Deep Isolation Forest for Anomaly Detection

Hongzuo Xu ⓘ, Guansong Pang ⓘ, *Member, IEEE*, Yijie Wang ⓘ, and Yongjun Wang ⓘ

Abstract—**Isolation forest (iForest) has been emerging as arguably the most popular anomaly detector in recent years due to its general effectiveness across different benchmarks and strong scalability. Nevertheless, its linear axis-parallel isolation method often leads to (i) failure in detecting hard anomalies that are difficult to isolate in high-dimensional/non-linear-separable data space, and (ii) notorious algorithmic bias that assigns unexpectedly lower anomaly scores to artefact regions. These issues contribute to high false negative errors. Several iForest extensions are introduced, but they essentially still employ shallow, linear data partition, restricting their power in isolating true anomalies. Therefore, this paper proposes deep isolation forest. We introduce a new representation scheme that utilises casually initialised neural networks to map original data into random representation ensembles, where random axis-parallel cuts are subsequently applied to perform the data partition. This representation scheme facilitates high freedom of the partition in the original data space (equivalent to non-linear partition on subspaces of varying sizes), encouraging a unique synergy between random representations and random partition-based isolation. Extensive experiments show that our model achieves significant improvement over state-of-the-art isolation-based methods and deep detectors on tabular, graph and time series datasets; our model also inherits desired scalability from iForest.**

Index Terms—**Anomaly detection, isolation forest, deep representation, ensemble learning.**

I. INTRODUCTION

A NOMALY detection has broad applications in various domains, such as detection of insurance fraud and financial crime, surveillance of complex systems like data centres and

Hongzuo Xu and Yijie Wang are with the Science and Technology on Parallel and Distributed Processing Laboratory, College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: xuhongzuo13@nudt.edu.cn; wangyijie@nudt.edu.cn).

Yongjun Wang is with the College of Computer, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: wangyongjun@nudt.edu.cn).

Guansong Pang is with the School of Computing and Information Systems, Singapore Management University, Singapore 178902 (e-mail: pangguansong@gmail.com).
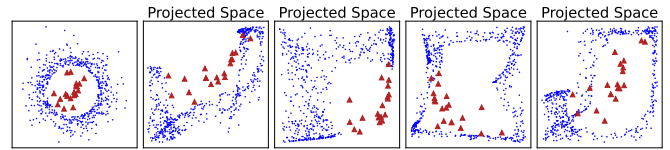
Fig. 1. Illustration of the hard anomaly challenge. The left figure is a synthetic dataset, where red triangles are anomalies and blue points are normal samples. The following figures are projected spaces (i.e., random representations) created by our method. Linear data partition (in either vertical/horizontal or oblique forms) used in existing methods cannot effectively isolate these hard anomalies in the original data space. By contrast, these anomalies are possible to be exposed in newly created spaces.

spacecraft, and identification of attacks and potential threats in cyberspace. Given these important applications, this task has been a popular research topic for decades, and numerous anomaly detection approaches have been introduced [1], [2].

In recent years, isolation forest (iForest) [3] has been emerging as arguably the most popular anomaly detector due to its general effectiveness across different benchmarks and strong scalability. Compared to many existing methods such as distance/density-based methods, iForest better captures the key essence of anomalies, i.e., "few and different". It does not introduce extra assumptions of data characteristics, thus showing consistently effective performance across diverse datasets. Also, iForest is with linear time complexity, which is often a very appealing advantage in many industrial applications when there are large-scale data and strict requirements of time efficiency. Concretely, iForest uses an ensemble of isolation trees (iTree), in which each iTree is grown by iteratively branching. Leaf nodes are built by using random cuts in the values of randomly selected features until the data objects are isolated. Data abnormality is estimated according to the average depth traversing from the root node to the isolated leaf node in these iTrees.

Nevertheless, an explicit major issue is that it cannot handle hard anomalies (e.g., the anomalies that can be only isolated in higher-order subspaces by looking into the combination of multiple features) because it treats all features separately and considers only one feature per isolation operation. Fig. 1 exemplifies this issue with a simple toy example. Anomalies (represented as red triangles) are surrounded by ring-shaped normal samples, which cannot be isolated by either x-axis or y-axis slicing cuts. Although these anomalies might be finally isolated by multiple cuts, it results in indistinguishable isolation depth in iTrees compared to normal data. The failure of recalling these anomalies induces high false negative errors. Given that anomalies often contain critical information related to potential accidents or faults, those false negatives may cause serious consequences. Therefore, this issue has been a major
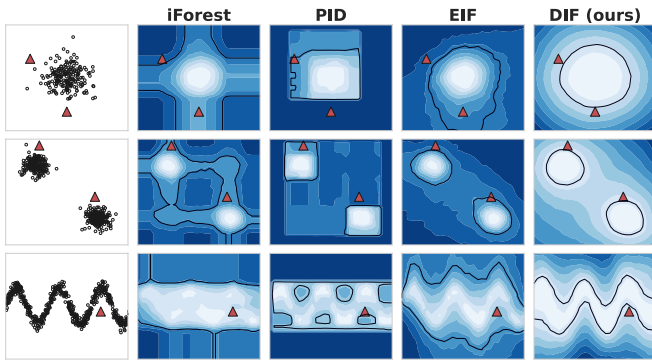
Fig. 2. Illustration of the algorithmic bias problem. The three figures on the far left are three synthetic datasets, and the following panels of each dataset are the anomaly score maps produced by iForest and its two state-of-the-art extensions – PID and EIF – and our method DIF. Score maps indicate anomaly score distribution in the full data space (deeper colour indicates higher abnormality). Black contour lines in each score map denote the 99th percentile of the anomaly scores of the original data, which can be seen as the boundary of data normality predicted by each anomaly detector. Red triangles represent possible anomalies. There are some artefact regions in the score maps of iForest and its two extensions, and these methods may fail to identify the anomalies since these anomalies are assigned similarly low anomaly scores as the presented normal samples (they are included in the predicted normal areas). By contrast, DIF produces more accurate and smooth score maps, and DIF can successfully assign a high abnormal degree to anomalies compared to those original data samples.

bottleneck hindering the performance of iForest, particularly on datasets with high-dimensional/non-linear-separable data spaces.

Another inherent imperfection of iForest is that it assigns unexpectedly low anomaly scores to artefacts introduced by the algorithm itself, which is revealed as the "ghost region" problem in [4]. To clearly demonstrate this issue, we visualise the data distribution of three 2-D synthetic datasets used in [4] and anomaly score maps generated by iForest in the first two columns in Fig. 2, respectively. As can be seen from all three anomaly score maps, iForest assigns clearly lower anomaly scores to some artefact regions, i.e., the four rectangular areas centred around the presented data objects in the single-blob dataset (1st row), the upper right and bottom left clique areas in the two-blob dataset (2nd row), and the vertical rectangular areas along the sinusoid in the sinusoidal dataset (3rd row). However, these artefact regions are similar to other regions that contain no samples or have similar radial distances to the presented data objects. The unusually lower anomaly scores are due to the intrinsic algorithmic bias of iForest, i.e., only axis-parallel partitions are admitted in iTree construction. Instead, an expected anomaly score map should smoothly approximate circular contour lines w.r.t. the density of the presented data. This problem also leads to false negative errors, i.e., iForest may fail to detect possible anomalies in these "ghost regions" such as the red triangles in Fig. 2 as they are assigned similarly low anomaly scores as some of the presented normal objects and included in the normal areas.

There have been a number of extensions of iForest introduced over the years [4], [5], [6], [7], [8], [9]. These extensions attempt to devise more advanced isolation methods by (i) using novel hierarchical criteria to selectively pick splitting thresholds and/or dimensions [6], [8], [9] or (ii) using optimal/random

hyper-planes [4], [5], [7]. These improvements successfully contribute to better detection performance, but their isolation methods are still confined by linear partition. Instead, data partition in the isolation process is expected to form non-linearly arbitrary shapes in the data space to handle hard anomalies in complicated datasets and eliminate the algorithmic bias in derived anomaly scores. We show two state-of-the-art extensions (PID [6] and EIF [4]) in Fig. 2. PID [6] selectively chooses splitting dimensions and uses similar axis-parallel isolation operations. Thus, its score maps also encounter the same algorithmic bias problem. EIF [4] works better due to the use of hyper-planes with random slopes and intercepts as branching criteria, eliminating the limits of axis-parallel partitions to some extent. However, the partitions still operate linearly, which works ineffectively on challenging datasets where only non-linear partitions are useful in isolating anomalies from the other data. This problem is reflected by the rather wrinkled anomaly score contours in all three datasets in Fig. 2.

Therefore, the key to these limitations becomes very clear now, i.e., a step further is required to liberate the isolation method from the underlying linear constraints. To this end, this paper proposes a novel extension of iForest named DEEP ISOLATION FOREST (DIF for short). The key idea in DIF is to harness the strong representation power of neural networks to map the original data into a group of new data spaces, and non-linear isolation can be easily achieved by performing simple axis-parallel partitions upon these newly created data spaces (equivalent to non-linear partition on subspaces of varying sizes in the original data space). Specifically, we propose a novel representation scheme – random representation ensemble – produced by optimisation-free deep neural networks. These networks are only casually initialised and do not involve any optimisation or training process. DIF then simply utilises random axis-parallel cuts upon these representations. The randomness in our representation scheme allows high freedom of the partition in the original data space, encouraging a unique synergy effect between random representations and random partition-based isolation. This facilitates effective isolation of hard anomalies and eliminates the algorithmic bias, thus significantly enhancing the detection performance. Fig. 1 shows four random representations created by DIF, where those hard anomalies are possible to be exposed and easily isolated by using a few axis-parallel cuts. Besides, as shown in Fig. 2, DIF accurately assigns anomaly scores to the presented data objects and has a smooth estimation of the anomaly scores in the other regions, alleviating the aforementioned algorithmic bias problem.

Further, in our specific implementation of DIF, we propose the Computation-Efficient Representation Ensemble method (CERE) and the Deviation-Enhanced Anomaly Scoring function (DEAS). With the help of CERE, DIF can efficiently produce the representation ensemble by taking full advantage of parallel accelerators in the mini-batch calculation, largely eliminating the computational overhead. DEAS leverages the hidden quantitative information enclosed in the mapped dense representations along with qualitative comparisons. This additional information enables a more accurate assessment of the isolation difficulty of data objects, leading to a better anomaly scoring function.

Our main contributions are summarised as follows:

- We propose the DEEP ISOLATION FOREST (DIF) method. A new isolation method is introduced, which enables non-linear partition on subspaces of varying sizes, offering a more effective anomaly isolation solution than the current methods that are capable of linear isolation only. The resulting approach can better handle hard anomalies in complex datasets with high-dimensional/non-linear-separable data spaces and eliminate the algorithmic bias. We also show that DIF is a high-level generalisation of iForest and its very recent extension EIF.
- We propose a novel representation scheme, the random representation ensemble, in which only casually initialised neural networks are required. This representation scheme facilitates high freedom of partitions in the original data space. The unique synergy between random representations and random partition-based isolation brings excellent randomness and desired diversity into the overall ensemble-based abnormality estimation process.
- We propose the representation ensemble method CERE to ensure that DIF can inherit good scalability from iForest. The anomaly scoring function DEAS is introduced to leverage additional quantitative information enclosed in mapped dense representations, enhancing the quality of anomaly scoring of DIF.
- DIF offers a data-type-agnostic anomaly detection solution. We show that DIF is versatile to detect anomalies in different types of data by simply plugging in corresponding randomly initialised neural networks in the feature mapping.

Extensive experiments on a large collection of real-world datasets, including not only tabular data but graph and time series data, show that: (i) DIF significantly outperforms iForest and its state-of-the-art extensions (Section VI-B); (ii) DIF also achieves remarkable improvement compared to the ensemble of advanced deep anomaly detectors (Section VI-B); (iii) DIF has desired scalability on high-dimensional, large-scale datasets (Section VI-C) and presents good robustness to anomaly contamination (Section VI-D); (iv) the significance of the synergy between random representations and random partition-based isolation is justified by comparing to several alternatives (Section VI-E); and (v) the contribution of CERE and DEAS is separately verified in our ablation study (Section VI-F).

