Intelligent Systems and Technologies,
Artificial Intelligence

Интеллектуальные системы и техологии,
искусственный интеллект

Research article
DOI: https://doi.org/10.18721/JCSTCS.18101
UDC 004.85

IMPROVED ANOMALY DETECTION BY USING THE ATTENTION-BASED ISOLATION FOREST WITH TRAINABLE SCORING FUNCTION

A.Yu. Ageev ✉ *, L.V. Utkin, A.V. Konstantinov*

Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russian Federation

✉ andreyageev1@mail.ru

Abstract. This paper proposes a novel anomaly detection model, called Attention-Based Isolation Forest with trainable Scoring Function (ABIF-SF). ABIF-SF enhances the original isolation forest algorithm by incorporating attention weights determined by scoring functions whose parameters are trained using gradient descent. The attention weights indicate the relevance of each data instance to the anomaly assessment task for each tree in the isolation forest. Two scoring functions are explored – scaled dot product and additive attention. Numerical experiments on real-world datasets demonstrate that ABIF-SF achieves better anomaly detection performance compared to isolation forest and attention-based isolation forest with the contamination model. The proposed method simplifies the computation of attention weights by using scoring functions and hinge loss optimization. The code implementation of ABIF-SF has been made publicly available for further research and benchmarking. Overall, the incorporation of trainable scoring functions to compute context-aware attention weights improves isolation forests for anomaly detection tasks.

Keywords: anomaly detection, attention mechanism, isolation forest, Nadaraya–Watson regression, quadratic programming, contamination model, additive attention

Acknowledgements: The research is partially financially supported by the Ministry of Science and Higher Education of the Russian Federation within the framework of the state assignment “Development and research of machine learning models for solving fundamental problems of artificial intelligence for the fuel and energy complex” (FSEG-2024-0027).

Citation: Ageev A.Yu., Utkin L.V., Konstantinov A.V. Improved anomaly detection by using the attention-based isolation forest with trainable scoring function. Computing, Telecommunications and Control, 2025, Vol. 18, No. 1, Pp. 7–22. DOI: 10.18721/JCSTCS.18101

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.18101
УДК 004.85

УЛУЧШЕННОЕ ОБНАРУЖЕНИЕ АНОМАЛИЙ С ПОМОЩЬЮ ЛЕСА ИЗОЛЯЦИИ НА ОСНОВЕ ВНИМАНИЯ С ОБУЧАЕМОЙ ФУНКЦИЕЙ ОЦЕНКИ

А.Ю. Агеев ✉ , *Л.В. Уткин, А.В. Константинов*

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

✉ andreyageev1@mail.ru

Аннотация. В данной статье предлагается новая модель обнаружения аномалий, называемая лесом изоляции на основе внимания с обучаемой функцией оценки (Attention-Based Isolation Forest with trainable Scoring Function, ABIF-SF). ABIF-SF улучшает исходный алгоритм леса изоляции, включая веса внимания, определяемые функциями оценки, параметры которых обучаются с помощью градиентного спуска. Веса внимания указывают на релевантность каждого экземпляра данных для задачи оценки аномалии для каждого дерева в лесу изоляции. Исследуются две функции оценки – масштабированное скалярное произведение и аддитивное внимание. Численные эксперименты на реальных наборах данных показывают, что ABIF-SF достигает лучшей производительности обнаружения аномалий по сравнению с лесом изоляции и лесом изоляции на основе внимания с моделью загрязнения. Предложенный метод упрощает вычисление весов внимания за счет использования функций оценки и оптимизации потерь шарнира. Реализация кода ABIF-SF была сделана общедоступной для дальнейших исследований и сравнительного анализа. В целом, включение обучаемых функций оценки для вычисления весов внимания с учетом контекста улучшает леса изоляции для задач обнаружения аномалий.

Ключевые слова: обнаружение аномалий, механизм внимания, лес изоляции, регрессия Надарая–Уотсона, квадратичное программирование, модель загрязнения, аддитивное внимание

Финансирование: Исследование выполнено при частичной финансовой поддержке Министерства науки и высшего образования Российской Федерации в рамках государственного задания «Разработка и исследование моделей машинного обучения для решения фундаментальных задач искусственного интеллекта в топливно-энергетическом комплексе» (FSEG-2024-0027).

