, and Recall is the ratio of true positives to the sum of true positives and false negatives.

A higher F1-score indicates a better balance between precision and recall, suggesting a model that performs well in both identifying outliers and avoiding false positives.

2) Area Under Precision-Recall Curve (AUC-PRC): The precision-recall curve is a graphical representation of the trade-off between precision and recall for different threshold values in the model. The Area Under the Precision-Recall Curve (AUC-PRC) provides a summarized measure of the model's performance across various threshold levels.

In the context of outlier detection, AUC-PRC quantifies how well the model distinguishes between normal and outlier instances. A higher AUC-PRC value signifies better discrimination, implying a model that achieves high precision while maintaining high recall across different decision thresholds.

[1] https://engineering.linkedin.com/blog/2019/isolation-forest

TABLE I: The datasets used in our experiments along with the associated details about dimensions, outliers portion, etc.

Dataset	#instances	#features	#outliers	#outliers (%)	from	#instances	#outliers
SFD	6,362,620	29	100,000	10	HuggingFace	500000	233
2018-FinData	3,167	224	121	3.82	Kaggle	All	All
CrCFraud	1,500,000	22	492	0.05	HuggingFace	200000	1645
Defaulter	10000	3	1000	10	Kaggle	9,841	174
campaign	41,188	62	4640	11.27	ADBench	All	All
fraud	284,807	29	492	0.17	ADBench	All	All

3) Area Under ROC Curve (AUC-ROC): The Receiver Operating Characteristic (ROC) curve plots the true positive rate (sensitivity) against the false positive rate for different decision thresholds. The Area Under the ROC Curve (AUC-ROC) summarizes the model's performance in distinguishing between classes.

A higher AUC-ROC value indicates better discrimination, suggesting a model that achieves low false positive rates while maintaining high true positive rates across different decision thresholds.

B. Datasets description

The scarcity of financial datasets for outlier detection tasks is a significant challenge in the field. While there are a few notable datasets available, such as the creditcard dataset containing transactions made by European cardholders, the number of publicly available financial datasets specifically designed for outlier detection is limited or not interpretable if public at all. Researchers often have to rely on a small number of datasets or resort to synthetic data generation techniques to address this scarcity in the field of outlier detection in finance. In total, we have used 7 datasets hand-picked from different sources.

synthetic-fraud-detection (SFD): Synthetic-fraud-detection in banking activities including Payment, Withdrawal, Debit, Transfer, and Cash-out, etc. The dataset as opposed to creditcard dataset has attributes that are not PCA transformed and are more interpretable. With more than 6 million instances.

2018-financial-data (2018-FinData): This dataset collects 200+ financial indicators for all the stocks of the US stock market. The financial indicators have been scraped from Financial Modeling Prep API, and are those found in the 10-K filings that publicly traded companies release yearly. The dataset is originally for stock direction prediction, but we use it for outlier detection with stocks that go up as normal and outliers otherwise. We use the 2018 data and downsample the stocks that went down to represent the outliers.

CIS435-CreditCardFraudDetection (CrCFraud): This dataset contains transactions made by credit cards with detailed attributes including card number, time, amount, etc. The dataset is originally for credit card fraud detection and it has 1+M transactions.

Defaulter: This dataset contains information about the customers of a bank. It is originally intended to decide whether a customer is a defaulter or not (meaning whether the customer will pay back the loan or not). We downsample defaulters and take them to be outliers. The dataset does not have many attributes but important ones are: balance, income, and student.

campaign: campaign is originally a dataset from ADBench (Anomaly Detection Benchmark) [36] and is used for outlier detection. The dataset has 62 attributes and 41188 instances. with 4640 outliers. Although not much information is provided as to what kind of data it is, we consider it a good example of an outlier detection dataset as it is hosted in ADBench, a known benchmark for outlier detection.

fraud: Similar to campaign, fraud is again originally a dataset from ADBench (Anomally Detection Benchmark) [36] and is used for outlier detection. The dataset has 29 attributes and 284807 instances. with 492 outliers.

(a) F1-Score

Dataset	ABOD	COPOD	ECOD	IForest	LOF	QMCD	Rapid D
SFD	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-FinData	**0.07**	0.01	0.05	0.04	0.06	0.01	0.05
CrCFraud	**0.07**	0.03	0.03	0.05	0.04	0.02	0.05
Default	0.15	**0.18**	0.15	0.16	0.12	**0.18**	0.2
campaign	0.08	**0.12**	**0.12**	0.09	0.05	0.1	0.11
fraud	0.0	**0.03**	**0.03**	**0.03**	0.01	**0.03**	**0.03**
AVG	0.06	0.06	0.06	0.06	0.05	0.06	**0.07**