II. RELATED WORK

A. Anomaly Detection

Anomaly detection has been intensively studied in the last decades by using different data characteristics like distance, density, clustering membership, or probability [1]. Recent studies [10], [11], [12], [13] devise deep anomaly detection models based on the representation learning, while some other methods use concepts like reconstruction [14] or deviation [15] in the deep learning framework as scoring function. Surveys and comparative studies can be found in [2], [16], [17]. Also, some practical anomaly detection tools, e.g., [19], [20], [21], [22], are developed to facilitate the use of anomaly detection models in real-world applications.

B. Isolation Forest and its Extensions

iForest [3] is a very popular anomaly detection method that identifies anomalies according to the isolating difficulty in the data space. Although iForest has shown effective performance on several benchmarks, a number of its extensions have been proposed, attempting to fix its notorious vulnerabilities and obtain better detection performance.

The mainstream of these extensions focuses on devising more effective isolation methods. SCIF [5] introduces a non-axis-parallel way as branching criteria, i.e., instead of selecting one feature during splitting, an optimal slicing hyper-plane is used. EIF [4] also uses hyper-planes but with random slopes and intercepts. Another work [7] further fixes the empty branching problem of EIF by choosing splitting thresholds from the range of projected values onto the slope direction. PID [6] selectively picks dimensions that have greater variance and selects split points according to the sparsity of the partition branches. The literature [9] proposes a probability-based method to find better split values than random splitting. These extensions normally achieve better performance by introducing non-axis-parallel and/or heuristic partition. However, the major problem is that they still rely on *linear isolation operations*, which means it is also hard to handle complicated data that require non-linear partitions as the isolation method. Also, they suffer from the aforementioned artefact problem due to the implicit algorithmic bias hidden in the isolation process. There are also some extensions employing the nearest neighbour information into isolation process, e.g., LeSiNN [23] and iNNE [24]. The literature [25] further uses locally-sensitive hashing to extend the isolation mechanism to any distance measures and data types. These distance-based methods introduce an extra assumption, i.e., anomalies are far from other data objects. However, this assumption does not always hold since clustered anomalies are also very close to their adjacent neighbours [5]. The detection performance is also sensitive to the choice of distance metrics.

Another angle is to enhance the scoring method. The literature [26] introduces a path-weighted scoring method and a probability-based aggregation function. PID [6] redefines the scoring function according to the sparsity rather than depth in the tree. Similarly, these extensions are still vulnerable due to their linear isolation methods.

C. Deep Ensembles

Deep ensemble [27], a simple framework that combines prediction results of a group of independently trained networks together, has garnered much interest. It can improve prediction accuracy and provide uncertainty estimation in a simple framework without modifying the original working pipeline. Similarly to other ensemble-based approaches, the quality of deep ensembles also largely hinges on the diversity of its members. Besides, albeit simple, deep ensembles still induce considerably larger computational costs. Thus, many related studies attempt to address these two key limitations, e.g., [28] uses repulsive terms to ensure individual diversity, [29] increases diversity by adversarially preventing features from being conditionally predictable from each other, and [30] proposes a distilled model that can absorb as much function diversity inside the ensemble

TABLE I
THE MAIN NOTATIONS USED IN THE PAPER

Format	Notations – Descriptions
Calligraphic fonts	$\mathcal{D}$ – datasets, $\mathcal{X}$ – representations, $\mathcal{T}$ – the set of iTrees, $\mathcal{P}$ – data pools in iTree nodes
Script typeface	$\mathscr{F}$ – anomaly scoring function, $\mathscr{G}$ – representation function
Bold lowercase letters	$\boldsymbol{o}$ – original data objects, $\boldsymbol{x}$ – representation vectors, $\boldsymbol{p}, \boldsymbol{q}$ – random vectors used in CERE
Bold uppercase letters	$\mathbf{W}$ – weight matrices of neural networks, $\mathbf{P}, \mathbf{Q}$ – matrices used in CERE
Others	r – number of representations, t – number of iTrees per representation, T – total number of iTrees, $p(\cdot\|\cdot)$ – traversed node path, $g(\cdot\|\cdot)$ – averaged deviation degree

as possible. In addition, this framework inspires related studies on anomaly detection. The literature [31], [32] combines a group of autoencoders and uses the median of reconstruction errors as anomaly scores. To improve member diversity, these autoencoders are sparsely connected.

Our work also involves the process of integrating neural networks. From the deep ensemble aspect, our work can effectively tackle the above two key issues. The diversity between ensemble members can be guaranteed and the calculation efficiency can be well maintained since only initialised networks are required. Our work may also provides valuable insights into the deep ensemble research line.

III. PROBLEM STATEMENT AND NOTATIONS

Let $\mathcal{D} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_N\}$ be a dataset with N data objects, anomaly detection is to give a scoring function $f : \mathcal{D} \mapsto \mathbb{R}^N$ that estimates the abnormality of each data object. Different from many existing studies that only focus on an individual data type, we do not restrict the type of data objects in this work, which means they can be multi-dimensional vectors, time series data, or graphs. Throughout the paper, we use calligraphic fonts for sets, script typeface for functions, bold lowercase letters for vectors, bold uppercase letters to denote matrices. Table I summarises main notations.

IV. PRELIMINARIES: ISOLATION FOREST

For the sake of clarity, we recall the basic procedure of isolation forest (iForest) [3]. A basic structure named isolation tree (iTree for short) is proposed. iTree τ is essentially a binary tree, and each node in the tree corresponds to a pool of data objects. A subset containing n data objects is used as the data pool of the root node, which is randomly subsampled from the whole dataset. iTree τ grows by recursively isolating data objects in the leaf node (i.e., a disjoint partition of data objects into two child nodes) in a top-down fashion until remaining one data object in the node or reaching the maximum depth limit. iForest uses a simple isolation method that performs a comparison between the j-th dimension of the data object $\boldsymbol{o}^{(j)}$ and a splitting value η as the branching criterion of each data object $\boldsymbol{o}$, where j and η respectively denote a randomly selected feature index and a split value within the range of available values of the j-th feature. Each data object $\boldsymbol{o}$ has a traversing path

$p(\boldsymbol{o}|\tau)$ in the iTree τ, and length of the traversed path, $|p(\boldsymbol{o}|\tau)|$, can be naturally viewed as an indication of the abnormal degree of $\boldsymbol{o}$ (anomalies are often easier to be isolated with shorter path length). iForest constructs a forest of T iTrees $\mathcal{T} = \{\tau_i\}_{i=1}^T$. The anomaly score of data object $\boldsymbol{o}$ is calculated based on its averaged path length $\mathbb{E}_{\tau_i \in \mathcal{T}}(|p(\boldsymbol{o}|\tau_i)|)$ over all of the iTrees in the forest $\mathcal{T}$, i.e., $\mathscr{F}_{\text{iForest}}(\boldsymbol{o}|\mathcal{T}) = 2^{-\frac{\mathbb{E}_{\tau_i \in \mathcal{T}} \frac{|p(\boldsymbol{o}|\tau_i)|}{C(T)}}}$, where $C(T)$ is a normalising factor.

iForest uses a linear axis-parallel isolation method that only considers one dimension each time, and existing extension work introduces hyper-plane-based isolation that involves multiple dimensions, but similarly, only *linear partition is admitted*. These current isolation methods are limited to effectively handle hard anomalies that cannot be isolated using linear partitions on individual features or simple combinations of multiple features. Additionally, these existing methods generally suffer from the algorithmic bias brought by constraints hidden in their isolation strategies.

V. DEEP ISOLATION FOREST

We require a new isolation method that is unleashed from the above constraints so that it can effectively isolate those hard anomalies and avoid algorithmic bias. To this end, we introduce the DEEP ISOLATION FOREST (DIF) method. In a nutshell, DIF constructs an ensemble of representations derived from deep neural networks, and simple axis-parallel isolation is operated upon new data spaces to build iTrees in the forest. Instead of following the deep ensemble framework that combines independently trained neural networks, we use an ensemble of random representations produced by optimisation-free neural networks that only require simple casual initialisation. This new representation scheme allows high freedom of the partition in the original data space. Each feature in the newly projected representation space is based on non-linear interactions among a number of original features, meaning that axis-parallel partitions of new space can form effective non-linear cuts in the original space. This way successfully liberates the slicing cuts from current linear constraints. Those hard anomalies that cannot be easily isolated in the original data space are possible to be exposed in representation spaces, and they can be isolated via fewer cuts and lead to clearer abnormality. Meanwhile, a unique synergy between random representations and random partition-based isolation can facilitate the overall ensemble-based abnormality estimation.

A. Formulation of DIF

DIF first produces the random representation ensemble via optimisation-free neural networks, which is defined as

$$\mathscr{G}(\mathcal{D}) = \left\{ \mathcal{X}_u \subset \mathbb{R}^d \,\middle|\, \mathcal{X}_u = \phi_u(\mathcal{D}; \theta_u) \right\}_{u=1}^r, \tag{1}$$

where r is the ensemble size, $\phi_u : \mathcal{D} \mapsto \mathbb{R}^d$ is the network that maps original data into new d-dimensional spaces, and the network weights in θ_u are randomly initialised. Each representation is assigned with t iTrees, and a forest $\mathcal{T} = \{\tau_i\}_{i=1}^T$ containing $T = r \times t$ iTrees is constructed. iTree τ_i of $\mathcal{X}$ is

initialised by a root node with a set of projected data $\mathcal{P}_1 \subset \mathcal{X}$. The k-th node with the data pool $\mathcal{P}_k$ is branched into two leaf nodes with disjoint subsets, i.e., $\mathcal{P}_{2k} = \{\boldsymbol{x} | \boldsymbol{x}^{(j_k)} \leq \eta_k, \boldsymbol{x} \in \mathcal{P}_k\}$ and $\mathcal{P}_{2k+1} = \{\boldsymbol{x} | \boldsymbol{x}^{(j_k)} > \eta_k, \boldsymbol{x} \in \mathcal{P}_k\}$, where j_k is selected uniformly at random among all the dimensions of the newly created data space $\{1, \ldots, d\}$, $\boldsymbol{x}^{(j_k)}$ is the j_k-th dimension of the projected data object, and η_k is a split value within the range of $\{\boldsymbol{x}^{(j_k)} | \boldsymbol{x} \in \mathcal{P}_k\}$.

After constructing $\mathcal{T}$, the abnormality of a data object $\boldsymbol{o}$ is evaluated by the isolation difficulty in each iTree of the forest $\mathcal{T}$. The scoring function is defined as

$$\mathscr{F}(\boldsymbol{o} | \mathcal{T}) = \Omega_{\tau_i \sim \mathcal{T}} I(\boldsymbol{o} | \tau_i), \qquad (2)$$

where $I(\boldsymbol{o} | \tau_i)$ denotes a function to measure the isolation difficulty in iTree τ_i, and Ω denotes an integration function.

B. Implementation of DIF

There are two main components in DIF, i.e., random representation ensemble function $\mathscr{G}$ and isolation-based anomaly scoring function $\mathscr{F}$. To improve the time efficiency of representation function $\mathscr{G}$, we propose Computation-Efficient deep Representation Ensemble method (CERE), in which all the ensemble members can be computed simultaneously in a given mini-batch. To improve the accuracy of anomaly scoring, we propose Deviation-Enhanced Anomaly Scoring function (DEAS) by leveraging the hidden quantitative information enclosed in the projected dense representations along with qualitative comparisons.

1) CERE: Computation-Efficient Deep Representation Ensemble Method: Successively feeding raw data into r independent networks in Eq. (1) can induce a considerably high memory and time overhead. To inherit outstanding scalability of the original iForest, we introduce CERE to efficiently implement the representation ensemble function $\mathscr{G}(\mathcal{D})$.

