Article | Received 9 December 2025; Revised 29 April 2026; Accepted 8 May 2026; Published 18 August 2026
https://doi.org/10.55092/blockchain20260006

Unsupervised blockchain anomaly transaction detection via feature distribution learning

Yuzhu Qing[1], Chi Jiang[1], Ming Tao[2] and Yin Zhang[1,*]

[1] School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

[2] School of Computer Science and Technology, Dongguan University of Technology, Dongguan, China

* Correspondence author; E-mail: zhangyin123@uestc.edu.cn.

Highlights:

- Proposes a feature distribution learning framework.
- Compact latent representation learning via Kolmogorov–Arnold Networks.
- Superior anomaly detection performance on blockchain transaction data.
- Interpretable symbolic rules revealing key features for anomaly reasoning.

Abstract: Blockchain transaction data exhibit heterogeneous structures, complex interactions, and highly imbalanced distributions, which pose significant challenges for unsupervised anomaly detection. Existing methods often fail to effectively model the intrinsic feature distributions of normal transactions and typically lack interpretability, limiting their reliability and usability in practical blockchain scenarios. To address these issues, this paper proposes an unsupervised anomaly detection framework based on feature distribution learning. The core idea is to explicitly model the distribution of normal transaction behavior in a compact latent space while preserving interpretability. Specifically, a Kolmogorov–Arnold Network (KAN) is employed to learn a two-dimensional latent representation of blockchain transactions, capturing nonlinear relationships between features. During training, the model minimizes the radius of a hypersphere enclosing normal samples, encouraging a compact and structured distribution. During inference, anomalies are identified based on their geometric deviation from the learned distribution, measured by the distance to the latent-space center. This design avoids reliance on reconstruction errors and enables a more direct and stable decision mechanism. Experiments on Ethereum, Blockchain Network Attack Traffic dataset (BNaT) and Real World Dataset of Cryptocurrency Addresses with Transaction Profiles (Real-CATS) demonstrate that the proposed method consistently outperforms state-of-the-art unsupervised baselines across multiple metrics. Furthermore, the symbolic expressions derived from the learned mapping reveal key transaction features that drive anomaly detection decisions, providing clear interpretability. These results highlight the effectiveness, robustness, and practical value of feature distribution learning for blockchain anomaly detection.

Keywords: blockchain; anomaly detection; unsupervised learning; Kolmogorov–Arnold Network; feature distribution learning; interpretability

1. Introduction

Blockchain platforms generate massive volumes of transactional data, and their open and permissionless architecture enables any participant to interact freely with the network. While such openness facilitates decentralized applications, it also introduces various abnormal behaviors, including malicious smart contract deployment, fraudulent transactions, and transaction flooding attacks, which threaten transaction confirmation efficiency, network performance, and system stability [1,2]. Therefore, accurately identifying abnormal transactions has become an important task for maintaining the security and reliability of blockchain systems.

To address this problem, existing studies have explored different anomaly detection paradigms. Early research mainly relied on supervised learning methods that formulate anomaly detection as a classification task by learning discriminative patterns from labeled transaction data. Recent studies have further incorporated graph neural networks, Transformer architectures, and hybrid learning frameworks to capture complex structural and temporal dependencies in blockchain transactions [3–5]. These methods have achieved encouraging performance when sufficient labeled anomalies are available. However, collecting representative abnormal transaction labels in real blockchain environments is expensive and often impractical, resulting in severe class imbalance and limited generalization capability [6].

To alleviate the dependence on labeled data, increasing attention has been devoted to unsupervised anomaly detection. Representative approaches include reconstruction-based models, distance-based methods, and one-class learning algorithms [7–9]. These methods learn the characteristics of normal transactions using only unlabeled or normal samples and identify anomalies according to reconstruction errors or feature-space distances. Nevertheless, most existing approaches primarily optimize sample reconstruction or feature compactness, while paying insufficient attention to the intrinsic distribution of normal transaction features. Consequently, they often struggle to learn discriminative latent representations in high-dimensional blockchain data, leading to unstable decision boundaries and limited model interpretability.

Motivated by these observations, this paper proposes an unsupervised blockchain anomaly detection framework based on feature distribution learning. Instead of relying solely on reconstruction quality, the proposed method explicitly learns the distribution of normal transaction features in the latent space and constructs a compact hypersphere to characterize normal behavior. Anomalies are then identified according to their deviations from the learned feature distribution. Furthermore, symbolic rule extraction is incorporated to improve model interpretability by revealing the transaction features that contribute most to anomaly detection.

The main contributions of this work are summarized as follows:

(1) We propose a feature distribution learning framework for unsupervised blockchain anomaly detection, which learns compact latent representations while explicitly modeling the distribution of normal transaction features.

(2) We develop a hypersphere-based decision mechanism that detects anomalies according to deviations from the learned feature distribution, avoiding the limitations of reconstruction-error-based methods under highly imbalanced settings.

(3) We derive symbolic expressions from the learned feature mapping to provide interpretable explanations of anomaly detection, revealing key transaction features associated with abnormal behaviors.

Extensive experiments on representative blockchain datasets demonstrate that the proposed method achieves superior detection performance while providing better latent-space separability and enhanced interpretability.

2. Related work

2.1. Blockchain anomaly transaction detection

Blockchain anomaly transaction detection has attracted increasing attention due to the rapid growth of decentralized applications and the security challenges arising from open transaction environments. Existing studies mainly focus on identifying malicious accounts, fraudulent transactions, and abnormal contract behaviors by exploiting transaction topology, temporal information, and behavioral characteristics.

Graph-based methods model transaction dependencies to capture high-order interaction patterns. For example, Transaction Subgraph Networks (TSGN) construct transaction graphs to improve phishing account detection [3]. Sketch-based frameworks compress large-scale blockchain transaction streams to improve detection efficiency while reducing storage overhead [10]. Machine learning and hybrid approaches further combine statistical analysis with handcrafted transaction features to distinguish abnormal behaviors [4,11]. Representative dedicated graph learning methods include Detecting Illicit Accounts in Multigraphs (DIAM), which leverages Edge2Seq and multigraph discrepancy modules on attributed directed multi-graphs to detect illicit crypto accounts [12]. More recently, Transformer-based models have been introduced to capture temporal dependencies and complex feature interactions in blockchain transactions [5]; a typical work is BERT4ETH, a pre-trained Transformer encoder equipped with Ethereum-specific pre-training strategies for universal account fraud representation learning [13], while unsupervised learning methods have been explored to reduce the reliance on labeled anomaly samples [7].

Although these methods have demonstrated promising performance, they still suffer from several limitations. Most supervised approaches require sufficient labeled anomalies, which are difficult to obtain in real blockchain systems. Existing unsupervised methods mainly rely on reconstruction errors or conventional feature learning and often fail to explicitly characterize the intrinsic distribution of normal transaction features, limiting both detection robustness and model interpretability.

2.2. General unsupervised anomaly detection

General unsupervised anomaly detection methods can be broadly categorized into discriminative methods (DMs), generative models (GMs), and outlier exposure (OE)-based approaches. These paradigms differ in how they characterize normality and distinguish anomalous samples. As summarized in Table 1, we briefly review these representative categories together with their main limitations.

Table 1. Summary of unsupervised anomaly detection categories and their limitations.

Category	Description	Limitations
DMs	Learn decision boundaries of normal data	Sensitive to distribution and scaling, weak interpretability
GMs	Model data distribution via reconstruction or latent variables	Reconstruction bias, latent space hard to interpret
OE	Use auxiliary anomalies to improve detection	Depend on anomaly quality, limited generalization

Discriminative methods aim to characterize the boundary of normal data. Classical approaches such as One-Class Support Vector Machine (OCSVM) [14] and Isolation Forest [15] construct decision boundaries based on feature space separation or random partitioning. Deep Support Vector Data Description (DeepSVDD) [16] further learns compact representations by enclosing normal samples within a hypersphere. Density-aware methods such as Copula-Based Outlier Detection (COPOD) [17] and Empirical Cumulative Distribution Functions-Based Outlier Detection (ECOD) [18] estimate extremeness using feature distributions, while contrastive and meta-learning based approaches including Internal Contrastive Learning (ICL) [19], Adaptive Centered Representations (ACR) [20], and Rejecting via Exceed (REJEX) [21] improve generalization and robustness. However, these methods heavily depend on boundary assumptions and are sensitive to feature scaling and distribution shifts, often lacking semantic interpretability of the learned decision boundaries.

Generative modeling provides another important direction. Autoencoders (AEs) [22] and Variational Autoencoders (VAEs) [23] detect anomalies via reconstruction errors or latent distributions. Variants such as L0-Norm Constrained Autoencoders (L0-AE) [24], Fourier-based models [25], Normalized Autoencoder (NAE) [26], and Robust Dual Autoencoder (RDAE) [27] improve robustness and representation learning. Hybrid and advanced architectures, including Detecting Outlier Machine Instances (DOMI) [28], Non-Parametric Transformers (NPTs) [29], Active Autoencoders (AAE) [30], Scale Learning-based Deep Anomaly Detection (SLAD) [31], Heterogeneous Autoencoder (HAE) [32], Variational AutoEncoder with Spatial constrained network and recursive reconstruction strategy (VESC) [33], Memory-Guided Transformer (MEMTO) [34], and Nominality Score Conditioned Time Series Anomaly Detection by Point/Sequential Reconstruction (NPSR) [35], further enhance modeling capacity. Nevertheless, reconstruction-based methods tend to suffer from over-generalization, where anomalies can be well reconstructed, and their latent representations are often difficult to interpret.