Для цитирования: Ageev A.Yu., Utkin L.V., Konstantinov A.V. Improved anomaly detection by using the attention-based isolation forest with trainable scoring function // Computing, Telecommunications and Control. 2025. Т. 18, № 1. С. 7–22. DOI: 10.18721/JCSTCS.18101

Introduction

Anomalies are objects or events that significantly differ from normal or expected objects or events [1]. Anomalies can occur for various reasons, such as measurement errors, malicious attacks, equipment malfunctions or rare natural phenomena [2].

There are several classifications of anomalies in data. One of them is based on five dimensions: data type, relationship cardinality, anomaly level, data structure, and data distribution [3]. These dimensions lead to three broad groups of anomalies: point, collective, and contextual [3].

Anomaly detection is the process of identifying and detecting such anomalous data [4–7]. Anomaly detection is a challenging task due to the high dimensionality of data, noise, and heterogeneous distributions [5].

Depending on the type of anomalies, there are different detection methods that take into account the data characteristics. Point anomalies are individual data points that deviate significantly from the rest of the data in the set [6].

Point anomaly detection is the task of detecting such points and labeling them as anomalous or normal. Point anomaly detection can be useful in many cases, such as detecting malfunctions in industrial systems [7], identifying fraud in financial transactions [8], determining unusual user behavior in cybersecurity [9], etc.

Collective anomalies are those where a single data object in isolation appears normal. However, if it is considered in relation to other data objects or a subset of data objects, the object appears anomalous [6, 10].

Contextual anomalies are those that depend on the situation or context. Anomaly is determined based on certain conditions or rules. For example, high air temperature may be normal in summer but anomalous in winter [11].

There are many methods for point anomaly detection, which can be divided into three main groups: statistical methods based on probability distribution of data [12–14]; methods based on measuring proximity or distance between data points [15–17]; and density-based methods based on estimating the density of data in local neighborhoods [18–20]. Classical methods for detecting such anomalies include the Z-score, Tukey's test, and Grubb's test for statistical approach; k-nearest neighbors and local outlier factor for density-based approach; k-means and DBSCAN for clustering approach; isolation forest for isolation approach, which constructs random decision trees for separating normal and anomalous objects [21–25]. These methods work well for small and simple data but have their limitations. For example, they are sensitive to parameter selection, do not consider dependencies between features or temporal structure of data, and are not capable of generalizing to new types of anomalies.

Various machine learning (ML) methods can also be used for anomaly detection, which play an important role in this field. Depending on the presence or absence of class labels for normal and anomalous objects, ML methods can be divided into three types: supervised, semi-supervised, and unsupervised [26]. Supervised methods require enough examples for each class and are suitable for classification or regression tasks. Semi-supervised methods use only examples from one class (usually normal) and are suitable for one-class learning or generating new examples.

More modern methods use deep learning for detecting anomalies in complex and large data. They are based on constructing a model of normal data behavior using different neural network architectures: autoencoders, generative adversarial networks, recurrent networks, convolutional networks and others [27–31]. These methods have their own features and improvements compared to classical methods. For example, they can detect complex dependencies in data, working with weakly annotated or unlabeled data altogether.

Recently, attention-based methods have started to gain popularity, which allows models to focus on the most important parts of the data [32–36].

Attention weights are numerical coefficients that determine the degree of relevance of each data element to the task at hand. The use of attention weights can improve the quality of anomaly detection by providing a more accurate representation of the data and taking context into account [37]. Attention weights can be applied to various types of data, such as images, text, or time series [5, 37]. In the context of anomaly detection, the attention mechanism can be used to highlight the features or subsequences of data that are most relevant to determining the normality or abnormality of an object or event. Thus, the attention mechanism can help the model better understand the structure and dynamics of the data and increase the accuracy and efficiency of anomaly detection.

This article introduces a new method, ABIF-SF, based on isolation forest algorithm, which improves anomaly detection by incorporating a scoring function with trained weights in the attention mechanism. The attention weight computation process is simplified using gradient descent and hinge loss function.

The effectiveness of ABIF-SF is demonstrated through numerical experiments on real datasets and shows promising results.