Let $\mathbf{W} \in \mathbb{R}^{m \times n}$ be a weight matrix of a neural network layer, then following [33], we use a tuple of small random vectors $\boldsymbol{p}_i \in \mathbb{R}^m$ and $\boldsymbol{q}_i \in \mathbb{R}^n$ to yield a rank-one matrix via multiplication, which is used to derive the full weight matrices of each ensemble member. Formally, based on a base weight matrix $\mathbf{W}_0$, the weight matrix $\mathbf{W}_i$ of the i-th ensemble member is generated as

$$\mathbf{W}_i = \mathbf{W}_0 \circ (\boldsymbol{p}_i \boldsymbol{q}_i^\top), \qquad (3)$$

where $\circ$ denotes the Hadamard product.

The mapping process of incoming neurons $\boldsymbol{x} \in \mathbb{R}^m$ and the weight $\mathbf{W}_i$ can be further derived as follows:

$$\begin{aligned} \mathbf{W}_i^\top \boldsymbol{x} &= (\mathbf{W}_0 \circ \boldsymbol{p}_i \boldsymbol{q}_i^\top)^\top \boldsymbol{x} \\ &= \mathbf{W}_0^\top (\boldsymbol{x} \circ \boldsymbol{p}_i) \circ \boldsymbol{q}_i. \end{aligned} \qquad (4)$$

Given r tuples of weight vectors $\{\langle \boldsymbol{p}_i, \boldsymbol{q}_i \rangle\}_{i=1}^r$ and a mini-batch of data $\mathcal{X} \in \mathbb{R}^{b \times m}$ with mini-batch size b and dimension m, the ensemble of mapped results $\{\mathcal{X}\mathbf{W}_1, \ldots, \mathcal{X}\mathbf{W}_r\}$ can be calculated via

$$\begin{bmatrix} \mathcal{X}\mathbf{W}_1 \\ \mathcal{X}\mathbf{W}_2 \\ \cdots \\ \mathcal{X}\mathbf{W}_r \end{bmatrix} = \left(\left(\begin{bmatrix} \mathcal{X} \\ \mathcal{X} \\ \cdots \\ \mathcal{X} \end{bmatrix} \circ \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \cdots \\ \mathbf{P}_r \end{bmatrix} \right) \mathbf{W}_0 \right) \circ \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \cdots \\ \mathbf{Q}_r \end{bmatrix}, \qquad (5)$$

where each row in $\mathbf{P}_i$ and $\mathbf{Q}_i$ is duplicated $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$. Let $\mathbf{1}_b$ be an all-one vector with size b, $\mathbf{P}_i$ and $\mathbf{Q}_i$ are obtained via $\mathbf{P}_i = \mathbf{1}_b \boldsymbol{p}_i^\top$ and $\mathbf{Q}_i = \mathbf{1}_b \boldsymbol{q}_i^\top$.

The above vectorisation derivation allows the deep representation ensemble process in DIF to be efficiently calculated. With the help of CERE, the time complexity of the ensemble process is similar to the feed-forward process of a single neural network since all the ensemble members can be computed simultaneously in a given mini-batch. Eq. (5) requires additional Hadamard product steps, but this operation is very cheap compared to matrix multiplication. One possible limitation is the batch size in Eq. (5). A mini-batch of $r \times t$ objects is simultaneously computed, which is larger than conventional settings. However, as computation within a mini-batch is automatically parallelisable, increasing the batch size incurs almost no time overhead. As for memory cost, this process is also feasible in a typical device because DIF does not involve optimisation, i.e., gradients are not calculated and saved. For example, a dataset with 10,000 features costs about 3 GB of memory when using the recommended ensemble size $r = 50$ and the batch size $b = 64$.

Let Φ be the L-layer neural network using the newly-defined feed-forward step in Eq. (5). The ensemble of representations can be directly generated as

$$\mathscr{G}_{\text{CERE}}(\mathcal{D}) = \Phi(\mathcal{D}; \Theta) = \{\mathcal{X}_i \subset \mathbb{R}^d\}_{i=1}^r, \qquad (6)$$

where $\Theta = \{\mathbf{W}_l, \{\boldsymbol{p}_{(l,i)}\}_{i=1}^r, \{\boldsymbol{q}_{(l,i)}\}_{i=1}^r\}_{l=1}^L$. Note that other operations like activation or pooling and some layers that do not use a weight matrix are processed sequentially.

2) DEAS: Deviation-Enhanced Anomaly Scoring Function: We further introduce a new anomaly scoring function DEAS. Recall that the standard anomaly scoring process in iForest only uses the length of the traversed path, i.e., all nodes are considered to have the same importance. The path length only provides limited information, which may not sufficiently delineate the isolation difficulty of data objects. Except for qualitative comparison in each node, additional quantitative information is readily available to be leveraged, such as relation between the feature values of data objects and the branching threshold.

Motivated by this, we utilise the deviation degree of the feature value to the branching threshold as additional weighting information to further improve the measurement of isolation difficulty. These deviation degrees are important indicators to the isolation difficulty because the feature values in newly created data spaces are typically densely distributed and these deviations reflect the local density in the projected space., e.g., a small deviation value indicates that the slicing cut is on a dense region and thus is hard to isolate the data object. Specifically, let $\boldsymbol{x}_u$ be the corresponding representation of a data object $\boldsymbol{o}$ in an iTree τ_i. $p(\boldsymbol{x}_u | \tau_i) = \{1, \ldots, K\}$ is its traversed node path. We define

the averaged deviation degree of $\boldsymbol{x}_u$ in τ_i as

$$g(\boldsymbol{x}_u|\tau_i) = \frac{1}{|p(\boldsymbol{x}_u|\tau_i)|} \sum_{k \in p(\boldsymbol{x}_u|\tau_i)} |\boldsymbol{x}_u^{(j_k)} - \eta_k|. \qquad (7)$$

We further combine the path length $|p(\boldsymbol{x}_u|\tau_i)|$ as in iForest and the deviation measure in Eq. (7) to specify the function in Eq. (2) by defining our deviation-enhanced isolation anomaly scoring function as:

$$\mathscr{F}_{\text{DEAS}}(\boldsymbol{o}|\mathcal{T}) = 2^{-\mathbb{E}_{\tau_i \in \mathcal{T}} \frac{|p(\boldsymbol{x}_u|\tau_i)|}{C(\mathcal{T})}} \times \mathbb{E}_{\tau_i \in \mathcal{T}} \left(g(\boldsymbol{x}_u|\tau_i)\right), \qquad (8)$$

where the first term is the averaged depth used as anomaly scores in iForest and the second term is the deviation-based anomaly score we introduce.

C. Algorithm of DIF

Algorithm 1 presents the procedure of the construction of deep isolation trees $\mathcal{T}$. Step (2) prepares r random representations, and t isolation trees $\{\tau_i\}_{i=1}^t$ are built upon each representation in Steps (4–15). For each τ_i, a subset of transformed data objects $\mathcal{P}_1$ is first randomly subsampled in Step (5) to initialise the root node. Each leaf node P_k is then iteratively split by using a comparison branching criteria based on a randomly selected representation dimension j_k and a split point η_k in Steps (6–13).

We report the anomaly scoring procedure in Algorithm 2. Data object $\boldsymbol{o}$ is transformed to vectorised representations $\{\boldsymbol{x}\}_{u=1}^r$ in Step (1) After the initialisation in Step (4), the data object traverses each tree τ_i by the criteria of each node and reaches the final node, during which the traverse path $p(\boldsymbol{x}_u|\tau_i)$ and the accumulated difference β are recorded in Steps (5–12). The path length $|p(\boldsymbol{x}_u|\tau_i)|$ and the deviation $g(\boldsymbol{x}_u|\tau_i)$ in iTree τ_i are calculated in Step (13). The anomaly score of $\boldsymbol{o}$ is calculated and returned in Steps (16–17).

D. Theoretical Analysis

1) Time Complexity Analysis: We first analyse the time complexity for the production of the random representation ensemble via CERE (i.e., Step 2 in Algorithm 1). Let the input data $\mathcal{D}$ be a tabular dataset with size $N \times D$. Multi-layer perceptron network is used for ϕ. The used network ϕ comprises L layers, and the l-th layer is with d_l hidden units, the representation dimension is d. We use CERE to implement the ensemble with r members within each mini-batch, and thus the whole feed-forward computation induces $O(r \times N \times (Dd_1 + d_l d + \sum_{l=1}^{L-1} d_l d_{l+1}))$. Only feed-forward steps are required in DIF, and the number of hidden units and representation dimension is generally small. Thus, this process is linear w.r.t. both data size and dimensionality, which does not introduce much extra computational overhead than the original iForest. In terms of the subsequent process of the iTree construction, given the depth limit J, we have the maximum 2^{J-1} splits (Steps (8–11)) during the growth of each iTree. For a node with n samples, each split takes $O(n)$ complexity in determining the maximum and minimum value of the selected dimension and the assigning process. The overall process induces $O(2^{J-1} \times n \times r \times t)$. J and n often use fixed small values (typically 8 and 256 respectively). Therefore, the

---

Algorithm 1: Construction of Deep Isolation Trees.

  Input: $\mathcal{D}$ - input dataset
  Output: $\mathcal{T}$ - forest of deep isolation trees
1: Initialise $\mathcal{T} \leftarrow \varnothing$
2: Generate representations $\{\mathcal{X}_u\}_{u=1}^r$ via $\mathscr{G}_{\text{CERE}}$
3: **for** $u = 1$ to r **do**
4:   **for** $i = 1$ to t **do**
5:     Initialise an isolation tree τ_i by setting the root node using $\mathcal{P}_1 \subseteq \mathcal{X}_u, |\mathcal{P}_1| = n$
6:     **while** $\mathcal{P}_k$ is a leaf node of tree τ_i **do**
7:       **if** $|\mathcal{P}_k| > 1$ and the depth is smaller than J **then**
8:         Randomly select a dimension $j_k \in \{1, \ldots, d\}$
9:         Randomly select a split point η_k between the *max* and *min* values of dimension j_k in $\mathcal{P}_k$
10:         $\mathcal{P}_{2k} \leftarrow \{\boldsymbol{x}|\boldsymbol{x}^{(j_k)} \leq \eta_k, \boldsymbol{x} \in \mathcal{P}_k\}$
11:         $\mathcal{P}_{2k+1} \leftarrow \{\boldsymbol{x}|\boldsymbol{x}^{(j_k)} > \eta_k, \boldsymbol{x} \in \mathcal{P}_k\}$
12:       **end if**
13:     **end while**
14:     $\mathcal{T} \leftarrow \mathcal{T} \cup \tau_i$
15:   **end for**
16: **end for**
17: **return** $\mathcal{T}$

Algorithm 2: Deviation-Enhanced Anomaly Scoring.

  Input: $\boldsymbol{o}$ - data object, $\mathcal{T}$ - set of deep isolation trees
  Output: anomaly score $\mathscr{F}_{\text{DEAS}}(\boldsymbol{o}|\mathcal{T})$
1: Generate representations $\{\boldsymbol{x}_u\}_{u=i}^r$ via $\mathscr{G}_{\text{CERE}}$
2: **for** $u = 1$ to r **do**
3:   **for** $i = 1$ to t **do**
4:     Initialise $k \leftarrow 1, \beta \leftarrow 0, p(\boldsymbol{x}_u|\tau_i) \leftarrow \varnothing$
5:     **while** $|\mathcal{P}_k| > 1$ and not reaching J **do**
6:       **if** $\boldsymbol{x}_u^{(j_k)} \leq \eta_k$ **then**
7:         $k \leftarrow 2k$
8:       **else**
9:         $k \leftarrow 2k + 1$
10:       **end if**
11:       $p(\boldsymbol{x}_u|\tau_i) \leftarrow p(\boldsymbol{x}_u|\tau_i) \cup k, \beta \leftarrow \beta + |\boldsymbol{x}_u^{(j_k)} - \eta_k|$
12:     **end while**
13:     $g(\boldsymbol{x}_u|\tau_i) \leftarrow \beta/|p(\boldsymbol{x}_u|\tau_i)|$
14:   **end for**
15: **end for**
16: **return**
  $\mathscr{F}_{\text{DEAS}}(\boldsymbol{o}|\mathcal{T}) \leftarrow 2^{-\mathbb{E}_{\tau_i \in \mathcal{T}} \frac{|p(\boldsymbol{x}_u|\tau_i)|}{C(\mathcal{T})}} \times \mathbb{E}_{\tau \in \mathcal{T}}(g(\boldsymbol{x}_u|\tau_i))$

---

overall complexity is linear w.r.t. the ensemble size $r \times t$. As for the process in Algorithm 2, the traversing process takes a similar computation process, which has linear time complexity w.r.t. the size of testing sets and the ensemble size. Overall, according to the above analysis, the time complexity of DIF is $O(ND(r \times t))$. It has linear complexity w.r.t. data size, dimensionality, and ensemble size, which inherits desired scalability from iForest.