OE-based methods improve anomaly detection by incorporating auxiliary anomalous samples. Distributional-Agnostic Outlier Exposure (DOE) [36] generates worst-case perturbations, while Diversified Outlier Exposure (DivOE) [37] and Reliable Outlier Synthesis Under Noisy Feature Space (RONF) [38] increase anomaly diversity. Prototypical Learning-Guided Context-Aware Segmentation Network (PCSNet) [39] enhances separability through anomaly feature separation, and Robust Outlier Detection via Exposing Adaptive Out-of-Distribution Samples (RODEO) [40] leverages adversarial generation to synthesize contextual anomalies. More recent methods such as Normalized Outlier Distribution Adaptation (AdaptOD) [41] and Diffusion-based Layer-wise Semantic Reconstruction (DLSR) [42]

address distribution shifts using dynamic or diffusion-based strategies. However, OE-based approaches rely on the quality and representativeness of auxiliary anomalies, which may not reflect real-world abnormal patterns and can introduce bias.

Although these general anomaly detection methods have achieved remarkable success across different application domains, directly applying them to blockchain transaction analysis remains challenging. Blockchain transactions exhibit complex feature distributions, strong heterogeneity, and severe class imbalance, making it difficult for existing discriminative, generative, and OE-based methods to accurately characterize normal transaction behaviors. Moreover, most existing approaches provide limited interpretability, as discriminative methods learn implicit decision boundaries, generative models encode information into difficult-to-explain latent representations, and OE-based methods rely on externally generated anomalies. Existing explanation techniques are also primarily designed for supervised settings and do not explicitly model the distribution of normal transaction behaviors [43]. Therefore, there remains a need for an interpretable unsupervised framework that explicitly learns feature distributions for blockchain anomaly transaction detection.

3. Preliminaries

3.1. Blockchain anomaly transaction detection

Blockchain systems maintain a decentralized and append-only ledger that records all transaction activities. Their transparency and immutability provide strong security guarantees, yet the openness of participation and the complexity of transaction execution create opportunities for abnormal and malicious behaviors. In practice, many anomalies are reflected through abnormal transaction patterns, which directly affect network performance, system stability, and user security [44,45].

This work focuses on anomaly types that are directly observable through transaction-related behaviors. Abnormal transactions are mainly associated with resource consumption and system-level attacks, including malicious smart contracts that trigger abnormal execution through excessive opcode usage leading to memory exhaustion, large-scale transaction flooding that causes network congestion and abnormal gas dynamics, and state expansion attacks that generate massive numbers of empty accounts and significantly increase storage and synchronization overhead [46,47]. These anomalies manifest as clear deviations in transaction frequency, value distribution, and address interaction patterns.

Network-level malicious behaviors can also be reflected in transaction-related features. Typical examples include denial-of-service attacks such as Synchronize (SYN) flooding, transaction flooding that overwhelms the transaction pool, brute-force account access attempts, and man-in-the-middle behaviors that manipulate or inject abnormal transactions. Although these attacks originate from different layers of the blockchain ecosystem, they ultimately introduce observable irregularities in transaction statistics and communication patterns.

Therefore, this work focuses on detecting behavioral anomalies manifested through transaction features. By modeling the distribution of normal transaction behavior and identifying deviations from this distribution, anomaly detection methods can effectively capture a wide range of abnormal activities within this defined scope.

Blockchain data itself is high-dimensional and weakly structured, involving complex relational dependencies and heterogeneous temporal dynamics. Simple statistical features are insufficient to capture these properties, and emerging abnormal patterns may differ from historical observations. As a result, data-driven models that learn transaction feature distributions in an unsupervised manner have become a central research topic [48].

Given a dataset consisting of n transaction samples,

$$\{x_1, x_2, \ldots, x_n\}, \quad x_i \in \mathbb{R}^d \tag{1}$$

each x_i denotes a d-dimensional feature vector. The objective of anomaly detection is to learn a mapping that distinguishes normal instances from abnormal ones,

$$f : \mathbb{R}^d \to \{0, 1\} \tag{2}$$

where $f(x) = 1$ indicates an anomaly and $f(x) = 0$ indicates a normal transaction.

In practical blockchain environments, abnormal samples are rare and labels are often unavailable. Therefore, the training process typically follows an unsupervised formulation, where the model must learn the distribution of normal transactions and identify deviations. Let z denote the latent representation learned from the input:

$$z = E(x) \tag{3}$$

and let $\widehat{x}$ be a reconstruction or distribution-based estimation,

$$\widehat{x} = D(z) \tag{4}$$

then the anomaly score is often derived from a deviation measure such as

$$\|x - \widehat{x}\| \tag{5}$$

A sample is treated as anomalous when its deviation exceeds a threshold determined from normal-pattern statistics. This distribution-oriented viewpoint motivates the use of representation learning methods that explicitly model feature distributions and identify subtle irregularities.

3.2. KAN

Kolmogorov–Arnold Network (KAN) [49] is a neural architecture inspired by the Kolmogorov–Arnold representation theorem. The theorem states that any multivariate continuous function can be decomposed into finite compositions and sums of univariate continuous functions. Formally, for a continuous function $f : [0, 1]^d \to \mathbb{R}$,

$$f(x_1, \ldots, x_d) = \sum_{k=1}^{2d+1} \Phi_k \left(\sum_{j=1}^{d} \varphi_{k,j}(x_j) \right) \tag{6}$$

where $\varphi_{k,j}$ and Φ_k are univariate functions.

KAN operationalizes this theorem by replacing traditional fixed-form activation functions with learnable one-dimensional functions. For a KAN layer with input dimension d_{in} and output dimension d_{out}, the layer consists of a matrix of learnable univariate functions:

$$\Phi = \{\varphi_{k,j}\}, \quad j = 1, \ldots, d_{\text{in}}, \quad k = 1, \ldots, d_{\text{out}} \tag{7}$$

A full KAN network composed of L such layers can be written as

$$\mathrm{KAN}(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_1 \circ \Phi_0)(x) \tag{8}$$

The expressive power of KAN stems from its adaptive nonlinear transformations. Unlike multilayer perceptrons (MLPs) that rely on fixed activations such as Rectified Linear Unit (ReLU) or Tanh, each $\varphi_{k,j}$ in a KAN layer is learned directly from data. This flexibility enables the model to capture complex nonlinear relationships with substantially fewer parameters.

The parameter complexity of a KAN layer depends on the number of learnable spline segments. For a network of depth L, width N, spline interval count G, and spline degree k (typically $k = 3$), the approximate number of parameters is

$$O(N^2 L(G+k)) \sim O(N^2 LG) \tag{9}$$

Although this is slightly larger than an MLP with the same width, the required width for KAN is typically lower, leading to competitive or even reduced computational cost.

Given its strong local adaptability, efficient parameterization, and inherent interpretability, KAN serves as an effective backbone for feature-distribution modeling in blockchain anomaly detection tasks, where subtle deviations in transaction patterns must be accurately captured.

4. Methods

4.1. Method framework

This subsection presents the overall framework of the proposed unsupervised blockchain anomaly transaction detection method. Blockchain transaction features are typically heterogeneous, nonlinear, and highly imbalanced, making normal behavior difficult to characterize using reconstruction-based methods. To address this challenge, the proposed framework employs KAN to learn the intrinsic distribution of normal transaction features in a compact latent space and detects anomalies according to their deviations from this distribution. Unlike autoencoders or DeepSVDD, the proposed method explicitly models feature distributions while preserving the symbolic interpretability of KAN. Only normal transactions are required during training, making the framework suitable for practical blockchain anomaly detection scenarios where abnormal samples are scarce or unavailable.

As illustrated in Figure 1, the framework consists of a training phase and a testing phase. During training, each normal blockchain transaction represented by its feature vector $X \in \mathbb{R}^d$ is mapped into a latent representation $Z \in \mathbb{R}^m$ through the KAN-based symbolic mapping function $\phi(\cdot)$, yielding

$$Z = \phi(X) \tag{10}$$

The training objective encourages the latent representations of normal transactions to lie within a hypersphere of radius R, so that the latent distribution becomes concentrated around a shared center. This promotes compact representations of normal transaction features and improves their separability from abnormal behaviors without relying on manually designed heuristics. In the testing phase, an incoming transaction with feature vector X^* is processed by the same mapping function to obtain its latent representation

$$Z^* = \phi(X^*) \tag{11}$$

Its anomaly score is then computed as the Euclidean distance to the center of the hypersphere

$$s(X^*) = \|Z^* - c\|_2 \tag{12}$$

and transactions whose distance exceeds the learned boundary are regarded as anomalous. This geometric decision rule derives directly from the structure of the latent space shaped during training, and the symbolic formulation provided by KAN allows the anomaly decision to be expressed as an explicit analytic equation.

By explicitly learning the distribution of normal blockchain transaction features and expressing the decision boundary in symbolic form, the proposed framework provides an efficient and interpretable solution for unsupervised blockchain anomaly transaction detection.