Our contributions are:

1. We propose ABIF-SF, a novel anomaly detection method that enhances isolation forests through an attention mechanism implemented as a trainable scoring function. This allows the model to learn contextual weights indicating the relevance of different regions of the isolation forest for assessing anomalies.

2. A simplified optimization approach for computing attention weights based on gradient descent and the hinge loss function is introduced. Avoiding more complex contamination models streamlines training.

3. Demonstration of the effectiveness of ABIF-SF through numerical experiments on real datasets, which showed promising results.

Related works

Approaches to anomaly detection

Anomaly detection, a critical and well-explored problem across various domains, has seen significant advancements through deep learning techniques. Key methods include self-supervised learning [38], One-Class Classification (OCC) [39], time series anomaly detection [40], and domain-specific deep learning-based techniques [41]. Additionally, the use of deep learning for log file anomaly detection [42], GAN-based methods [43], video anomaly detection [44], and medical imaging [45] highlight the diversity of applications in this field.

Anomaly detection using the attention mechanism

The attention mechanism, vital for emphasizing significant data features in anomaly detection, originated in text translation models and has since expanded to other data types and tasks [5, 36, 37, 46, 47]. Significant works include anomaly detection in semiconductor production using GANs with attention [46], attention-based deep learning for vector magnetic field anomalies [47], and graph-based anomaly detection leveraging attention mechanisms [36].

iForest and its variations

The Isolation Forest (iForest) algorithm [23], known for its efficiency in large datasets, identifies anomalies based on the ease of isolation in binary trees. Despite its popularity, iForest faces limitations like feature correlation ignorance and potential normal sample misclassification [48]. To address these, enhancements such as local anomaly detection through k-means [49], the k-means-based iForest [50], and the minimum spanning tree-based approach [51] have been proposed.

Attention-Based iForest (ABIForest)

Building on the concept of iForest, ABIForest (ABIF) [32] incorporates an attention mechanism through Nadaraya–Watson regression to refine anomaly detection. This method, inspired by the ABRF model [52], requires careful parameter tuning for both the attention mechanism and the iForest component.

Preliminaries

Attention Mechanism as Nadaraya–Watson Regression

The attention mechanism prioritizes relevant elements in input data for specific tasks [53, 54], using the softmax function for weight calculation. Given a vector $\boldsymbol{z} = (z_1, \dots , z_n)$ the softmax function is:

$$\text{softmax}\left(z_j\right) = \frac{e^{z_j}}{\sum_j e^{z_j}}, \tag{1}$$

where each z_i is an element of the vector, and the denominator normalizes the sum of weights to 1.

Nadaraya–Watson regression [55, 56] uses weighted averages for prediction. Let $\boldsymbol{x}_i \in \mathbb{R}^d$ be the i-th data point, $y_i \in \mathbb{R}$ – its value, and $w_i(\boldsymbol{x})$ – the weight based on its proximity to target $\boldsymbol{x}$. The regression formula is:

$$\hat{y}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} w_i(\boldsymbol{x}) y_i}{\sum_{i=1}^{n} w_i(\boldsymbol{x})}, \tag{2}$$

where $\hat{y}(\boldsymbol{x})$ is the predicted value. The weights represent an attention mechanism, where the kernel function determines the similarity between query $\boldsymbol{x}$ and keys $\boldsymbol{x}_i$.

The attention weight $\alpha(\boldsymbol{x}, \boldsymbol{x}_i)$ is given by:

$$\alpha(\boldsymbol{x}, \boldsymbol{x}_i) = \frac{w(\boldsymbol{x}, \boldsymbol{x}_i)}{\sum_{i=1}^{n} w(\boldsymbol{x}, \boldsymbol{x}_i)}, \tag{3}$$

for a Gaussian kernel with parameter ω:

$$\alpha(\boldsymbol{x}, \boldsymbol{x}_i) = \sigma\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{\omega}\right), \tag{4}$$

where σ is the softmax function, and the expression within is the scoring function $\alpha(\boldsymbol{x}, \boldsymbol{x}_i)$.