2) DIF as a Generalisation of iForest and EIF: EIF [4] is a recent extension of iForest [3], which has been shown to be a
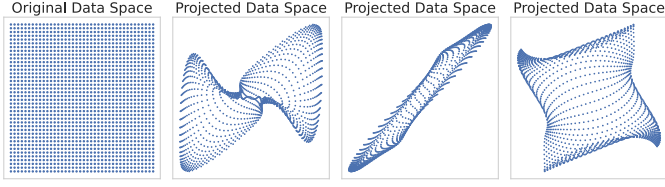
Fig. 3. Data space transformation produced by casually initialised neural networks (original data space vs. three transferred data spaces).

generalisation of iForest in [4]. We show that DIF can be viewed as a further higher-level generalisation of isolation methods used in both iForest and EIF, i.e., the branching criteria used in iForest and EIF can be transformed into the format of DIF.

Let $\boldsymbol{o} \in \mathbb{R}^D$ be a vectorised data object. Recall that the branching criterion in DIF is $\phi(\boldsymbol{o})^{(j)} \leq \eta$. Both iForest and EIF are special cases of DIF when the neural network ϕ is with one linear layer parameterised by a weight matrix $\mathbf{W}$, i.e., $\phi(\boldsymbol{o}) = \mathbf{W}^{\top}\boldsymbol{o}$. iForest splits the node by using the criterion $\boldsymbol{o}^{(j)} \leq \eta$, while DIF degrades to iForest if the weight matrix is set as an identity matrix, i.e., $\mathbf{W} = I_D$. EIF uses a slicing hyper-plane for each node branching, and the slope of its hyper-plane is a normal vector $\boldsymbol{k} \in \mathbb{R}^D$, and $\boldsymbol{k}^{(i)} \sim \mathcal{N}(0,1), \forall i \in \{1, \cdots D\}$. The intercept of the hyper-plane $\boldsymbol{p} \in \mathbb{R}^D$ is uniformly selected over the range of possible values at each branching point. The branching criterion is $(\boldsymbol{o} - \boldsymbol{p}) \cdot \boldsymbol{k} \leq 0$, which is equivalent to $\boldsymbol{o} \cdot \boldsymbol{k} \leq \boldsymbol{p} \cdot \boldsymbol{k}$. We can fulfil exactly the same operation in DIF when the weight matrix satisfies $\mathbf{W} \in \mathbb{R}^{D \times 1}$. The elements in $\mathbf{W}$ should also be initialised by a normal distribution $\mathcal{N}(0,1)$ to satisfy $\mathbf{W} = \boldsymbol{k}$. Additionally, the splitting point $\eta = \boldsymbol{p} \cdot \boldsymbol{k}$ can be understood as a standard random vector $\boldsymbol{p}$ sampled in possible values with a Gaussian noise $\boldsymbol{k}$.

E. Discussions

The power of DIF mainly depends on (i) the strong representation ability of neural networks, (ii) the discard of optimised representations, and (iii) the synergy between random representations and random partition-based isolation, which are respectively discussed below.

1) Representation Ability of Neural Networks: Neural networks have strong representation power, even for randomly initialised networks. As shown in Fig. 3 where the new data spaces are generated by random neural networks, the randomness in these initialised networks can create highly diversified data spaces, on which simple axis-parallel cuts can be equivalent to sophisticated slicing cuts in the original data space. Non-linear activation functions can effectively tweak and fold partition bounds to embed non-linearity into the isolation process, even though the networks are not optimised at all. On the other hand, there have been different deep learning architectures developed for various data types, so DIF is empowered to handle diverse data types by plugging the data-specific network backbone (e.g., multi-perceptron networks, recurrent networks, or graph neural networks) to produce the representations (see Section VI-A3 for different neural networks used in DIF and Section VI-B for their performance in different data types).

2) Optimised Versus Casually Initialised Representations: In general, we can use many representation learning networks specifically designed for anomaly detection, such as those in [10], [11], [13], [34], to obtain well-optimised feature representations. These representations are more expressive than randomly initialised representations if the optimisation objective well fits the input data. However, instead of using optimised representations, DIF uses the casually initialised representations due to the following two main reasons. (i) These loss functions are not versatile. It is difficult to devise one representation learning loss that can fit different data with diversified characteristics. (ii) The downstream data partition might be strongly controlled by the optimisation process. This way weakens the randomness and diversity of the feature representations, which are required in isolation-based anomaly scoring methods. These two intuitions are empirically investigated in our experiments by showing the performance of DIF on optimised representations produced by recent normality feature learning algorithms and the quality of these optimised representations (see Section VI-E1).

3) Synergy Between Random Representations and Random Partition-Based Isolation: DIF employs a novel representation scheme, i.e., the random representation ensemble produced via optimisation-free neural networks. The parameters of these networks can be initialised by randomly sampling from widely-used initialisation distributions (e.g., normal or uniform distribution), easily yielding a set of feature representations with excellent randomness and diversity. Given a sufficiently large set of such random representations, we can largely boost the isolation power in the random data partition, making it possible to effectively isolate some really hard anomalies on some subsets of these representations. For example, as shown in Fig. 1, among a large set of new representation spaces, there are some selective new spaces where hard anomalies become easy-to-isolate data objects. Recall that isolation methods, including DIF, are based on an average measure for anomaly scoring. Thus, the anomalies would stand out in the anomaly scores as long as they are effectively isolated in some of the isolation trees. DIF utilises this unique synergy between random representations and random partition-based isolation to largely improve the isolation process and subsequently the anomaly scoring function, resulting in significantly improved effectiveness of isolation-based anomaly detection. We empirically investigate the significance of this synergy by respectively replacing random representations and random partition-based isolation with multiple alternatives (see Section VI-E).

VI. EXPERIMENTS

We now introduce our experimental analysis. This section is organised as follows: In Section VI-A, we first start with the experimental setup including the used datasets, competing methods, parameter settings, and evaluation metrics; Section VI-B, C, and D evaluate the performance of our method w.r.t. effectiveness, scalability, and robustness; and in Section VI-E and F, we empirically analyse our method by investigating the significance of the synergy between random representations and

TABLE II
DATASET INFORMATION OF THE USED TABULAR, GRAPH, AND TIME SERIES
(TS) DATASETS

	Data	N	D	#Anom (ratio)
Tabular	Analysis	95,677	197	2,677 (2.80%)
	Backdoor	95,329	197	2,329 (2.44%)
	DoS	96,000	197	3,000 (3.13%)
	Exploits	96,000	197	3,000 (3.13%)
	R8	3,974	9,468	51 (1.28%)
	Cover	286,048	11	2,747 (0.96%)
	Fraud	284,807	30	492 (0.17%)
	Pageblocks	5,393	11	510 (9.46%)
	Shuttle	1,013	10	13 (1.28%)
	Thrombin	1,909	139,352	42 (2.20%)
Graph	HSE	8,417 (8,150/267)	17	10 (3.75%)
	MMP	7,558 (7,320/238)	18	38 (0.50%)
	p53	8,903 (8,634/269)	18	28 (10.41%)
	PPAR	8,451 (8,184/267)	17	15 (5.62%)
TS	Mars	11,349 (3,500/7,849)	5	64 (0.82%)
	Gait	65,000 (22,167/42,833)	3	401 (0.94%)
	Heart	55,374 (20,185/35,189)	3	201 (0.57%)
	Heart-w	80,000 (20,000/60,000)	1	251 (0.42%)

N indicates the number of data objects, with parenthesis denoting the pre-defined training/testing size of graph and time series datasets. D denotes the number of dimensionality in tabular datasets, the average number of nodes per graph in graph datasets, and the number of sequences in each time series dataset. #anom (ratio) is the number of anomalies and the corresponding ratios.

random partition-based isolation and the contribution of CERE and DEAS.

A. Experimental Setup

1) Datasets: We employ a large collection of publicly available and commonly-used real-world datasets, including ten tabular datasets, four graph datasets, and four time-series datasets. Their basic information is shown in Table II.

Tabular Data. Analysis, *Backdoor*, *DoS*, and *Exploits* are taken from a popular intrusion detection benchmark UNSW_NB 15. Following [15], [35], we select different attacks as anomalies in these datasets against normal network traffic. *R8* is a highly-imbalanced text classification dataset, where the rare class are treated as anomalies by following [10], [36]. *Cover* is from the ecology domain. *Fraud* is for fraudulent credit card transaction detection. *Pageblocks* and *Shuttle* are provided by an anomaly benchmark study [37]. *Thrombin* is to detect unusual molecular bio-activity for drug design, which is an ultrahigh-dimensional anomaly detection dataset used in [10].

Graph Data: We employ datasets from the popular graph benchmark Tox21, which is a project on toxicity evaluation of newly synthesised or used chemical compounds. These datasets are chosen since they are inherently imbalanced and contain real anomalies. The task is to detect abnormal graphs.

Time-Series Data: Time series data are taken from the UCR time series anomaly archive [38]. We employ data with natural anomalies. *Mars* is from NASA spacecraft. *Gait* is sensor data of a subject who has Huntington's disease (highly asymmetric gait), and anomalies are data from the weak leg. *ECG* is heartbeat data, where anomalies are ventricular beats, and *ECG-wandering* (*ECG-w* for short) is a long stretch of ECG with a wondering baseline.

2) Competing Methods: To have a comprehensive comparison, DIF is compared with the following two types of anomaly detection approaches.

iForest and its Extensions (IF-Based Methods): Apart from the popular iForest algorithm [3], its three advanced variants, i.e., EIF [4], PID [6], and LeSiNN [23], are employed. EIF slices data by using hyperplanes with random slopes and intercepts. In PID, the choice of splits is optimised based on the variance in the sparsity, and the anomaly scoring is redefined according to the sparsity. LeSiNN employs the nearest neighbour distance-based isolation ensemble, which is also known as aNNE in [39].

Ensemble of Deep Anomaly Detectors[1]: Different state-of-the-art (SOTA) deep anomaly detection methods are employed as base models, and we use the deep ensemble framework [27] to construct a suite of ensemble-based deep contenders. For tabular data, we utilise four state-of-the-art deep methods including RDP [13], REPEN [10], Deep SVDD [11], and a reconstruction-based Autoencoder baseline (RECON for short) [1]. We also employ the CERE ensemble method used in DIF for those methods to ensure their time efficiency and have a fair competition. As for graph data, a deep graph-level anomaly detector GLocalKD [40] is used. TranAD [41] is employed for time-series data. Their ensemble versions are denoted as eRDP, eREPEN, eDSVDD, eRECON, eGLocalKD, and eTranAD. All these methods are specifically designed for anomaly detection on the corresponding data type.