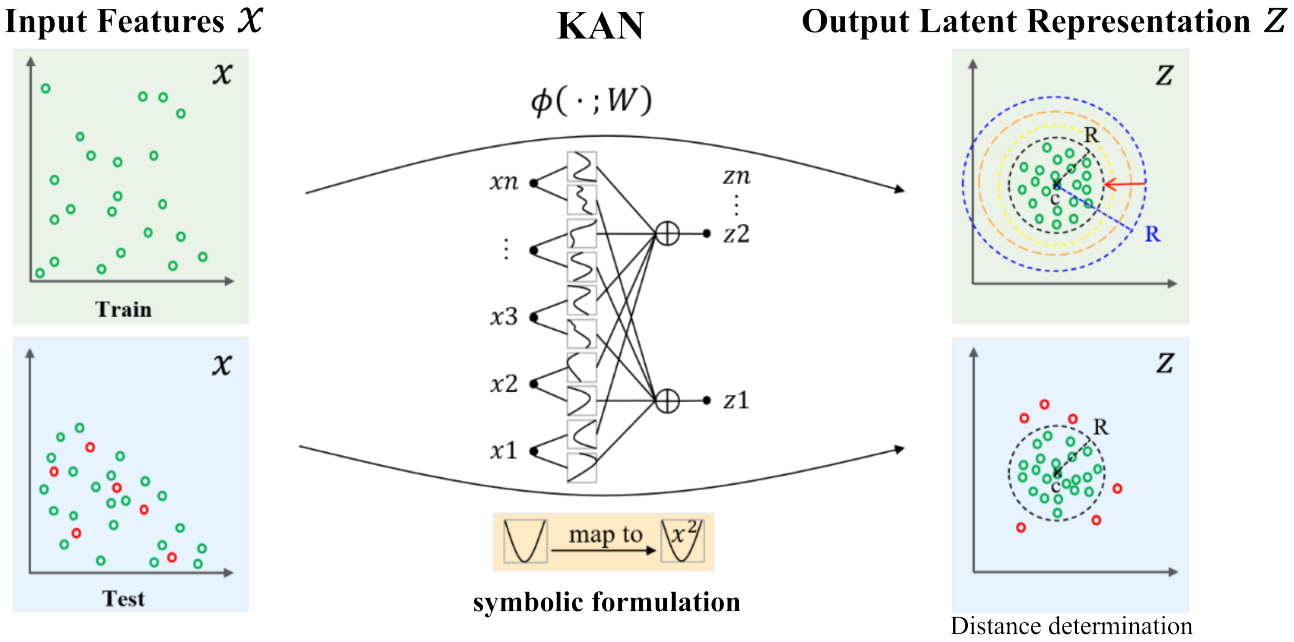

Figure 1. Overall framework of the proposed method.

4.2. Training stage

The training stage aims to learn a compact latent feature distribution from normal blockchain transaction samples. A KAN-based nonlinear mapping is used to project each input transaction into a low-dimensional latent space, where the normal samples are constrained to lie inside a hypersphere with minimal radius. The training process jointly optimizes the KAN parameters and the hypersphere radius while fixing the hypersphere center to avoid degenerate collapse. This section details the objective function, center selection strategy, and training algorithm.

4.2.1. Feature distribution optimization objective

Let the input space be $X \subset \mathbb{R}^d$ and the latent feature space be $F \subset \mathbb{R}^p$. A KAN-based mapping function

$$\Phi(\cdot; W) : X \to F \tag{13}$$

parameterized by W is used to obtain the latent representation for each input sample $x \in X$. The optimization objective encourages all normal samples to be enclosed within a hypersphere centered at c with radius R. The objective function is

$$\min_{W,R} R^2 + \lambda \|W\|_F^2 \tag{14}$$

subject to the constraint

$$\|\Phi(x_i;W) - c\|^2 \leq R^2, \quad \{i = 1,\ldots,n\} \tag{15}$$

where $c \in F$ is the hypersphere center, R is the radius, and $\lambda > 0$ is a regularization coefficient. The term $\|W\|_F^2$ smooths the learnable spline coefficients of the KAN, improving generalization and stability. The nonlinear mapping learned by KAN allows the boundary to adapt to the intrinsic feature distribution of normal blockchain transactions while retaining symbolic interpretability.

4.2.2. Selection of the hypersphere center

When the hypersphere center c is optimized jointly with the network parameters, the model may collapse to a trivial solution where all samples are mapped to a single point. Specifically, consider a parameter configuration $W_0 = 0$, under which the KAN produces a constant output for any input,

$$\Phi(x;W_0) = c_0, \quad \forall x \tag{16}$$

and if the hypersphere center satisfies $c = c_0$, the optimal solution degenerates to $W = W_0$ and $R = 0$. In this case, the hypersphere becomes meaningless and anomaly detection fails entirely.

To avoid such degeneracy, the hypersphere center is fixed during training. In this work, the center is set to the origin, $c = \mathbf{0}$, which is consistent with the architectural characteristics of KAN. Due to the spline-based activations and layer compositions, the latent representations produced by KAN tend to be naturally centered around zero at the early stage of optimization. This makes the origin a stable and symmetric reference point for hypersphere learning, preventing collapse and ensuring that the learned latent space maintains a meaningful geometric structure for effective anomaly detection.

4.2.3. Training algorithm

The complete training phase is summarized in Algorithm 1. The algorithm iteratively updates network parameters using mini-batch optimization while enforcing the hypersphere constraint. The hypersphere radius is periodically adjusted to maintain a tight boundary around normal samples.

Algorithm 1 Training procedure of the proposed feature distribution learning method

Input: Normal training set $X = \{x_1,\ldots,x_n\}$; learning rate η; regularization parameter λ; update interval k
Output: Trained KAN parameters W and hypersphere radius R

```
Initialize KAN parameters W, randomly initialize R;
Fix the hypersphere center: c ← 0;
while not converged do
    for each mini-batch B = {x_1,...,x_b} ⊂ X do
        Compute latent features Φ(x_i;W) for all x_i ∈ B;
        Compute loss: L = R^2 + λ‖W‖_F^2;
        Apply constraint: ‖Φ(x_i;W) − c‖^2 ≤ R^2;
        Compute gradients ∇_W L, ∇_R L;
        Update parameters:;
        W ← W − η∇_W L;
        R ← R − η∇_R L;
    if epoch is a multiple of k then
        Adjust radius R;
return W, R;
```

During training, multiple regularization terms are incorporated to improve stability and interpretability. L1 regularization promotes sparsity among spline coefficients, while entropy regularization constrains the complexity of activation functions. Additional coefficient and smoothness constraints further regulate adjacent spline parameters, preventing oscillatory behavior. The Adam optimizer is used throughout training with a step size of 50 and an initial learning rate of 1.0, providing efficient and stable convergence.

4.3. Testing stage

In the testing stage, both normal and anomalous blockchain transactions are passed through the trained KAN to obtain their latent representations. Each test sample is evaluated according to its distance from the fixed hypersphere center, and anomaly detection is performed based on the hypersphere threshold determined during training.

Given a test sample x_{test}, its latent representation is first computed using the trained KAN parameters W,

$$z_{\text{test}} = \Phi(x_{\text{test}}; W) \tag{17}$$

The Euclidean distance between the latent point and the hypersphere center c is then computed as

$$D = \|z_{\text{test}} - c\| \tag{18}$$

To determine whether the test sample lies inside or outside the normal region, the distance is compared with a threshold R_{th}. This threshold is derived from the distribution of distances of normal training samples, typically chosen as the 99% percentile,

$$R_{\text{th}} = P_{99}(D_{\text{train}}) \tag{19}$$

where D_{train} denotes the set of distances between normal training samples and the center in the latent space, and $P_{99}(\cdot)$ denotes the empirical 99% percentile operator.

The final decision rule is

$$\text{if } D > R_{\text{th}},\ x_{\text{test}} \text{ is anomalous} \tag{20}$$

$$\text{if } D \leq R_{\text{th}},\ x_{\text{test}} \text{ is normal} \tag{21}$$

The testing procedure of this approach is outlined in Algorithm 2. This distance-based detection approach leverages the compact latent distribution learned during training and avoids reconstruction noise or density estimation errors common in autoencoder-based or probabilistic methods. By measuring the deviation from the learned hypersphere boundary, it provides a stable and discriminative anomaly indicator.

Algorithm 2 Anomaly detection procedure in the testing stage

Input: Test set X_{test}; trained KAN parameters W; fixed center c; threshold R_{th}
Output: Predicted anomaly labels Y_{pred}

Initialize prediction list: $Y_{\text{pred}} \leftarrow [\,]$;
for *each sample* $x_i \in X_{test}$ **do**
    Compute latent representation: $z_i = \Phi(x_i; W)$;
    Compute distance to center: $D_i = \|z_i - c\|$;
    if $D_i > R_{th}$ **then**
        Mark x_i as anomalous and append 1 to Y_{pred};
    else
        Mark x_i as normal and append 0 to Y_{pred};
return Y_{pred};

4.4. Network relation symbolization

In the anomaly detection setup, the variables in the latent space often exhibit noticeable nonlinear interactions. To enhance model interpretability and enable structured analysis of the learned representations, symbolic regression is introduced to transform the latent-space mapping into explicit mathematical expressions. The idea is to search within a predefined set of symbolic operators and identify functional compositions that best describe the relations embedded in the latent space. This symbolic form allows the mapping behavior of the model to be expressed in a readable and mathematically grounded manner.