(b) PRC-AUC

Dataset	ABOD	COPOD	ECOD	IForest	LOF	QMCD	Rapid D
SFD	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2018-FinData	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**	**0.04**
CrCFraud	**0.02**	0.01	0.01	0.01	0.01	0.01	**0.02**
Default	0.05	0.07	0.05	0.06	0.04	0.07	**0.08**
campaign	0.03	**0.04**	**0.04**	0.03	0.02	0.03	0.03
fraud	0.0	**0.01**	**0.01**	**0.01**	0.0	**0.01**	**0.01**
AVG	0.02	**0.03**	**0.03**	**0.03**	0.02	**0.03**	**0.03**

(c) ROC-AUC

Dataset	ABOD	COPOD	ECOD	IForest	LOF	QMCD	Rapid D
SFD	0.58	0.55	0.57	0.62	**0.66**	0.52	0.49
2018-FinData	**0.51**	0.46	0.5	0.48	**0.51**	0.46	0.5
CrCFraud	**0.67**	0.55	0.54	0.62	0.59	0.51	0.62
Default	0.7	0.76	0.71	0.72	0.63	0.76	**0.78**
campaign	0.64	**0.69**	**0.69**	0.63	0.56	0.66	0.67
fraud	0.5	0.89	0.89	**0.9**	0.52	**0.9**	**0.9**
AVG	0.6	**0.66**	0.65	**0.66**	0.58	0.64	**0.66**

(d) Execution Time

Dataset	ABOD	COPOD	ECOD	IForest	LOF	QMCD	Rapid D
SFD	48.25	1.31	1.66	3.21	31.32	325.68	**0.21**
2018-FinData	1.01	0.15	0.36	0.11	0.08	1.04	**0.01**
CrCFraud	38.74	1.48	1.7	1.65	26.43	160.46	**0.17**
Default	1.61	**0.01**	0.31	0.28	0.07	0.86	**0.01**
campaign	15.32	1.25	2.26	1.91	4.05	9.6	**0.04**
fraud	70.55	2.75	2.85	2.0	53.95	250.32	**0.19**
AVG	29.25	1.16	1.52	1.53	19.32	124.66	**0.12**

V. Experimental results

First of all, it should be mentioned that the best results per each dataset are shown in bold font. Then, by looking at the average F1-score and PRC-AUC results, one can quickly realize that most of the numbers are very low. The main reason for this typical matter is the imbalance present in the data as by a tinge of misclassification between inliers and outliers, the false alarm rate abruptly goes high, making the final aggregate result plunge quickly [37]. It is also clear that not so much gap is among various methods concerning average ROC-AUC outcomes, and all of them are below 0.7 which does not look appealing for a classification algorithm.

The underlying reason for this concern here is the fact that most probably our datasets are not well crafted and preprocessed for outlier detection purposes. However, in the case of *frad* data in the AUC-ROC table, most of the results seem reasonable which shows that this dataset has been most probably scrutinized and cut out well by some domain experts for anomaly identification purposes.

Finally, in terms of execution time, as expected, Rapid D takes the lead by an average execution time lower than one second. COPOD, ECOD, and IForest are not so much away from Rapid D with an average consumption time lower than two seconds. Rapid D enjoys a very concise distance computation regarding the very small subsample it takes in the beginning. IForest is also gaining benefit from an ensemble of subsamples and similar quick computations to Rapid D. COPOD and ECOD are probabilistic techniques in which every attribute is processed mostly independently from others by which a fast processing speed is attained.

VI. Conclusion

In this short survey, we just went over various techniques, either state-of-the-art (SOTA) or recently published papers for anomaly/outlier detection. Our main concentration was identifying anomalies inside financial datasets, even though we are suffering from a major issue here and it is a lack of enough labeled data in this area. Another issue that we had is the lack of preciseness in outlier labeling which makes the detection process faulty. Except for one of the real-life datasets in our experimentation, named *fraud* which is examined by authentic practitioners and referenced in multiple publications, the attained outcomes make sense; however, even for this dataset, just in the case of ROC-AUC which is not usually recommended for anomaly detection purposes, the acquired outcomes look more appealing than others.

References

[1] C. C. Aggarwal, *Data mining: the textbook*. Springer, 2015.

[2] C. C. Aggarwal and C. C. Aggarwal, *An introduction to outlier analysis*. Springer, 2017.

[3] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406–421, 2018.

[4] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[5] F. Y. Edgeworth, “Xli. on discordant observations,” *The london, edinburgh, and dublin philosophical magazine and journal of science*, vol. 23, no. 143, pp. 364–375, 1887.

[6] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, “Copod: copula-based outlier detection,” in *2020 IEEE international conference on data mining (ICDM)*. IEEE, 2020, pp. 1118–1123.

[7] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[8] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, “Ecod: Unsupervised outlier detection using empirical cumulative distribution functions,” *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[9] H.-P. Kriegel, M. Schubert, and A. Zimek, “Angle-based outlier detection in high-dimensional data,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 444–452.