3) Parameter Settings and Implementations: DIF uses 50 representations ($r = 50$) and 6 isolation trees per representation ($t = 6$), with 256 as subsampling size ($n = 256$) for each iTree. DIF processes tabular data by using fully-connected multi-layer-perceptron networks. All of the IF-based competing methods use 300 trees (the ensemble size is the same as DIF). The subsampling size is set as 256. We use the maximum extension level of EIF, i.e., the extension level is adaptively set as the dimensionality minus 1. For LeSiNN, the subsampling size is 8 by following [23]. As IF-based competitors cannot directly process non-tabular data, we employ the latest powerful unsupervised representation learning models to yield high-quality vectorised representations, and specifically, InfoGraph [42] and TS2Vec [43] are respectively utilised for graph data and time series. InfoGraph uses GIN as its graph encoder architecture, and TS2Vec applies a dilated CNN module with residual blocks. They propose specific learning objectives to optimise generated representations. For the sake of fairness, DIF also respectively utilises the same GIN and CNN network structure to handle graph data and time series but without any optimisation steps. The deep anomaly detectors are trained by 50 epochs and 30 steps per epoch. REPEN, DSVDD, and RECON take an Adam optimiser with a 1e-3 learning rate and use 64 objects per mini-batch, and we empirically found that RDP can work significantly better when using 1e-4. The default/recommended settings are used for the other parameters of these competing methods.

---

[1]The ensemble performance of these deep models generally outperforms their solitary versions, and thus we focus on the comparison between ensemble-based results in the following experiments.

TABLE III
AUC-ROC AND AUC-PR PERFORMANCE (MEAN ± STANDARD DEVIATION) OF DIF AND IF-BASED COMPETING METHODS ON TABULAR DATASETS

Data	AUC-ROC					AUC-PR				
	DIF (ours)	EIF	PID	LeSiNN	IF	DIF (ours)	EIF	PID	LeSiNN	IF
Analysis	$\mathbf{0.931}_{\pm 0.006}$	$0.910_{\pm 0.005}$	$0.820_{\pm 0.019}$	$0.903_{\pm 0.008}$	$0.782_{\pm 0.017}$	$\mathbf{0.404}_{\pm 0.051}$	$0.198_{\pm 0.022}$	$0.075_{\pm 0.007}$	$0.183_{\pm 0.028}$	$0.063_{\pm 0.006}$
Backdoor	$\mathbf{0.918}_{\pm 0.002}$	$0.902_{\pm 0.005}$	$0.808_{\pm 0.016}$	$0.894_{\pm 0.006}$	$0.731_{\pm 0.021}$	$\mathbf{0.453}_{\pm 0.051}$	$0.218_{\pm 0.028}$	$0.066_{\pm 0.005}$	$0.205_{\pm 0.031}$	$0.046_{\pm 0.004}$
DoS	$\mathbf{0.932}_{\pm 0.003}$	$0.918_{\pm 0.004}$	$0.802_{\pm 0.013}$	$0.896_{\pm 0.009}$	$0.747_{\pm 0.020}$	$\mathbf{0.440}_{\pm 0.023}$	$0.269_{\pm 0.027}$	$0.075_{\pm 0.004}$	$0.185_{\pm 0.028}$	$0.060_{\pm 0.005}$
Exploits	$\mathbf{0.858}_{\pm 0.010}$	$0.840_{\pm 0.008}$	$0.797_{\pm 0.011}$	$0.816_{\pm 0.005}$	$0.745_{\pm 0.010}$	$\mathbf{0.273}_{\pm 0.020}$	$0.167_{\pm 0.011}$	$0.077_{\pm 0.003}$	$0.120_{\pm 0.013}$	$0.062_{\pm 0.003}$
R8	$\mathbf{0.930}_{\pm 0.008}$	$0.854_{\pm 0.006}$	$0.881_{\pm 0.018}$	$0.859_{\pm 0.001}$	$0.853_{\pm 0.016}$	$\mathbf{0.145}_{\pm 0.031}$	$0.101_{\pm 0.009}$	$0.078_{\pm 0.011}$	$0.094_{\pm 0.000}$	$0.075_{\pm 0.008}$
Cover	$\mathbf{0.972}_{\pm 0.010}$	$0.872_{\pm 0.017}$	$0.939_{\pm 0.007}$	$0.885_{\pm 0.008}$	$0.888_{\pm 0.017}$	$\mathbf{0.246}_{\pm 0.069}$	$0.040_{\pm 0.006}$	$0.069_{\pm 0.006}$	$0.051_{\pm 0.004}$	$0.055_{\pm 0.008}$
Fraud	$\mathbf{0.953}_{\pm 0.002}$	$0.950_{\pm 0.001}$	$0.950_{\pm 0.002}$	$0.952_{\pm 0.000}$	$0.950_{\pm 0.001}$	$0.387_{\pm 0.031}$	$0.378_{\pm 0.027}$	$0.186_{\pm 0.033}$	$\mathbf{0.401}_{\pm 0.001}$	$0.155_{\pm 0.015}$
Pageblocks	$\mathbf{0.903}_{\pm 0.006}$	$0.902_{\pm 0.001}$	$0.851_{\pm 0.003}$	$0.887_{\pm 0.002}$	$0.900_{\pm 0.005}$	$\mathbf{0.547}_{\pm 0.012}$	$0.537_{\pm 0.006}$	$0.421_{\pm 0.011}$	$0.511_{\pm 0.007}$	$0.476_{\pm 0.013}$
Shuttle	$\mathbf{0.941}_{\pm 0.006}$	$0.843_{\pm 0.009}$	$0.864_{\pm 0.017}$	$0.805_{\pm 0.019}$	$0.862_{\pm 0.019}$	$\mathbf{0.150}_{\pm 0.017}$	$0.061_{\pm 0.003}$	$0.059_{\pm 0.008}$	$0.048_{\pm 0.001}$	$0.075_{\pm 0.014}$
Thrombin	$\mathbf{0.913}_{\pm 0.003}$	OOM	OOM	$0.912_{\pm 0.000}$	$0.905_{\pm 0.002}$	$\mathbf{0.468}_{\pm 0.020}$	OOM	OOM	$0.458_{\pm 0.001}$	$0.372_{\pm 0.008}$
Average	$\mathbf{0.925}_{\pm 0.006}$	$0.888_{\pm 0.006}$	$0.857_{\pm 0.011}$	$0.881_{\pm 0.004}$	$0.836_{\pm 0.013}$	$\mathbf{0.351}_{\pm 0.033}$	$0.219_{\pm 0.015}$	$0.123_{\pm 0.010}$	$0.226_{\pm 0.011}$	$0.144_{\pm 0.008}$
p-value	-	0.004	0.004	0.002	0.002	-	0.004	0.004	0.006	0.002

PID and EIF run out of memory (OOM) on the ultrahigh-dimensional dataset *Thrombin*.
The best performer is boldfaced.

TABLE IV
AUC-ROC AND AUC-PR PERFORMANCE OF DIF AND ITS DEEP ENSEMBLE-BASED COMPETING METHODS

Data	AUC-ROC					AUC-PR				
	DIF (ours)	eRDP	eREPEN	eDSVDD	eRECON	DIF (ours)	eRDP	eREPEN	eDSVDD	eRECON
Analysis	$\mathbf{0.931}_{\pm 0.006}$	$0.899_{\pm 0.005}$	$0.883_{\pm 0.026}$	$0.844_{\pm 0.006}$	$0.862_{\pm 0.002}$	$\mathbf{0.404}_{\pm 0.051}$	$0.400_{\pm 0.019}$	$0.168_{\pm 0.051}$	$0.374_{\pm 0.028}$	$0.093_{\pm 0.002}$
Backdoor	$\mathbf{0.918}_{\pm 0.002}$	$0.900_{\pm 0.006}$	$0.863_{\pm 0.022}$	$0.916_{\pm 0.004}$	$0.840_{\pm 0.003}$	$\mathbf{0.453}_{\pm 0.051}$	$0.422_{\pm 0.020}$	$0.163_{\pm 0.055}$	$0.448_{\pm 0.030}$	$0.082_{\pm 0.004}$
DoS	$\mathbf{0.932}_{\pm 0.003}$	$0.905_{\pm 0.004}$	$0.861_{\pm 0.025}$	$0.900_{\pm 0.004}$	$0.856_{\pm 0.004}$	$\mathbf{0.440}_{\pm 0.023}$	$0.420_{\pm 0.013}$	$0.154_{\pm 0.047}$	$0.409_{\pm 0.028}$	$0.105_{\pm 0.004}$
Exploits	$\mathbf{0.858}_{\pm 0.010}$	$0.795_{\pm 0.008}$	$0.744_{\pm 0.040}$	$0.748_{\pm 0.017}$	$0.808_{\pm 0.002}$	$\mathbf{0.273}_{\pm 0.020}$	$0.250_{\pm 0.008}$	$0.080_{\pm 0.019}$	$0.249_{\pm 0.013}$	$0.086_{\pm 0.001}$
R8	$\mathbf{0.930}_{\pm 0.008}$	$0.864_{\pm 0.020}$	$0.904_{\pm 0.004}$	$0.883_{\pm 0.014}$	$0.779_{\pm 0.000}$	$\mathbf{0.145}_{\pm 0.031}$	$0.123_{\pm 0.017}$	$0.086_{\pm 0.005}$	$0.074_{\pm 0.006}$	$0.082_{\pm 0.000}$
Cover	$0.972_{\pm 0.010}$	$0.966_{\pm 0.008}$	$0.896_{\pm 0.013}$	$\mathbf{0.981}_{\pm 0.005}$	$0.922_{\pm 0.004}$	$0.246_{\pm 0.069}$	$0.192_{\pm 0.049}$	$0.044_{\pm 0.006}$	$\mathbf{0.353}_{\pm 0.055}$	$0.067_{\pm 0.003}$
Fraud	$0.953_{\pm 0.002}$	$0.947_{\pm 0.002}$	$\mathbf{0.954}_{\pm 0.001}$	$0.948_{\pm 0.002}$	$0.946_{\pm 0.001}$	$0.387_{\pm 0.039}$	$0.329_{\pm 0.025}$	$0.345_{\pm 0.030}$	$\mathbf{0.480}_{\pm 0.026}$	$0.372_{\pm 0.002}$
Pageblocks	$\mathbf{0.903}_{\pm 0.006}$	$0.850_{\pm 0.008}$	$0.896_{\pm 0.005}$	$0.882_{\pm 0.005}$	$0.829_{\pm 0.012}$	$\mathbf{0.547}_{\pm 0.012}$	$0.466_{\pm 0.011}$	$0.511_{\pm 0.010}$	$0.397_{\pm 0.009}$	$0.542_{\pm 0.032}$
Shuttle	$\mathbf{0.941}_{\pm 0.006}$	$0.900_{\pm 0.009}$	$0.780_{\pm 0.025}$	$0.837_{\pm 0.037}$	$0.779_{\pm 0.014}$	$\mathbf{0.150}_{\pm 0.017}$	$0.132_{\pm 0.009}$	$0.031_{\pm 0.003}$	$0.144_{\pm 0.022}$	$0.044_{\pm 0.002}$
Thrombin	$0.913_{\pm 0.003}$	$0.869_{\pm 0.019}$	$\mathbf{0.914}_{\pm 0.004}$	$0.366_{\pm 0.019}$	$0.911_{\pm 0.000}$	$\mathbf{0.468}_{\pm 0.020}$	$0.368_{\pm 0.053}$	$0.399_{\pm 0.056}$	$0.016_{\pm 0.001}$	$0.457_{\pm 0.000}$
Average	$\mathbf{0.925}_{\pm 0.006}$	$0.889_{\pm 0.009}$	$0.870_{\pm 0.016}$	$0.831_{\pm 0.011}$	$0.853_{\pm 0.004}$	$\mathbf{0.351}_{\pm 0.034}$	$0.310_{\pm 0.022}$	$0.198_{\pm 0.029}$	$0.294_{\pm 0.022}$	$0.193_{\pm 0.005}$
p-value	-	0.002	0.01	0.01	0.002	-	0.002	0.002	0.232	0.002

All the anomaly detection algorithms in our experiments are implemented using Python, with iForest from `scikit-learn` package, EIF from `eif` package, and other methods from their authors' releases. The implementation of our method is publicly available at https://github.com/xuhongzuo/deep-iforest.