The symbolic library contains polynomial operators x, x^2, x^3, x^4 for representing higher-order nonlinear patterns, the exponential operator $\exp(\cdot)$ for capturing growth or decay behavior, the logarithmic operator $\log(\cdot)$ for range compression, the square-root operator $\sqrt{\cdot}$ for smoothing extreme values, trigonometric operators $\sin(\cdot)$, $\tan(\cdot)$, and $\tanh(\cdot)$ for periodic and bounded transformations, and the absolute-value operator $|\cdot|$ for modeling symmetric variations. These operators cover a wide spectrum of expressiveness, follow efficient numerical computations, and carry clear mathematical semantics, which strengthens the interpretability of the final symbolic expressions.

Symbol generation is performed based on the latent representations produced by the model. A symbolic regression algorithm explores compositions from the operator library and optimizes the parameters associated with the constructed expressions. During optimization, gradient descent is used to minimize the discrepancy between the symbolic output and the latent-space values. The loss function is defined as $L(\theta) = \sum_{i=1}^{N} (z_i - \hat{z}_i)^2$, where z_i denotes the latent representation obtained from the model, $\hat{z}_i$ is the value predicted by the symbolic expression, and θ denotes the parameters in the symbolic formula. To prevent overfitting and maintain concise expressions, an L_2 regularization term $\lambda \sum_{j=1}^{M} \theta_j^2$ is included. This regularization stabilizes the optimization process and constrains the symbolic expression to remain compact.

After optimization, the resulting symbolic expressions provide an explicit representation of the latent-space relationships. The expressions offer a transparent view of how the model organizes and utilizes latent features, which supports interpretability and strengthens the reliability of the anomaly detection pipeline.

4.5. Computational complexity analysis

In this section, we analyze the computational complexity of the proposed model in terms of both space and time, highlighting the efficiency implications of our spline-based edge parameterization. Let L denote the network depth (number of layers), N the layer width (assuming all layers have the same number of nodes), G the number of intervals in the univariate B-spline grid (corresponding to $G+1$ control points), and k the order of the B-spline basis functions (default $k = 3$ for cubic splines). A key characteristic of our model is that it does not rely on traditional linear weight matrices; instead, each edge is parameterized by a univariate B-spline function, with parameters determined by the spline coefficients.

The space complexity is primarily determined by the spline coefficients associated with each edge. Each layer contains $N_{\text{in}} \times N_{\text{out}}$ edges, which simplifies to N^2 when $N_{\text{in}} = N_{\text{out}} = N$. Each edge is represented by a univariate B-spline function requiring $G + k$ coefficients to define the

spline. Consequently, for L layers, the total number of parameters is $L \times N^2 \times (G+k)$, which can be approximated as $O(N^2LG)$ since in practice $k \ll G$. This linear dependence on the grid size G, the square of the layer width N^2, and the depth L allows the model to scale efficiently while maintaining flexible spline-based edge representations.

The time complexity can be analyzed separately for the training and inference phases. During training, let T denote the number of iterations. In each iteration, the core computation involves both forward and backward propagation through all edges and layers. Evaluating a univariate B-spline function requires $O(G)$ operations due to the linear combination of its basis functions. Therefore, the computational cost of a single training iteration is $O(N^2LG)$, and the total training complexity across T iterations is $O(T \times N^2LG)$. For inference, the complexity is limited to forward propagation only, without iterative updates. Each edge still requires $O(G)$ operations to evaluate the spline function, and the nodes aggregate these edge contributions across L layers. Thus, the inference time complexity is $O(N^2LG)$, which matches the complexity of a single training iteration, with no additional overhead.

In summary, both the storage and computational costs of the proposed spline-based network grow quadratically with the layer width, linearly with network depth, and linearly with the number of spline grid intervals, demonstrating that the model is scalable and computationally tractable for large blockchain transaction datasets.

5. Experiments

5.1. Experimental setup

5.1.1. Environment

All experiments were conducted on a workstation running Windows 10, equipped with an Intel Core i7-1165G7 CPU (2.80 GHz) and 16 GB RAM. The implementation was developed in Python 3.8 using PyTorch 1.10.2, and all models were evaluated under identical computational settings.

5.1.2. Datasets

We evaluate our method on three blockchain-related datasets: the Ethereum transaction dataset (Ethereum [50]), the Blockchain Network Attack Traffic dataset (BNaT [51]), and Real World Dataset of Cryptocurrency Addresses with Transaction Profiles (Real-CATS [52]). The Ethereum dataset contains externally owned account (EOA) transactions collected between 2016 and 2017, with anomalous transactions identified from documented attack activities. The BNaT dataset contains normal and malicious network-layer traffic generated from a private Ethereum network, covering multiple realistic attack types. The Real-CATS dataset is a Bitcoin transaction dataset whose collected historical on-chain data spans up to March 2025, featuring 41-dimensional transaction node features extracted from raw Bitcoin block data parsed via a full Bitcoin node. Anomalous transaction samples in Real-CATS are labelled based on user-reported malicious cryptocurrency activity records collected from the ChainAbuse platform, while normal transaction samples are sourced from verified exchange addresses complying with strict Know Your Customer (KYC) policies. All three datasets provide highly imbalanced anomaly distributions, making them suitable for evaluating unsupervised anomaly detection methods. Following

common practice in unsupervised anomaly detection, each dataset is split into training, validation, and test sets with a ratio of 6:2:2, where the training set contains only normal samples.

5.1.3. Evaluation metrics

We compare model performance using widely adopted anomaly detection metrics: true positive rate (TPR), true negative rate (TNR), precision, F1 score, Area Under the Receiver Operating Characteristic curve (AUROC), Area Under the Precision-Recall curve (AUPR), and False Positive Rate at 95% True Positive Rate (FPR95). These metrics capture different aspects of detection capability, including classification accuracy under imbalance, threshold-independent ranking quality, and robustness at high-recall operating points.

5.1.4. Baseline methods

For comparison, we evaluate a suite of baselines including widely-used generic anomaly detection paradigms and two representative dedicated blockchain fraud detection methods, as elaborated below. All these baselines are mainstream benchmark algorithms adopted in blockchain anomaly transaction detection literature and cover core technical paradigms from existing research [53–57].

Discriminative models: OCSVM [14], IForest [15], DeepSVDD [16], and COPOD [17]. These methods aim to learn boundaries or scoring functions that separate normal and abnormal samples through discriminative feature learning.

Generative models: AE [22], VAE [23], RDAE [27], and AAE [30]. They reconstruct normal patterns or model latent distributions, identifying anomalies as samples with low reconstruction fidelity or low likelihood.

Outlier Exposure methods: PCSNet [39] and RODEO [40]. These methods enhance robustness by synthesizing or incorporating pseudo-anomalies during training to improve decision boundaries in open-set environments.

Blockchain-Specific methods (BSMs): DIAM [12], BERT4ETH [13]. DIAM designs Edge2Seq and multigraph discrepancy modules to capture transaction patterns and node feature gaps on directed multi-graphs for illicit account detection. BERT4ETH leverages a pre-trained Transformer encoder with tailored training strategies to capture sequential transaction dynamics and extract account representations for Ethereum fraud detection.

All baseline results were reproduced under the same data splits and evaluation procedures to ensure fairness.

5.2. Comparative results

To comprehensively assess the effectiveness of the proposed feature distribution learning (FDL) method for unsupervised blockchain anomaly transaction detection, this subsection compares FDL with a broad set of representative baseline approaches under a unified experimental environment. The evaluation adopts four standard metrics widely used in anomaly detection, namely TPR, TNR, Precision, and F1-score. These metrics jointly reflect detection sensitivity, false-positive control, and overall decision accuracy. All methods are tested on three blockchain transaction datasets—Ethereum, BNaT and Real-CATS—to examine performance stability under heterogeneous data distributions.

Table 2 summarizes the comparative results. As shown in the table, FDL achieves consistently superior performance and outperforms all considered baselines across all four metrics. On the BNaT dataset in particular, FDL reaches an F1-score of 0.9998, approaching the theoretical optimum, while it also obtains the best F1 of 0.9864 on the latest Real-CATS dataset; the majority of competing methods remain below 0.99 on BNaT and lag notably on Real-CATS.

Discriminative approaches such as OCSVM exhibit relatively stable behavior on the Ethereum dataset but degrade noticeably on BNaT and Real-CATS, suggesting limited generalization to distributional shifts. IForest shows substantial fluctuations in TNR due to the randomness of partitioning, which also leads to low precision. Although DeepSVDD attains high TPR, its precision is significantly lower than that of FDL, indicating a higher false-alarm rate. COPOD performs poorly on Ethereum and yields unbalanced results on BNaT and Real-CATS, likely due to its reliance on strong distributional assumptions that do not hold for complex transaction patterns.