[10] K.-T. Fang and C.-X. Ma, “Wrap-around l2-discrepancy of random sampling, latin hypercube and uniform designs,” *Journal of complexity*, vol. 17, no. 4, pp. 608–624, 2001.

[11] D. Muhr, M. Affenzeller, and J. Küng, “A probabilistic transformation of distance-based outliers,” *Machine Learning and Knowledge Extraction*, vol. 5, no. 3, pp. 782–802, 2023.

[12] R. H. Gharaei and H. Nezamabadi-Pour, “Rdod: A robust distance-based technique for outlier detection,” in *2022 30th International Conference on Electrical Engineering (ICEE)*. IEEE, 2022, pp. 885–890.

[13] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, “Loci: Fast outlier detection using the local correlation integral,” in *Proceedings 19th international conference on data engineering (Cat. No. 03CH37405)*. IEEE, 2003, pp. 315–326.

[14] M. Sugiyama and K. Borgwardt, “Rapid distance-based outlier detection via sampling,” in *Advances in Neural Information Processing Systems*, 2013, pp. 467–475.

[15] S. D. Bay and M. Schwabacher, “Mining distance-based outliers in near linear time with randomization and a simple pruning rule,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 29–38.

[16] F. Angiulli and F. Fassetti, “Dolphin: An efficient algorithm for mining distance-based outliers in very large datasets,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 1, pp. 1–57, 2009.

[17] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[18] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, “Loop: local outlier probabilities,” in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1649–1652.

[19] F. Aydın, “Boundary-aware local density-based outlier detection,” *Information Sciences*, vol. 647, p. 119520, 2023.

[20] O. A. Bustos-Brinez, J. A. Gallego-Mejia, and F. A. González, “Ad-dmkde: Anomaly detection through density matrices and fourier features,” in *International Conference on Information Technology & Systems*. Springer, 2023, pp. 327–338.

[21] K. G. Mehrotra, C. K. Mohan, H. Huang, K. G. Mehrotra, C. K. Mohan, and H. Huang, “Clustering-based anomaly detection approaches,” *Anomaly Detection Principles and Algorithms*, pp. 41–55, 2017.

[22] J. M. Jobe and M. Pokojovy, “A cluster-based outlier detection scheme for multivariate data,” *Journal of the American Statistical Association*, vol. 110, no. 512, pp. 1543–1551, 2015.

[23] P. J. Rousseeuw and K. V. Driessen, “A fast algorithm for the minimum covariance determinant estimator,” *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999.

[24] J. Huang, Q. Zhu, L. Yang, D. Cheng, and Q. Wu, “A novel outlier cluster detection algorithm without top-n parameter,” *Knowledge-Based Systems*, vol. 121, pp. 32–40, 2017.

[25] A. Hoang, T. N. Mau, D.-V. Vo, and V.-N. Huynh, “A mass-based approach for local outlier detection,” *IEEE Access*, vol. 9, pp. 16 448–16 466, 2021.

[26] K. Liu, Y. Dou, Y. Zhao, X. Ding, X. Hu, R. Zhang, K. Ding, C. Chen, H. Peng, K. Shu *et al.*, “Benchmarking node outlier detection on graphs,” *arXiv preprint arXiv:2206.10071*, 2022.

[27] Y. Li, Y. Wang, X. Ma, C. Qian, and X. Li, “A graph-based method for active outlier detection with limited expert feedback,” *IEEE Access*, vol. 7, pp. 152 267–152 277, 2019.

[28] J. Ko and M. Comuzzi, “Keeping our rivers clean: Information-theoretic online anomaly detection for streaming business process events,” *Information Systems*, vol. 104, p. 101894, 2022.

[29] F. Ye, H. Zheng, C. Huang, and Y. Zhang, “Deep unsupervised image anomaly detection: An information theoretic framework,” in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 1609–1613.

[30] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[31] ——, “Isolation-based anomaly detection,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[32] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[33] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, and J. R. Wells, “Efficient anomaly detection by isolation using nearest neighbour ensemble,” in *2014 IEEE International Conference on Data Mining Workshop*. IEEE, 2014, pp. 698–705.

[34] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, Y. Zhu, and J. R. Wells, “Isolation-based anomaly detection using nearest-neighbor ensembles,” *Computational Intelligence*, vol. 34, no. 4, pp. 968–998, 2018.

[35] S. Hariri, M. C. Kind, and R. J. Brunner, “Extended isolation forest,” *IEEE transactions on knowledge and data engineering*, vol. 33, no. 4, pp. 1479–1489, 2019.

[36] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, “Adbench: Anomaly detection benchmark,” 2022.

[37] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *Proceedings of the 23rd*

international conference on Machine learning, 2006, pp. 233–240.