4) Evaluation Metrics and Computing Infrastructure: Following the mainstream evaluation protocols of anomaly detection [3], [4], [15], the detection accuracy is evaluated by two complementary metrics including Area Under the ROC Curve (AUC-ROC) and Area Under the PR Curve (AUC-PR). ROC curve indicates the true positives against false positives, while PR curve summarises precision and recall of the anomaly class only. The paired *Wilcoxon* signed rank test is used to examine the statistical significance of the performance of DIF against each competing method.

We then introduce a new metric called Anomaly Isoability Index (AII) to measure the quality of representations. We borrow the concept of triplet loss [44] to count the percentage of effectively isolated anomalies among all true anomalies in each representation space, i.e.,

$$AII = \mathop{\mathbf{P}}_{\boldsymbol{a} \sim \mathcal{A}} \left(\mathop{median}_{\boldsymbol{n}_i \in \mathcal{N}} \left\{ \frac{1}{|\mathcal{C}|} \sum_{\boldsymbol{n}_j \in \mathcal{C}} (\omega(\boldsymbol{a}|\boldsymbol{n}_i, \boldsymbol{n}_j)) \right\} > 0 \right), \tag{9}$$

where $\omega(\boldsymbol{a}|\boldsymbol{n}_i, \boldsymbol{n}_j) = d(\boldsymbol{a}, \boldsymbol{n}_i) - d(\boldsymbol{n}_j, \boldsymbol{n}_j)$ denotes the difference between the euclidean distances of the two pairs, $\boldsymbol{a} \sim \mathcal{A}$ represents any anomaly drawn from the true anomaly set in the dataset, $\mathcal{C}$ is a group of randomly sampled normal anchors, and

$\mathcal{N}$ is another set of random normal samples to delegate the whole normal distribution. All the above data objects are from the target representation space. $|\mathcal{C}| = 20$ and $|\mathcal{N}| = 1000$ are used as we found empirically that these two settings are sufficiently large to compute the AII metric.

The computational time of all methods is based on a workstation with Intel Xeon Silver 4210R CPU, a single NVIDIA TITAN RTX GPU, and 64 GB RAM.

B. Effectiveness in Reducing False Negatives

1) Tabular Data: Tables III and IV present the AUC-ROC and AUC-PR results of our method DIF and eight competing methods. Overall, DIF largely reduces the false negatives compared to iForest and its three extensions, resulting in substantial averaged performance improvement in detection precision and/or recall rates, and thus, DIF obtains superior AUC-PR and AUC-ROC performance. Particularly, in the average AUC-PR, DIF significantly outperforms EIF (61%), PID (186%), LeSiNN (56%), iForest (144%), eRDP (13%), eREPEN (77%), eDSVDD (19%), and eRECON (82%). DIF also obtains 4% - 11% AUC-ROC improvement across these methods.

As shown in Table III, DIF significantly outperforms isolation-based methods at the 99% confidence level. DIF is the best isolation-based detector across all the ten datasets except AUC-PR on *Fraud*, on which the nearest neighbour-based anomaly measure LeSiNN is more effective. The features in *Fraud* are the results of PCA transformation due to the confidentiality issues, and thus the distance concept used in

TABLE V
RESULTS ON DETECTING ABNORMAL GRAPHS AND ANOMALIES IN TIME SERIES

	Data	AUC-ROC					AUC-PR				
		DIF (ours)	EIF	LeSiNN	iForest	eGLocalKD	DIF (ours)	EIF	LeSiNN	iForest	eGLocalKD
Graph	HSE	$\mathbf{0.737}_{\pm 0.013}$	$0.715_{\pm 0.014}$	$0.702_{\pm 0.001}$	$0.697_{\pm 0.014}$	$0.593_{\pm 0.002}$	$\mathbf{0.094}_{\pm 0.005}$	$0.088_{\pm 0.004}$	$0.084_{\pm 0.000}$	$0.082_{\pm 0.004}$	$0.054_{\pm 0.000}$
	MMP	$\mathbf{0.715}_{\pm 0.006}$	$0.663_{\pm 0.012}$	$0.666_{\pm 0.000}$	$0.667_{\pm 0.018}$	$0.675_{\pm 0.001}$	$\mathbf{0.260}_{\pm 0.006}$	$0.216_{\pm 0.006}$	$0.217_{\pm 0.000}$	$0.219_{\pm 0.011}$	$0.233_{\pm 0.001}$
	p53	$\mathbf{0.680}_{\pm 0.008}$	$0.597_{\pm 0.017}$	$0.606_{\pm 0.000}$	$0.619_{\pm 0.013}$	$0.640_{\pm 0.001}$	$\mathbf{0.177}_{\pm 0.006}$	$0.138_{\pm 0.004}$	$0.144_{\pm 0.000}$	$0.143_{\pm 0.004}$	$0.150_{\pm 0.000}$
	PPAR	$0.701_{\pm 0.013}$	$0.716_{\pm 0.005}$	$0.711_{\pm 0.000}$	$\mathbf{0.733}_{\pm 0.009}$	$0.643_{\pm 0.001}$	$0.127_{\pm 0.008}$	$0.173_{\pm 0.006}$	$0.165_{\pm 0.001}$	$\mathbf{0.208}_{\pm 0.012}$	$0.086_{\pm 0.000}$
		DIF (ours)	EIF	LeSiNN	iForest	eTranAD	DIF (ours)	EIF	LeSiNN	iForest	eTranAD
TS	Mars	$0.952_{\pm 0.017}$	$\mathbf{0.980}_{\pm 0.006}$	$0.942_{\pm 0.014}$	$0.947_{\pm 0.015}$	$0.947_{\pm 0.016}$	$\mathbf{0.626}_{\pm 0.024}$	$0.458_{\pm 0.031}$	$0.400_{\pm 0.009}$	$0.390_{\pm 0.043}$	$0.334_{\pm 0.020}$
	Gait	$\mathbf{0.998}_{\pm 0.001}$	$0.997_{\pm 0.001}$	$0.998_{\pm 0.000}$	$0.997_{\pm 0.001}$	$0.998_{\pm 0.000}$	$\mathbf{0.835}_{\pm 0.064}$	$0.772_{\pm 0.048}$	$0.829_{\pm 0.010}$	$0.741_{\pm 0.073}$	$0.806_{\pm 0.010}$
	ECG	$\mathbf{0.997}_{\pm 0.001}$	$0.986_{\pm 0.001}$	$0.987_{\pm 0.000}$	$0.987_{\pm 0.001}$	$0.976_{\pm 0.001}$	$\mathbf{0.809}_{\pm 0.031}$	$0.705_{\pm 0.004}$	$0.710_{\pm 0.000}$	$0.711_{\pm 0.002}$	$0.692_{\pm 0.001}$
	ECG-w	$\mathbf{1.000}_{\pm 0.000}$	$0.988_{\pm 0.001}$	$0.985_{\pm 0.001}$	$0.981_{\pm 0.001}$	$0.990_{\pm 0.000}$	$\mathbf{1.000}_{\pm 0.000}$	$0.255_{\pm 0.011}$	$0.219_{\pm 0.008}$	$0.181_{\pm 0.005}$	$0.297_{\pm 0.001}$

the nearest neighbour information can well reflect the proximity relationship. By contrast, DIF does not rely on such prior information that is not always reliable. These results demonstrate superior isolation power of DIF, which can effectively isolate anomalies that may not be possible in existing IF-based methods due to the challenges like data sparsity and non-linearity. This is particularly true on challenging high-dimensional datasets like *R8*, *Analysis*, *Backdoor* and *DoS*.

Compared to deep ensemble-based methods, DIF performs significantly better at the 99% confidence level according to the AUC-ROC performance in Table IV. DIF achieves the best performance on seven out of ten datasets in terms of both AUC-ROC and AUC-PR, and it obtains very competitive results on the rest three datasets with less than 0.01 difference in AUC-ROC. Nevertheless, the comparison results between DIF and its deep ensemble-based counterparts are very encouraging given the fact that DIF does not involve any optimisation while these deep methods need to be properly trained using pre-defined objective functions to better expose anomalies. The superiority of DIF in this comparison owes to the representation diversity of each ensemble member and the unique synergy between random representations and random partition-based isolation in the downstream anomaly scoring process. More importantly, DIF runs significantly faster than these deep ensemble-based contenders by around two orders of magnitude (see Section VI-C). Its superior computational efficiency endows DIF with stronger practicability in real-world applications.

2) Graph Data and Time Series: As reported in Section VI-A3, traditional isolation-based methods are performed upon vectorised representations learned by unsupervised representation learning method InfoGraph [42] and TS2Vec [43]. DIF respectively utilises the same network structure (GIN and dilated CNN) in InfoGraph and TS2Vec but without any optimisation steps. GLocalKD [40] is a graph-level anomaly detector, and TranAD [41] is a deep anomaly detection method specifically designed for time series. We also employ the ensemble version of these two SOTA methods as competitors.

Detection performance on graph data and time series are shown in Table V, in which EIF and LeSiNN are selected as contenders due to their preferable performance in tabular datasets over PID, with iForest also included as a baseline. DIF is the best performer on three out of four datasets in both graph-level and time series anomaly detection tasks. This performance of DIF is remarkable in that it outperforms not only the isolation-based methods that are empowered by the
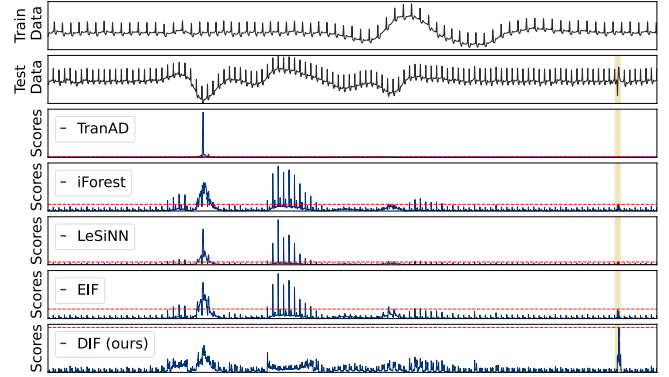

Fig. 4. **(Top to bottom)** Snippets of training/testing data of *ECG-w* with wandering baseline and detection results, i.e., anomaly scores, of DIF and its competitors. Anomalous heartbeats are highlighted in golden yellow. Red dashed lines indicate the reported highest anomaly scores in this anomalous duration.

latest powerful representation learning models but the recent SOTA methods that are specifically designed to extensively learn data-type-specific characteristics (e.g., holistic graph structure or temporal dependence) for effectively detecting these graph/sequential anomalies. By contrast, DIF performs well in a unified framework by only replacing the network structure with a different randomly initialised network backbone, offering a significantly simpler yet data-type-agnostic effective solution.

We further show how DIF works by visualising its results on a time series dataset. Fig. 4 visualises the ECG-w data (the id is sddb49 in the UCR benchmark) and the detection results of DIF and its competing methods. As illustrated in the introduction of the benchmark, this dataset is a challenging case that may confuse many algorithms due to its wandering shape. Albeit wandering baseline in testing data, we can also see this in the training set, and thus detection models are expected to tolerate these noisy regions. Our method DIF reduces the false negatives by successfully yielding significantly higher anomaly scores on anomalous heartbeats highlighted in golden yellow. In comparison, competing methods are misled by the fluctuating trends, showing much higher scores on noisy regions and omitting real anomalies.