For generative models, AE and AAE demonstrate competitive performance, yet FDL still surpasses them in TNR and precision across all datasets, confirming the effectiveness of learning feature distributions rather than reconstruction error alone. VAE produces slightly lower F1-scores than AE, possibly due to noise introduced by latent sampling. RDAE does not achieve the expected robustness and shows clear degradation on BNaT and Real-CATS, revealing sensitivity to adversarial or irregular anomalies. PCSNet performs close to generative baselines but remains constrained by PCA-based linear assumptions. RODEO fails on all datasets, suggesting that pseudo-anomaly generation is unsuitable for irregular blockchain transaction patterns. Among blockchain-specific baselines, DIAM and BERT4ETH achieve decent overall performance but are still outperformed by FDL across Ethereum, BNaT and Real-CATS. The superior results of FDL highlight its strong stability, generalization, and robustness. The method maintains high TPR and TNR across datasets with substantially reduced sensitivity to distributional variations, empirically validating that feature distribution learning provides a more reliable characterization of normal transaction behaviors compared with reconstruction-based, pseudo-anomaly-based or existing blockchain-specific strategies.

Table 2. Comparative results (TPR, TNR, Precision, F1) on Ethereum, BNaT and Real-CATS datasets (**Bold**: best).

Method		Ethereum				BNaT				Real-CATS			
		TPR	**TNR**	**Precision**	**F1**	**TPR**	**TNR**	**Precision**	**F1**	**TPR**	**TNR**	**Precision**	**F1**
DMs	OCSVM	0.9978	0.9970	0.9851	0.9914	0.9980	0.9923	0.9322	0.9640	0.9232	0.9478	0.9259	0.9245
	IForest	0.9944	0.7425	0.4308	0.6012	0.8681	0.9814	0.8308	0.8491	0.8576	0.9002	0.8553	0.8564
	DeepSVDD	0.9996	0.8989	0.6597	0.7948	0.9980	0.9021	0.5175	0.6816	0.9113	0.8328	0.6849	0.7820
	COPOD	0.0021	0.8952	0.0040	0.0027	**1.00**	0.9133	0.5484	0.7083	0.2681	0.8133	0.2484	0.2579
GMs	AE	0.9978	0.9956	0.9783	0.9880	0.9942	0.9993	0.9942	0.9942	0.9319	0.9539	0.9394	0.9356
	VAE	0.9978	0.9874	0.9396	0.9678	0.9885	0.9980	0.9819	0.9852	0.9423	0.9889	0.9469	0.9446
	RDAE	0.9765	0.9385	0.9541	0.9503	0.9376	0.9351	0.6028	0.7338	0.6142	0.9158	0.6440	0.6288
	AAE	0.9981	0.9946	0.9733	0.9856	0.9951	0.9992	0.9932	0.9942	0.9631	0.9894	0.9608	0.9619
OE	PCSNet	0.9979	0.9868	0.9367	0.9663	0.9952	0.9975	0.9766	0.9858	0.9056	0.9136	0.9043	0.9049
	RODEO	0.00	**1.00**	0.00	0.00	0.0010	**0.9998**	0.3333	0.0019	0.0467	0.9278	0.2707	0.0798
BSMs	DIAM	0.9885	0.9879	0.9891	0.9888	0.9844	0.9855	0.9837	0.9841	0.9319	0.9140	0.9495	0.9406
	BERT4ETH	0.9875	0.9839	0.9876	0.9876	0.9547	0.9561	0.9339	0.9442	0.9068	0.9567	0.9179	0.9123
Ours	FDL	**0.9987**	0.9997	**0.9987**	**0.9987**	0.9997	**0.9998**	**0.9988**	**0.9998**	**0.9863**	**0.9997**	**0.9865**	**0.9864**

Table 3 reports the AUROC, AUPR, and FPR95 results of all baseline methods and the proposed approach on the Ethereum, BNaT and Real-CATS datasets. These metrics jointly reflect detection ranking quality, anomaly identification precision under class imbalance, and false detection robustness. The results reveal substantial performance gaps among the evaluated methods across all three datasets.

As shown in Table 3, the proposed feature distribution learning method consistently achieves the best performance across nearly all metrics. On the Ethereum dataset, it reaches an AUROC of 0.9999 and an AUPR of 0.9998, both nearly perfect, while maintaining an extremely low FPR95 of 0.0002. On the latest Real-CATS dataset, our FDL still obtains the leading AUROC of 0.9924, AUPR of 0.9945 and ultra-low FPR95 of 0.0008. This indicates that the method not only ranks anomalies correctly but also produces accurate high-confidence predictions with minimal false alarms across heterogeneous blockchain data. In contrast, discriminative models such as IForest and COPOD perform significantly worse across all datasets, with COPOD reaching an FPR95 as high as 0.5095 on Ethereum and 0.4837 on Real-CATS, suggesting that statistical assumptions made by these models do not generalize well to complex blockchain transaction patterns.

Table 3. Comparative results (AUROC, AUPR, FPR95) on Ethereum, BNaT and Real-CATS datasets (**Bold**: best).

Method		Ethereum			BNaT			Real-CATS		
		AUROC	**AUPR**	**FPR95**	**AUROC**	**AUPR**	**FPR95**	**AUROC**	**AUPR**	**FPR95**
DMs	OCSVM	0.9974	0.9545	0.0019	0.9994	0.9877	0.0007	0.9306	0.9629	0.0084
	IForest	0.7511	0.2625	0.2553	0.9839	0.7773	0.0581	0.7078	0.6292	0.0795
	DeepSVDD	0.9923	0.8757	0.0074	0.9994	0.9951	0.0004	0.9804	0.9025	0.0031
	COPOD	0.5527	0.1656	0.5095	0.9993	0.9783	0.0018	0.5592	0.4359	0.4837
GMs	AE	0.9961	0.9233	0.0038	0.9962	0.9675	0.0004	0.9027	0.9795	0.0053
	VAE	0.9956	0.9394	0.0079	0.9992	0.9832	0.0006	0.9007	**0.9974**	0.0041
	RDAE	0.9906	0.9953	0.0274	0.9476	0.4801	0.0649	0.8043	0.6917	0.2628
	AAE	0.9952	0.9101	0.0053	0.9997	0.9937	0.0007	0.9812	0.9803	0.0054
OE	PCSNet	0.9981	0.9565	0.0019	0.9992	0.9946	0.0005	0.9175	0.9019	0.0295
	RODEO	0.9994	0.9990	**0.00**	0.9992	0.9756	0.0008	0.9453	0.9525	0.0194
BSMs	DIAM	0.9946	0.9865	0.0045	0.9983	0.9868	0.0010	0.9103	0.9474	0.0266
	BERT4ETH	0.9867	0.9874	0.0073	0.9528	0.9557	0.0114	0.9340	0.9584	0.0023
Ours	FDL	**0.9999**	**0.9998**	0.0002	**0.9998**	**0.9969**	**0.0001**	**0.9924**	0.9945	**0.0008**

On the BNaT dataset, the proposed approach again demonstrates the strongest performance, achieving 0.9998 AUROC, 0.9969 AUPR, and a remarkably low FPR95 of 0.0001. Generative models such as AAE and VAE show reasonable results, yet both fall short of the proposed method in precision–recall characteristics on all three datasets. RDAE performs well on Ethereum AUPR but degrades sharply on BNaT and Real-CATS, confirming its sensitivity to distributional shifts.

Among OE-based approaches, PCSNet and RODEO exhibit competitive AUROC values, and RODEO even attains the best FPR95 on Ethereum (0.00). However, both methods lag behind the

proposed approach in AUPR and general stability across datasets, indicating limitations in pseudo-anomaly generation for complex transactional behaviors. For BSMs DIAM and BERT4ETH, they achieve solid overall performance yet are consistently outperformed by FDL on AUROC, AUPR and FPR95 across Ethereum, BNaT and Real-CATS.

The proposed feature distribution learning approach delivers the most balanced and robust performance across three datasets, outperforming all baseline methods including discriminative, generative, OE and blockchain-specific baselines in key detection metrics and maintaining consistently low false positive rates.

5.3. Latent space visualization

To further interpret the decision mechanism of the proposed feature distribution learning method, we visualize the two-dimensional latent representations produced by the trained model. All test samples are projected into the latent space, where normal samples are shown in green and anomalous samples in red. The hypersphere center, fixed at the origin due to the structural characteristics of KAN, is marked with a blue "×". The decision boundary is visualized as an orange dashed circle whose radius corresponds to the optimized threshold learned during training.

Figure 2 presents the latent space distribution for the Ethereum dataset. Figure 3a,b shows magnified views of the anomalous and normal clusters, respectively, enabling more detailed inspection of local geometric patterns.