C. Scalability to High-Dimensional, Large-Scale Data

This experiment examines the scalability of DIF and its contenders. These anomaly detectors are performed on a group of synthetic tabular datasets with different sizes and dimensionalities to record their training times. Nine datasets are with 5,000
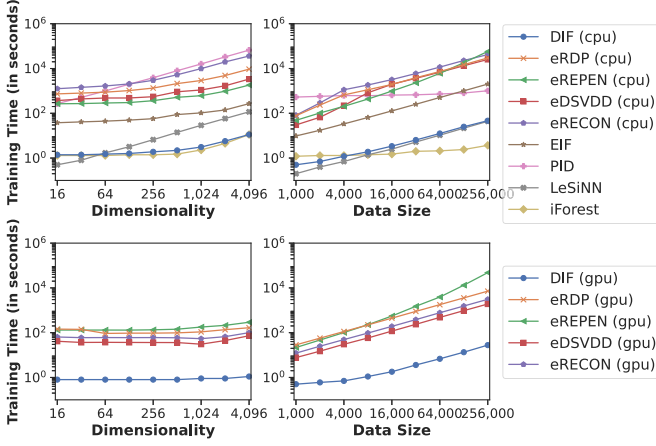
Fig. 5. Scalability test results. (**Top**) The training time of all the anomaly detectors on a CPU device; and (**Bottom**) The results of deep ensemble-based methods (including DIF) on a GPU device.

data objects, and their dimensions range from 16 to 4,096. The other nine datasets are with 32 features and varied sizes from a minimum of 1,000 up to 256,000. IF-based anomaly detectors only need CPU devices, while deep ensemble-based methods can leverage GPU acceleration. Therefore, we report the training time of DIF and its deep competitors on both GPU and CPU devices.

Fig. 5 (top) shows the scalability test results of DIF and its competing methods on a CPU device, and Fig. 5 (bottom) reports the comparison using GPU. DIF and all the other isolation-based methods present good scalability w.r.t. both dimensionality and data size compared to deep ensemble-based methods when using CPU computation. This is owed to the subsampling-based methods applied to both data samples and dimensions in isolation-based detectors. Besides, deep ensemble-based methods can greatly benefit from GPU acceleration when handling high-dimensional data. DIF shows outstanding time efficiency compared to its deep ensemble-based counterparts since it only requires one feed-forward step instead of a large number of training epochs. These results demonstrate that DIF inherits excellent computational efficiency from the iForest. Note that DIF can obtain almost the same scalability as iForest w.r.t. the data dimensionality, as DIF performs isolation on newly projected spaces of much smaller dimensionality while iForest works on the original data space.

D. Robustness W.r.t. Anomaly Contamination

This experiment examines the performance of DIF and its contenders when datasets containing different anomaly contamination ratios. Time-series datasets and graph datasets have pre-defined train-test split and their training sets do not contain anomalies, and thus we use tabular datasets as our bases here. Following [15], [35], we adjust the contamination ratios by injecting/removing anomalies, such that ratios range from 0% to 10%. These detection approaches are trained on adjusted datasets with controlled contamination ratios and tested on the original version.

The AUC-ROC performance is reported in Fig. 6. Generally, the performance of all the anomaly detectors downgrades with the increasing contamination ratio. Nevertheless, DIF has relatively clear superiority and stronger robustness in most of the datasets. The success of deep ensemble on out-of-distribution robustness has been proved in many recent studies [28], [30], [45], which partially explains why deep ensemble-based methods show better robustness w.r.t. anomaly contamination than IF-based methods. However, these competing methods still fail to provide consistently good robustness (e.g., eRDP, eREPEN, and eDSVDD on *R8* and eRECON on *Backdoor*). It is mainly due to two reasons: (i) their ensemble processes may suffer from the diversity problem, and (ii) their scoring strategies and training objectives rely on strong assumptions such as the distance concept that might not hold in some datasets. It is interesting to note that eREPEN shows an uptrend in *Shuttle*. REPEN uses LeSiNN to estimate initial anomaly scores when generating triplet mini-batches. As LeSiNN is with good robustness on *Shuttle*, this initial estimation can reliably obtain more anomaly examples as positive data when the contamination ratio increases, and thus the triplet learning process might benefit from the augmentation of the positive class.

E. Significance of the Synergy Between Random Representations and Random Partition-Based Isolation

We respectively replace random representations and random isolation-based anomaly scoring in DIF to investigate their synergy effect.

1) Representation Scheme: This experiment evaluates the effectiveness of our novel representation scheme by comparing it with representations produced by optimised neural networks, including RDP [13], REPEN [10], DSVDD [11] and Reconstruction-based Autoencoder [1]. That is, we replace random representations with representations learned by one of these methods, with all the other components of DIF fixed. These four variants are denoted as RDP-DIF, REPEN-DIF, DSVDD-DIF and RECON-DIF. RDP, REPEN, and DSVDD are originally designed for learning a good representation for anomaly detection, and a dense representation can be implicitly derived from Autoencoders. The representations are evaluated by the AUC-ROC performance and the individual representation quality as follows.

AUC-ROC Results. The AUC-ROC results are shown in the upper half of Fig. 7. DIF outperforms four optimised representation ensemble-based methods on five datasets and shows very competitive performance to the best performer on the other five datasets. Averagely, the random representation ensemble contributes to 5%, 5%, 7%, and 15% AUC-ROC improvement than RDP-DIF, REPEN-DIF, DSVDD-DIF, and RECON-DIF, respectively. Although DIF may not be the best performer on all the datasets, it is very encouraging to see this performance achieved by the ensemble of random representations when compared to those optimised representations trained by state-of-the-art learning objectives.

Quality per Representation: To further analyse the mechanism behind the above results, we directly evaluate the quality
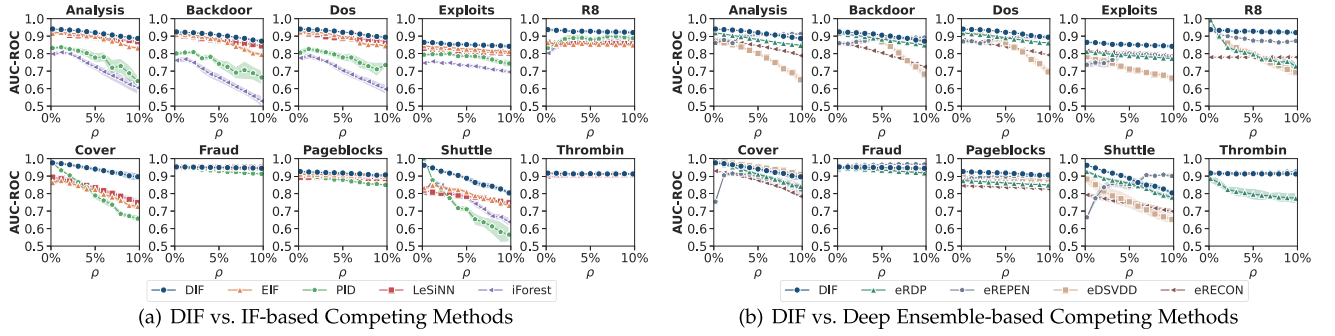
(a) DIF vs. IF-based Competing Methods   (b) DIF vs. Deep Ensemble-based Competing Methods

Fig. 6.   AUC-ROC w.r.t. different contamination ratios ρ (the percentage of anomalies in the training set).
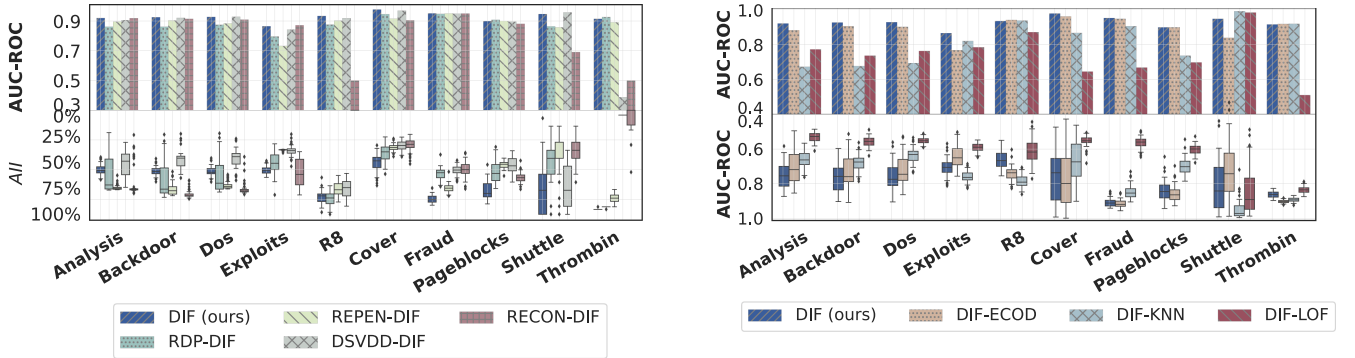

Fig. 7.   (**Top**) AUC-ROC of DIF and its variants that use optimised representations, and (**Bottom**) the quality (measured by AII) distribution of randomised representations used in DIF and representations optimised by RDP, REPEN, DSVDD, and RECON.

Fig. 8.   (**Top**) AUC-ROC of DIF and its variants of using other anomaly scoring strategies upon random representations, and (**Bottom**) their effectiveness (measured by AUC-ROC) on each ensemble member within the randomised representation groups.

of each representation produced by DIF and its variants. The representation quality is measured by Anomaly Isoability Index (AII), as introduced in Section VI-A4. The AII results of each representation used in five anomaly detection methods are shown in Fig. 7 (bottom), in which we use box plots to present the quality distribution of representations produced by 50 representations in the ensemble framework.

Based on the above experiment results, the following three remarks can be made.

- Our representation scheme achieves desired diversity and randomness, while simultaneously maintaining stable expressiveness in each representation, enabling excellent synergy with the downstream isolation-based anomaly scoring mechanism. This is the main driving force behind the superior performance of DIF.
- Optimised representations can be with consistently good quality on some datasets (e.g., near 80% true anomalies are well isolated by RECON-DIF on *Analysis*, *Backdoor*, and *DoS*), whereas the lack of diversity in representations downgrades the efficacy of this ensemble framework.
- Optimisation may even lead to worse representations on some datasets compared to random representations (e.g., *R8*, *Cover*, *Pageblocks* and *Thrombin*). This may be due to the fact that the underlying assumption (e.g., one-class assumption) in their learning objectives may not hold in those datasets.

2) Scoring Strategy: As shown in the prior experiment, random representations are with good diversity and stable quality, fostering a unique excellent synergy effect in downstream ensemble-based anomaly scoring. This section further justifies this intuition by investigating the effectiveness of combining random representations with other anomaly scoring methods, including probability-based anomaly scoring method ECOD [46], distance-based method KNN [47], and density-based method LOF [48]. Each of these methods is used to replace the isolation-based scoring process, with all the other modules fixed. These variants are denoted as DIF-ECOD, DIF-KNN, and DIF-LOF, respectively. Similarly, we evaluate their AUC-ROC performance and the individual scoring quality as follows.

AUC-ROC Results. The AUC-ROC results are shown in the upper half of Fig. 8. DIF outperforms these competing variants on seven out of ten datasets. Averagely, DIF outperforms DIF-ECOD, DIF-KNN, and DIF-LOF by 3%, 13%, and 25%, respectively. The superiority further justifies the synergy effect in DIF. Note that KNN and LOF are with very heavy computational overhead. They take around 60 hours to handle large-scale datasets *Cover* and *Fraud*.

Quality per Anomaly Scoring Result: We also study the quality of individual anomaly scoring results produced by DIF and its variants on each random representation, as shown in Fig. 8 (bottom). The quality is also estimated by AUC-ROC here.

We make the following three remarks.

TABLE VI
AUC-ROC AND AUC-PR RESULTS OF DIF AND **w/o** DEAS, AND TRAINING
TIME (IN SECONDS) OF DIF AND **w/o** CERE

Data	AUC-ROC		AUC-PR		Time (in seconds)	
	DIF	w/o DEAS	**DIF**	w/o DEAS	**DIF**	w/o CERE
Analysis	**0.931**	$0.922_{\pm 0.007}$	**0.404**	$0.315_{\pm 0.058}$	10.5	69.2
Backdoor	**0.918**	$0.914_{\pm 0.006}$	**0.453**	$0.381_{\pm 0.067}$	10.5	66.2
DoS	**0.932**	$0.926_{\pm 0.011}$	**0.440**	$0.388_{\pm 0.074}$	10.4	65.6
Exploits	**0.858**	$0.854_{\pm 0.008}$	**0.273**	$0.248_{\pm 0.041}$	10.4	65.8
R8	**0.930**	$0.915_{\pm 0.012}$	**0.145**	$0.123_{\pm 0.034}$	5.2	19.2
Cover	**0.972**	$0.964_{\pm 0.014}$	**0.246**	$0.210_{\pm 0.084}$	33.0	154.3
Fraud	**0.953**	$0.952_{\pm 0.002}$	**0.387**	$0.387_{\pm 0.042}$	33.6	169.1
Pageblocks	0.903	$\mathbf{0.912}_{\pm 0.007}$	0.547	$\mathbf{0.576}_{\pm 0.024}$	0.8	3.5
Shuttle	**0.941**	$0.923_{\pm 0.012}$	**0.150**	$0.103_{\pm 0.019}$	0.5	1.0
Thrombin	0.913	$\mathbf{0.916}_{\pm 0.002}$	**0.468**	$0.448_{\pm 0.020}$	13.8	336.7
Avg./Total	**0.925**	$0.920_{\pm 0.008}$	**0.351**	$0.318_{\pm 0.046}$	128.7	950.6
p-value	-	0.07	-	0.03	-	-

w/o DEAS is an ablated version by replacing deas with the standard scoring function. w/o CERE only uses the typical method to produce the representation ensemble.