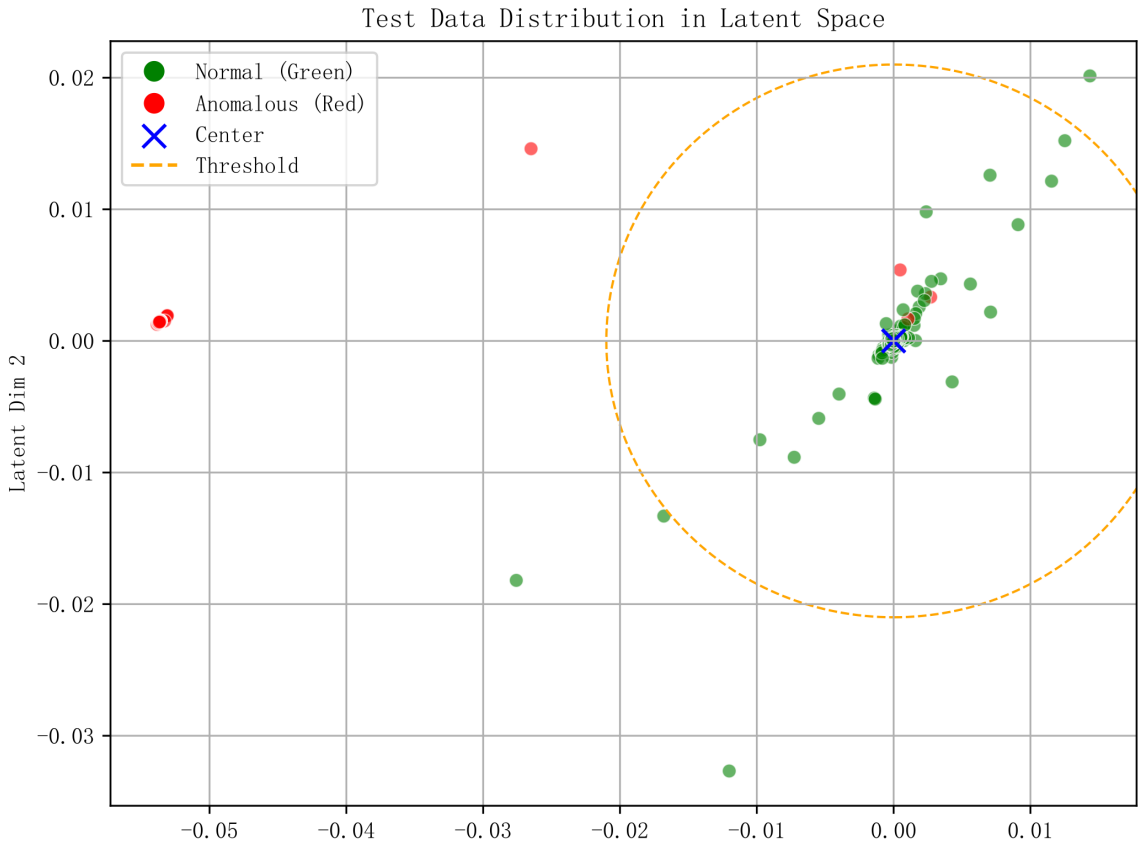

Figure 2. Visualization of test samples in the two-dimensional latent space for the Ethereum dataset.

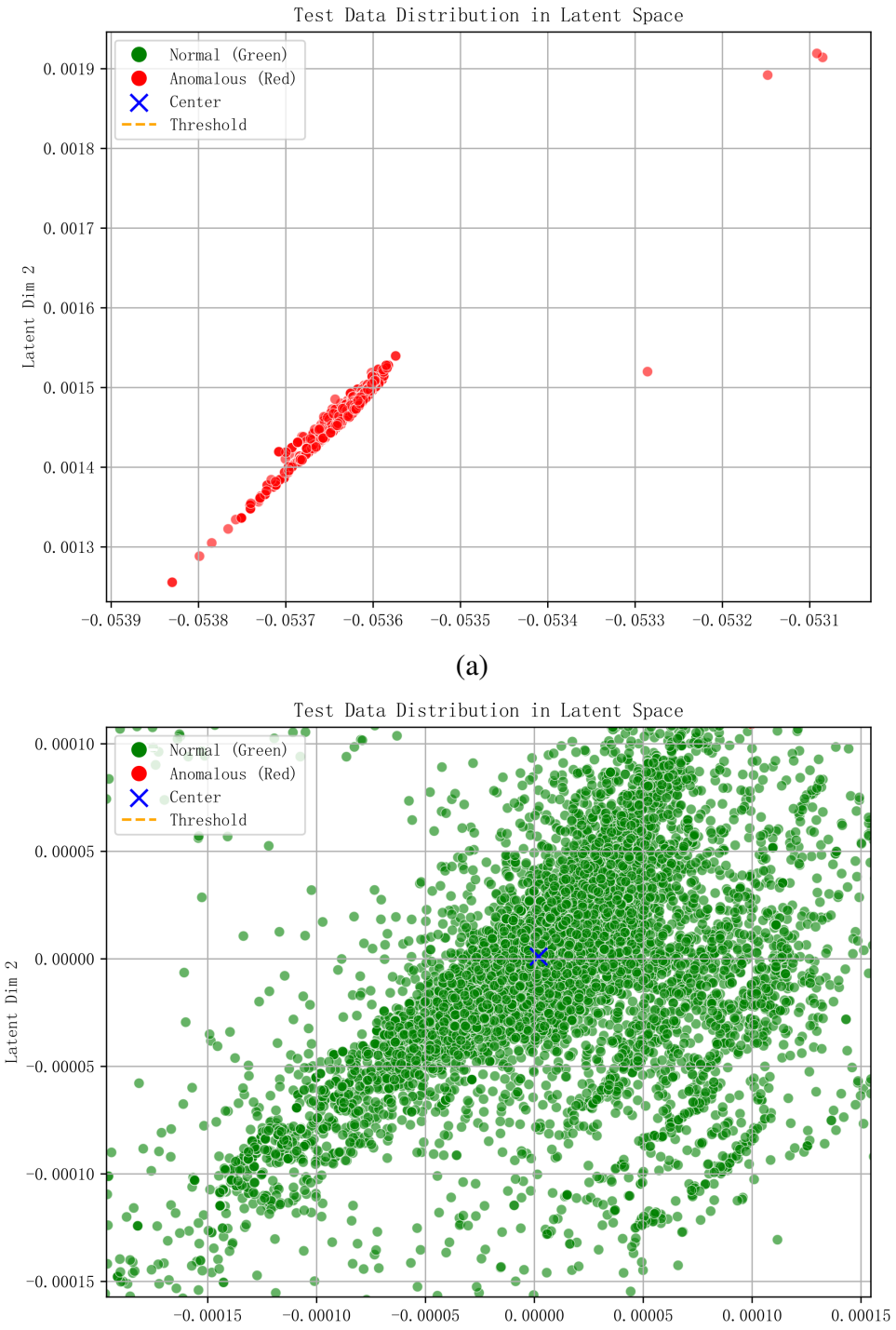

Figure 3. Magnified visualizations of local latent space regions. **(a)** Zoomed-in view of anomalous region; **(b)** Zoomed-in view of normal cluster.

As shown in Figures 2 and 3, the latent space exhibits several distinctive characteristics. Normal samples form a compact and highly concentrated cluster near the center, indicating that the model successfully encodes benign transactions into a well-structured and bounded region. In contrast, anomalous samples display clear spatial separation: most red points lie outside the decision boundary, demonstrating the effectiveness of the radius-based detection rule. A small number of anomalous points fall within the hypersphere, primarily distributed along a narrow band on the second latent dimension ($-0.01 < \text{dim}_1 < 0.01, 0 < |\text{dim}_2| < 0.005$).

These points correspond to borderline cases where malicious behavior closely mimics normal patterns, a common challenge in blockchain anomaly detection.

Despite this, the decision boundary precisely captures the variation of normal samples while excluding the vast majority of anomalies, yielding strong performance such as a recall of 0.9987 and an FPR95 of 0.0002 on the Ethereum dataset. The spatial regularity of the few misclassified anomalies also provides actionable insight for future model refinement.

The visualization indicates that the learned two-dimensional latent space provides strong discriminative capability. Normal samples are effectively constrained within a compact hypersphere, while anomalous samples exhibit systematic directional deviation, particularly along the negative axis of the first latent dimension. This geometric interpretability highlights that the proposed feature distribution learning method captures the underlying structural differences between normal and anomalous blockchain transactions.

5.4. Symbolic representation and interpretability analysis

To enhance interpretability, the proposed feature-distribution learning model is further transformed into an explicit symbolic representation using the KAN. The symbolic form reveals how each blockchain transaction feature contributes to the learned latent space and provides valuable insights for understanding model behavior. Figure 4 illustrates the learned KAN structure on the Ethereum dataset, where the top two nodes correspond to the latent dimensions (z_1, z_2) and the bottom nodes represent the 34 input features. The functions displayed inside the boxes denote the spline-based operators fitted during training, and the color intensity of the connections indicates the magnitude of the corresponding weights.

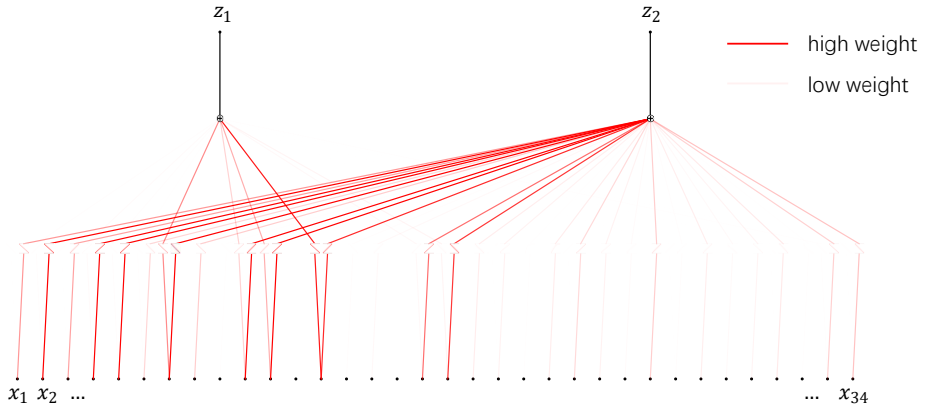

Figure 4. Symbolic KAN structure learned on the Ethereum dataset, showing the mapping from 34 input features to the two-dimensional latent space (z_1, z_2).

Using symbolic regression, the KAN mapping is converted into explicit analytical expressions that describe the transformation from the 34 input features to the latent space. For the Ethereum dataset, the resulting symbolic forms can be expressed as:

$$\begin{aligned} z_1 = &-0.0706x_7 + 0.0102x_{10} - 0.0533x_{13} + 0.0006x_4 + 0.0006x_5 + 0.0029x_{17} + 0.0002x_3 \\ &+ 0.0001(1 - 0.5076x_{14})^2 - 0.0005(-0.6x_{30} - 1)^2 + 0.0004\exp(0.16x_{25}) + \cdots \end{aligned} \tag{22}$$

$$z_2 = -0.0733x_7 + 0.0110x_{10} + 0.0017x_{13} + 0.0051x_5 + 0.0042x_4 + 0.0002x_{15} + 0.0006x_1 + 0.0002(1 - 0.2854x_{21})^2 - 0.0034\exp(0.248x_{30}) - 0.3066\sin(0.296x_{11} + 2.0002) + \cdots \quad (23)$$

Equations (22) and (23) list only the dominant terms—those with the largest absolute weights or significant nonlinear components. The remaining features, while still contributing to the model, have negligible coefficients and are omitted for clarity. This selective presentation enables a clearer interpretation of which blockchain attributes most strongly influence the latent representation.

Dominant linear features: Several input features exhibit strong linear influence on both latent dimensions. Feature x_7 (transaction time entropy) carries the largest negative coefficients in both z_1 and z_2, indicating that anomalous transactions tend to display highly irregular temporal patterns. Feature x_{10} (gas price coefficient of variation) also shows consistent positive weights, suggesting that fluctuations in gas price strongly correlate with abnormal behaviors, possibly due to adversarial attempts to manipulate transaction priority. Feature x_{13} (call depth) has a notable negative coefficient in z_1, implying that abnormal transactions often involve deeper call chains, which is consistent with complex attack execution patterns.

Nonlinear feature interactions: Feature x_{11} (address activity) contributes through a sinusoidal term $\sin(0.296x_{11} + 2.0)$, revealing periodic behavior in address-level activity that distinguishes anomalous from normal transactions.

Exponential features: Features such as x_{25} (contract complexity) and x_{30} (fund dispersion) appear through exponential terms, capturing abrupt variations associated with complex contract usage or irregular fund-flow patterns—both indicative of potential malicious activity.

The symbolic representation elucidates how the learned model combines linear, quadratic, sinusoidal, and exponential components to construct a discriminative latent space. This not only improves transparency but also provides actionable insights for blockchain security analysts. By monitoring high-impact features such as transaction time entropy, call depth, and fund-flow irregularities, practitioners can better understand and anticipate suspicious activity patterns in real-world blockchain environments.

6. Conclusion

In this paper, we proposed a distribution-oriented framework for unsupervised blockchain anomaly transaction detection. The method learns a compact latent representation of normal transactions and detects anomalies based on their deviation from the learned feature distribution, avoiding reliance on reconstruction-based objectives.

Experimental results on three blockchain datasets demonstrate that the proposed method achieves strong and consistent performance across multiple evaluation metrics, with clear separability between normal and abnormal transactions. In addition, the symbolic expressions derived from our method provide interpretable insights into how transaction features influence anomaly detection decisions.

The combination of effective detection and interpretability makes the proposed framework suitable for practical blockchain monitoring scenarios, supporting both automated analysis and informed decision-making.

Data availability statement

The code for the proposed method is publicly available at https://github.com/nr9876rn/FDL.

Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors used generative AI tools only to improve language and readability. Specifically, the authors used ChatGPT for translation and language polishing in sections other than the methodology and experiments. The authors take full responsibility for the content of the manuscript.

Acknowledgments

Funding: This work is supported by the National Natural Science Foundation of China (NSFC) under grant No. 62572099.

Authors' contribution

Conceptualization: Y.Q., C.J., Y.Z. and M.T.; methodology: Y.Q., C.J., Y.Z. and M.T.; validation: Y.Q.; investigation: Y.Q., C.J., Y.Z. and M.T.; resources:Y.Q.; writing—original draft preparation: Y.Q.; writing—review and editing: Y.Q., C.J., Y.Z. and M.T.; funding acquisition: C.J. and Y.Z.; project administration: Y.Q., C.J., Y.Z. and M.T. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

Yin Zhang holds the position of Associate Editor for *Blockchain* and has not peer reviewed or made any editorial decisions for this paper. The authors declare no conflicts of interest.

References

[1] Adam T, Babič F. Anomaly detection on distributed ledger using unsupervised machine learning. In *Proceedings of 2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, Berlin, Germany, November 11–15, 2019, pp. 1–4.

[2] Chen D, Liao Z, Chen R, Wang H, Yu C, *et al.* Privacy-preserving anomaly detection of encrypted smart contract for blockchain-based data trading. *IEEE Trans. Dependable Secure Comput.* 2024, 21(5):4510–4525.

[3] Wang J, Chen P, Xu X, Wu J, Shen M, *et al.* TSGN: transaction subgraph networks assisting phishing detection in ethereum. *IEEE Trans. Dependable Secure Comput.* 2025, 22(5):4861–4876.

[4] Elison MC, Immanuel GP. Ethereum transaction anomaly detection by integrating machine learning models and fuzzy networks for enhanced security and real-time monitoring. In *Proceedings of 2025 International Conference on Visual Analytics and Data Visualization (ICVADV)*, Tirunelveli, India, March 4–6, 2025, pp. 1120–1127.

[5] Abasi AK, Aloqaily M, Guizani M, Alatoom Z. Enhancing anomaly detection in blockchain transactions with transformer-based models. In *Proceedings of 2024 6th International Conference on Blockchain Computing and Applications (BCCA)*, Dubai, UAE, November 26–29, 2024, pp. 574–579.

[6] Huang J, Bu K, Han H, Gong B, Xiong A, *et al.* GAPLG: graph augmented with pseudolabels generation for blockchain anomaly transaction oetection. *IEEE Trans. Comput. Social Syst.* 2025, 12(6):4532–4546.

[7] Bosnyaková B, Babič F, Adam T, Biceková A. Anomaly detection in blockchain network using unsupervised learning. In *Proceedings of 2025 IEEE 23rd World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Stará Lesná, Slovakia, January 23–25, 2025, pp. 000221–000224.

[8] Henna S, Amjath M. Front-running attack detection in blockchain using conditional packing generative AI. In *Proceedings of 2024 IEEE International Conference on Big Data (BigData)*, Washington, USA, December 15–18, 2024, pp. 5585–5591.

[9] Huang D, Mu D, Yang L, Cai X. CoDetect: financial fraud detection with anomaly feature detection. *IEEE Access* 2018, 6:19161–19174.

[10] Voronov T, Raz D, Rottenstreich O. A framework for anomaly detection in blockchain networks With sketches. *IEEE/ACM Trans. Networking* 2024, 32(1):686–698.

[11] Özbilen ML, Özcan E, Keleş MB, Zeybel M, Dervişoğlu H, *et al.* Big data analytics for anomaly detection in blockchain. In *Proceedings of 2023 31st Signal Processing and Communications Applications Conference (SIU)*, Istanbul, Turkiye, July 5–8, 2023, pp. 882–885.

[12] Ding Z, Shi J, Li Q, Cao J. Effective illicit account detection on large cryptocurrency multiGraphs. In *CIKM '24: Proceedings of the 33rd ACM International Conference on Information and Knowledge Managem*, Boise, USA, October 21–25, 2023, pp. 457–466.

[13] Hu S, Zhang Z, Luo B, Lu S, He B, *et al.* BERT4ETH: a pre-trained transformer for Ethereum fraud detection. In *Proceedings of the ACM Web Conference 2023*, Austin, USA, April 30–May 4, 2023, pp. 2189–2197.

[14] Schölkopf B, Platt JC, Shawe-Taylor JC, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. *Neural Comput.* 2001, 13(7):1443–1471.

[15] Liu FT, Ting K, Zhou Z. Isolation forest. In *Proceedings of 2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, December 15–19, 2008, pp. 413–422.

[16] Ruff L, Görnitz N, Deecke L, Siddiqui SA, Vandermeulen RA, *et al.* Deep one-class classification. In *Proceedings of International Conference on Machine Learning*, Stockholm, Sweden, July 10–15, 2018, pp. 4393–4402.

[17] Li Z, Zhao Y, Botta N, Ionescu C, Hu X. COPOD: copula-based outlier detection. In *Proceedings of 2020 IEEE International Conference on Data Mining (ICDM)*, Sorrento, Italy, November 17–20, 2020, pp. 1118–1123.