- These competing scoring methods can produce markedly better individual scoring results than our isolation-based scoring mechanism on *Exploits*, *R8*, and *Thrombin*. However, they only yield less effective or slightly better integrated results compared to DIF. These scoring methods are unable to leverage the diversity embedded in our representation scheme.
- By contrast, DIF combines data representation and anomaly scoring in a successful unified ensemble learning framework. DIF achieves better integrated performance by fully leveraging the diversity and randomness of representations, e.g., above 0.9 AUC-ROC on *R8* with the maximum individual value only achieving around 0.7.
- DIF is inferior to its variants on *Shuttle*. It might be because anomalies in this dataset can be more easily identified by using the prior concepts used in these competing scoring methods (i.e., probability, distance, or density). However, these prior concepts may fail to work properly across all the datasets.

F. Ablation Study on CERE and DEAS

Considering detection effectiveness and computational efficiency, we propose the Deviation-Enhanced Anomaly Scoring function (DEAS) and the Computation-Efficient deep Representation Ensemble method (CERE) in the specific implementation of DIF. This experiment is conducted to examine whether DIF can have better detection performance and use less training time with the help of DEAS and CERE. Two ablated variants are employed, i.e., **w/o** CERE replaces $\mathscr{G}_{CERE}$ with the conventional sequential ensemble process, and **w/o** DEAS uses the standard scoring function used in iForest to replace our scoring function $\mathscr{F}_{DEAS}$.

The AUC-ROC and AUC-PR results of DIF and **w/o** DEAS are shown in Table VI. DIF significantly outperforms **w/o** DEAS at the 90% confidence interval and achieves approximate 11% AUC-PR improvement. Besides, DIF costs considerably less training time with the help of CERE. The total training time across all the ten datasets is only approximate one-tenth of the variant **w/o** CERE. The AUC-ROC/AUC-PR results of **w/o** CERE are on par with DIF, which are omitted due to the space limitation. Based on the above comparison results, the

contribution of the proposed DEAS and CERE is validated and quantitatively measured.

VII. CONCLUSION

This article introduces DIF, a novel extension of iForest. DIF takes the deep neural network-based random representation ensemble as a new representation scheme, enabling significantly versatile data partition in diverse random directions on subspaces of different sizes. The anomaly scoring can be facilitated by the synergy between random representations and random partition-based isolation. This enables DIF to fulfil (i) more effective isolation of anomalies, especially hard anomalies in data with intractable sparsity and non-linearity; (ii) liberation of isolation process from existing constraints to tackle the artefact problem; and (iii) versatile ability to handle different data types. Extensive experiments show that DIF significantly outperforms iForest and its existing extensions not only on tabular data but also on graph and time-series data. DIF also shows promising improvement compared to the ensemble of state-of-the-art deep anomaly detectors.

In our future work, we plan to devise new scoring strategies, together with relevant neural network backbones, to address other challenging yet important anomaly detection tasks, e.g., identify possible abnormal subsets [49] and multi-view data [50].

REFERENCES

[1] C. C. Aggarwal, *Outlier Analysis*. Berlin, Germany: Springer, 2017.
[2] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surveys*, vol. 54, no. 2, pp. 1–38, 2021.
[3] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. IEEE 8th Int. Conf. Data Mining*, 2008, pp. 413–422.
[4] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1479–1489, Apr. 2021.
[5] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "On detecting clustered anomalies using SCiforest," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discov. Databases*, Springer, 2010, pp. 274–290.
[6] P. Gopalan, V. Sharan, and U. Wieder, "PIDForest: Anomaly detection via partial identification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15 809–15 819.
[7] J. Lesouple, C. Baudoin, M. Spigai, and J.-Y. Tourneret, "Generalized isolation forest for anomaly detection," *Pattern Recognit. Lett.*, vol. 149, pp. 109–119, 2021.
[8] D. Cortes, "Revisiting randomized choices in isolation forests," 2021, *arXiv:2110.13402*.
[9] M. Tokovarov and P. Karczmarek, "A probabilistic generalization of isolation forest," *Inf. Sci.*, vol. 584, pp. 433–449, 2022.
[10] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2041–2050.
[11] L. Ruff et al., "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 4393–4402.
[12] L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–12.
[13] H. Wang, G. Pang, C. Shen, and C. Ma, "Unsupervised representation learning by predicting random distances," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 2950–2956.
[14] B. Liu, D. Wang, K. Lin, P.-N. Tan, and J. Zhou, "RCA: A deep collaborative autoencoder approach for anomaly detection," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, Art. no. 1505.
[15] G. Pang, C. Shen, and A. van den Hengel, "Deep anomaly detection with deviation networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 353–362.
[16] L. Ruff et al., "A unifying review of deep and shallow anomaly detection," *Proc. IEEE*, vol. 109, no. 5, pp. 756–795, May 2021.

[17] A. F. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, "Systematic construction of anomaly detection benchmarks from real data," in *Proc. ACM SIGKDD Workshop Outlier Detection Description*, 2013, pp. 16–21.

[18] D. Patel, D. Phan, M. Mueller, and A. Rajasekharan, "Toolkit for time series anomaly detection," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2022, pp. 4812–4813.

[19] K.-H. Lai et al., "TODS: An automated time series outlier detection system," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 16 060–16 062.

[20] S. Alnegheimish, D. Liu, C. Sala, L. Berti-Equille, and K. Veeramachaneni, "Sintel: A machine learning framework to extract insights from signals," in *Proc. Int. Conf. Manag. Data*, 2022, pp. 1855–1865.

[21] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python toolbox for scalable outlier detection," *J. Mach. Learn. Res.*, vol. 20, pp. 1–7, 2019.

[22] Anomaly detection by IBM. Accessed: Feb. 27, 2023. [Online]. Available: https://developer.ibm.com/apis/catalog/ai4industry--anomaly-detection-product/Introduction/

[23] G. Pang, K. M. Ting, and D. Albrecht, "LeSiNN: Detecting anomalies by identifying least similar nearest neighbours," in *Proc. IEEE 15th Int. Conf. Data Mining Workshops*, 2015, pp. 623–630.

[24] 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," *Comput. Intell.*, vol. 34, no. 4, pp. 968–998, 2018.

[25] X. Zhang et al., "LSHiForest: A generic framework for fast tree isolation based ensemble anomaly analysis," in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 983–994.

[26] A. Mensi and M. Bicego, "Enhanced anomaly scores for isolation forests," *Pattern Recognit.*, vol. 120, 2021, Art. no. 108115.

[27] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6405–6416.

[28] F. D'Angelo and V. Fortuin, "Repulsive deep ensembles are Bayesian," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 3451–3465.

[29] A. Ramé and M. Cord, "DICE: Diversity in deep ensembles via conditional redundancy adversarial estimation," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–31.

[30] G. Nam, J. Yoon, Y. Lee, and J. Lee, "Diversity matters when learning from ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 8367–8377.

[31] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proc. SIAM Int. Conf. Data Mining*, SIAM, 2017, pp. 90–98.

[32] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, "Outlier detection for time series with recurrent autoencoder ensembles," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 2725–2732.

[33] Y. Wen, D. Tran, and J. Ba, "BatchEnsemble: An alternative approach to efficient ensemble and lifelong learning," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–19.

[34] H. Xu et al., "Beyond outlier detection: Outlier interpretation by attention-guided triplet deviation network," in *Proc. Web Conf.*, 2021, pp. 1328–1339.

[35] G. Pang, A. van den Hengel, C. Shen, and L. Cao, "Toward deep supervised anomaly detection: Reinforcement learning from partially labeled anomaly data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1298–1308.

[36] S. Rayana and L. Akoglu, "Less is more: Building selective anomaly ensembles," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 4, pp. 1–33, 2016.

[37] G. O. Campos et al., "On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study," *Data Mining Knowl. Discov.*, vol. 30, no. 4, pp. 891–927, 2016.

[38] R. Wu and E. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2421–2429, Mar. 2023.

[39] K. M. Ting, T. Washio, J. R. Wells, and S. Aryal, "Defying the gravity of learning curve: A characteristic of nearest neighbour anomaly detectors," *Mach. Learn.*, vol. 106, no. 1, pp. 55–91, 2017.

[40] R. Ma, G. Pang, L. Chen, and A. V. D. Hengel, "Deep graph-level anomaly detection by glocal knowledge distillation," in *Proc. Int. Conf. Web Search Data Mining*, 2022, pp. 704–714.

[41] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data," *Proc. VLDB Endowment*, vol. 15, no. 6, pp. 1201–1214, 2022.

[42] F.-Y. Sun, J. Hoffman, V. Verma, and J. Tang, "InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–16.

[43] Z. Yue et al., "TS2Vec: Towards universal representation of time series," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 8980–8987.

[44] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[45] S. Liu et al., "Deep ensembling with no overhead for either training or testing: The all-round blessings of dynamic sparsity," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–15.

[46] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, "ECOD: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Trans. Knowl. Data Eng.*, Mar. 16, 2022, doi: 10.1109/TKDE.2022.3159580.

[47] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proc. Int. Conf. Manag. Data*, 2000, pp. 427–438.

[48] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. Int. Conf. Manag. Data*, 2000, pp. 93–104.

[49] L. Huang et al., "Hybrid-order anomaly detection on attributed networks," *IEEE Trans. Knowl. Data Eng.*, early access, Oct. 06, 2021, doi: 10.1109/TKDE.2021.3117842.

[50] Y.-X. Ji et al., "Multi-view outlier detection in deep intact space," in *Proc. IEEE 19th Int. Conf. Data Mining*, 2019, pp. 1132–1137.

Hongzuo Xu received the bachelor's and master's degrees from the National University of Defense Technology, China, in 2017 and 2019, where he is currently working toward the PhD degree in computer science. His research interests include anomaly detection, outlier interpretation, weakly-supervised learning and data mining.
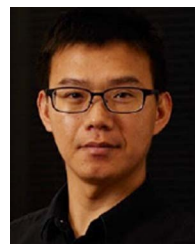
Guansong Pang (Member, IEEE) received the PhD degree from the University of Technology Sydney in 2019. He is a tenure-track assistant professor of Computer Science with the School of Computing and Information Systems, Singapore Management University (SMU), Singapore. Before joining SMU, he was a research fellow with the Australian Institute for Machine Learning (AIML). His research explores novel data mining and machine learning techniques and their applications, with a focus on detecting abnormal and unknown data.

Yijie Wang received the PhD degree in computer science and technology from the National University of Defense Technology in 1998. She was awarded the prize of National Excellent Doctoral Dissertation by Ministry of Education of PR China. She is currently a full professor with the Science and Technology on Parallel and Distributed Processing Laboratory (PDL), National University of Defense Technology. Her research interests include Big Data analysis, artificial intelligence and parallel and distributed processing.

Yongjun Wang received the PhD degree in computer architecture from the National University of Defense Technology, China, in 1998. He is currently a full professor with the College of Computer, National University of Defense Technology, Changsha, China. His research interests include network security and system security.