[18] Li Z, Zhao Y, Hu X, Botta N, Ionescu C, *et al.* ECOD: unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Trans. Knowl. Data Eng.* 2022, 35(12):12181–12193.

[19] Shenkar T, Wolf L. Anomaly detection for tabular data with internal contrastive learning. In *Proceedings of International Conference on Learning Representations*, Online, April 25–29, 2022.

[20] Li A, Qiu C, Kloft M, Smyth P, Rudolph M, *et al.* Zero-shot anomaly detection via batch normalization. In *Proceedings of Advances in Neural Information Processing Systems*, New Orleans, USA, December 10–16, 2023, pp. 40963–40993.

[21] Perini L, Davis J. Unsupervised anomaly detection with rejection. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS 2023)*, New Orleans, USA, December 10–16, 2023, pp. 69673–69691.

[22] Mirsky Y, Doitshman T, Elovici Y, Shabtai A. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv* 2018, arXiv:1802.09089.

[23] Xu X, Li J, Yang Y, Shen F. Toward effective intrusion detection using log-cosh conditional variational autoencoder. *IEEE Internet Things J.* 2021, 8(8):6187–6196.

[24] Ishii Y, Koide S, Hayakawa K. L0-norm constrained autoencoders for unsupervised outlier detection. In *Proceedings of 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Online, May 11–14, 2020, pp. 674–687.

[25] Lappas D, Argyriou V, Makris D. Fourier transformation autoencoders for anomaly detection. In *Proceedings of ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Online, June 6–11, 2021, pp. 1475–1479.

[26] Yoon S, Noh YK, Park FC. Autoencoding under normalization constraints. In *Proceedings of International Conference on Machine Learning*, Online, July 18–24, 2021, pp. 12087–12097.

[27] Kieu T, Yang B, Guo C, Jensen CS, Zhao Y, *et al.* Robust and explainable autoencoders for unsupervised time series outlier detection. In *Proceedings of 2022 IEEE 38th International Conference on Data Engineering (ICDE)*, Kuala Lumpur, Malaysia, May 9–12, 2022, pp. 3038–3050.

[28] Su Y, Zhao Y, Sun M, Zhang S, Wen X, *et al.* Detecting outlier machine instances through Gaussian mixture variational autoencoder with one dimensional CNN. *IEEE Trans. Comput.* 2021, 71:892–905.

[29] Thimonier H, Popineau F, Rimmel A, Doan BL. Beyond individual input for deep anomaly detection on tabular data. *arXiv* 2023, arXiv:2305.15121.

[30] Ning J, Chen L, Zhou C, Wen Y. Deep active autoencoders for outlier detection. *Neural Process. Lett.* 2022 54:1399–1411.

[31] Xu H, Wang Y, Wei J, Jian S, Li Y, *et al.* Fascinating supervisory signals and where to find them: deep anomaly detection with scale learning. In *Proceedings of the 40th International Conference on Machine Learning*, Honolulu, USA, July 23–29, 2023, pp. 38655–38673.

[32] Liao J, Hou B, Dong H, Zhang H, Zhang X, *et al.* Quadratic neuron-empowered heterogeneous autoencoder for unsupervised anomaly detection. *IEEE Trans. Artif. Intell.* 2022, 5:4723–4737.

[33] KaiZhang C, Wang X, Zhang J, Li S, Zhang H, *et al.* VESC: a new variational autoencoder based model for anomaly detection. *Int. J. Mach. Learn. Cybern.* 2022, 14:683–696.

[34] Song J, Kim K, Oh J, Cho S. MEMTO: memory-guided transformer for multivariate time series anomaly detection. In *Proceedings of Advances in Neural Information Processing Systems*, New Orleans, USA, December 10–16, 2023, pp. 57947–57963.

[35] Lai CYA, Sun F, Gao Z, Lang JH, Boning D. Nominality score conditioned time series anomaly detection by point/sequential reconstruction. In *Proceedings of Advances in Neural Information Processing Systems*, New Orleans, USA, December 10–16, 2023, pp. 76637–76655.

[36] Wang Q, Ye J, Liu F, Dai Q, Kalander M, *et al.* Out-of-distribution detection with implicit outlier transformation. *arXiv* 2023, arXiv:2303.05033.

[37] Zhu J, Geng Y, Yao J, Liu T, Niu G, *et al.* Diversified outlier exposure for out-of-distribution detection via informative extrapolation. In *Proceedings of Advances in Neural Information Processing Systems*, New Orleans, USA, December 10–16, 2023, pp. 22702–22734.

[38] He R, Han Z, Lu X, Yin Y. RONF: reliable outlier synthesis under noisy feature space for out-of-distribution detection. In *Proceedings of the 30th ACM International Conference on Multimedia*, Lisbon, Portugal, October 10–14, 2022, pp. 4242–4251.

[39] Jiang Y, Cao Y, Shen W. Prototypical learning guided context-aware segmentation network for few-shot anomaly detection. *IEEE Trans. Neural Networks Learn. Syst.* 2024, 36(7):12016–12026.

[40] Mirzaei H, Jafari M, Dehbashi HR, Ansari A, Ghobadi S, *et al.* RODEO: robust outlier detection via exposing adaptive out-of-distribution samples. *arXiv* 2025, arXiv:2501.16971.

[41] Miao W, Pang G, Zheng J, Bai X. Long-tailed out-of-distribution detection via normalized outlier distribution adaptation. In *Proceedings of Advances in Neural Information Processing Systems*, Vancouver, Canada, 2024, pp. 132106–132132.

[42] Yang Y, Cheng D, Fang C, Wang Y, Jiao C, *et al.* Diffusion-based layer-wise semantic reconstruction for unsupervised out-of-distribution detection. In *Proceedings of Advances in Neural Information Processing Systems*, Vancouver, Canada, December 10–15, 2024, pp. 26846–26871.

[43] Kamran M, Rehan MM, Nisar W, Rehan MW. ARCADE—adversarially robust cost-sensitive anomaly detection in blockchain using explainable artificial intelligence. *Electronics* 2025, 14(8):1648.

[44] Elmougy Y, Liu L. Demystifying fraudulent transactions and illicit nodes in the bitcoin network for financial forensics. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Long Beach, USA, August 6–10, 2023, pp. 3979–3990.

[45] Kumar N, Sharma V, Sharma A. Artificial intelligence in fraud prevention: exploring techniques and applications challenges and opportunities. *Comput. Sci. IT Res. J.* 2024, 5(6):1505–1520.

[46] Pranto TH, Hasib KTAM, Rahman T, Haque AB, Islam AKMN, *et al.* Blockchain and machine learning for fraud detection: a privacy-preserving and adaptive incentive based approach. *IEEE Access* 2022, 10:87115–87134.

[47] Chen Y, Zhao C, Xu Y, Nie C, Zhang Y. Deep learning in financial fraud detection: innovations, challenges, and applications. *Data Sci. Manage.* 2025, 12(4):100272.

[48] Patel S, Pandey M, D R. Fraud detection in financial transactions: a machine learning approach. In *Proceedings of 2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, Chennai, India, April 4–5, 2024, pp. 1–8.

[49] Liu Z, Wang Y, Vaidya S, Ruehle F, Halverson J, *et al.* KAN: Kolmogorov–Arnold networks. In *Proceedings of International Conference on Learning Representations*, Singapore, April 24–28, 2025, pp. 70367–70413.

[50] Patel V, Pan L, Rajasegarar S. Graph deep learning based anomaly detection in ethereum blockchain network. In *Proceedings of International Conference on Network and System Security*, Melbourne, Australia, November 25–27, 2020, pp. 132–148.

[51] Khoa TV, Son DH, Hoang DT, Trung NL, Quynh TTT, *et al.* Collaborative learning for cyberattack detection in blockchain networks. *IEEE Trans. Syst. Man Cybern.: Syst.* 2024, 54(7):3920–3933.

[52] Duan C, Lei H, Shi J, Wang L. DIBA: effective transaction topology-based detection of illicit Bitcoin addresses. *ACM Trans. Knowl. Discovery Data.* 2026, 20(4):1–23.

[53] Sankaewtong K, Kim T, Tessone CJ, Ikeda Y. SoK: advances in anomaly detection techniques for cryptoasset transactions. *IEEE Access* 2025, 13:202576–202618.

[54] Song A, Seo E, Kim H. Anomaly VAE-transformer: a deep learning approach for anomaly detection in decentralized finance. *IEEE Access* 2023, 11:98115–98131.

[55] Dong H, Liu S, Tran D. Enhanced autoencoder model for robust anomaly detection in financial fraud with imbalanced data. In *Neural Information Processing*, 1st ed. Singapore: Springer Nature Singapore, 2025. pp. 184–198.

[56] Xiong A, Qiao C, Li W, Wang D, Li D, *et al.* Block-chain abnormal transaction detection method based on generative adversarial network and autoencoder. *High-Confid. Comput.* 2025, 5(4):100313.

[57] Song P. Detection and prediction of transactional anomalies in blockchain based accounting using mining behavior with Autoencoder-LSTM. *Forensic Sci. Int. Digit. Investig.* 2026, 58:302138.