Attention Scoring Function

Scoring functions calculate relevance weights in the attention mechanism [54, 57]. The additive scoring function, for vectors $\boldsymbol{q} \in \mathbb{R}^k$ and $\boldsymbol{k} \in \mathbb{R}^k$, is:

$$\alpha(\boldsymbol{q}, \boldsymbol{k}) = \boldsymbol{w}^T \tanh(\boldsymbol{W}_q \boldsymbol{q} + \boldsymbol{W}_k \boldsymbol{k}), \tag{5}$$

where $\boldsymbol{w} \in \mathbb{R}^m$, $\boldsymbol{W}_q \in \mathbb{R}^{m \times k}$, and $\boldsymbol{W}_k \in \mathbb{R}^{m \times k}$ are weight matrices and vectors.

The dot product with scaling, for vectors of dimensionality k, is:

$$\alpha(\boldsymbol{q}, \boldsymbol{k}) = \frac{\boldsymbol{q}^T \cdot \boldsymbol{k}}{\sqrt{k}}. \tag{6}$$

These functions use softmax to assign weights:

$$\text{softmax}(\alpha(\boldsymbol{q}, \boldsymbol{k})) = \frac{e^{\alpha(\boldsymbol{q}, \boldsymbol{k})}}{\sum_j e^{\alpha(\boldsymbol{q}, \boldsymbol{k}_j)}}. \tag{7}$$

iForest

The iForest algorithm [23] identifies anomalies, especially effective in large datasets. It isolates anomalies using binary trees from random data subsets.

Each tree randomly selects a feature and a value, splitting the data into two groups. This continues until maximum depth or isolation is achieved.

The anomaly score is the average path length from the root to the leaf across trees. The formal definition involves $\boldsymbol{x}_i \in \mathbb{R}^d$ in a forest F of T trees. The isolation degree $h(x)$ is:

$$h(\boldsymbol{x}, F) = 2^{-\frac{E[h(\boldsymbol{x})]}{c(T)}}, \tag{8}$$

with $c(T) = 2H(T-1) - \frac{2(T-1)}{n}$, where $H(i)$ is the harmonic series, and n is the sample size.

An object's classification as an anomaly uses threshold τ:

$$y(\boldsymbol{x})=\begin{cases}\text{anomaly, if } h(\boldsymbol{x},F)<\tau\\ \text{non}-\text{anomaly, otherwise}\end{cases}. \tag{9}$$

iForest's performance is sensitive to hyperparameters, requiring careful tuning for large datasets.

Attention-based iForest with scoring function

We propose a new method for ABIF-SF anomaly detection that incorporates an attention mechanism into iForest using scoring functions with trainable parameters.

Attention mechanism: query, keys, values

In the iForest method, the average path length $h_k(\boldsymbol{x})$ for a point $\boldsymbol{x}$ over all T trees can be expressed as follows:

$$E\left[h(\boldsymbol{x})\right]=\frac{1}{T}\sum_{k=1}^{T}h_k(\boldsymbol{x}), \tag{10}$$

where $h_k(\boldsymbol{x})$ is the path length of instance $\boldsymbol{x}$ in tree k and serves as the value. Using the attention mechanism allows us to rewrite the computation of the expected path length $E[h(\boldsymbol{x})]$ in iForest using attention weights $\alpha(\boldsymbol{x}, \boldsymbol{A_k}(\boldsymbol{x}), \boldsymbol{w})$ [32, 52]:

$$E\left[h(\boldsymbol{x})\right]=\sum_{k=1}^{T}\alpha\left(\boldsymbol{x},\boldsymbol{A_k}(\boldsymbol{x}),\boldsymbol{w}\right)\cdot h_k(\boldsymbol{x}), \tag{11}$$

where $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{A_k}(\boldsymbol{x})$ is the average vector of all vectors $\boldsymbol{x_j}$ with indices $j \in J_i(k)$ in the i-th leaf of the k-th tree that contains the feature vector $\boldsymbol{x}$, and $J_i(k)$ is the set of indices $n_i(k)$ of training instances that also fall into the same leaf, and $\boldsymbol{w}$ is a set of trainable parameters.

$$\boldsymbol{A_k}(\boldsymbol{x})=\frac{1}{n_i(k)}\sum_{j\in J_i(k)}\boldsymbol{x_j}. \tag{12}$$

This vector characterizes the group of instances in the corresponding leaf and serves as the key, while $\boldsymbol{x}$ serves as the query.

$\alpha(\boldsymbol{x}, \boldsymbol{A_k}(\boldsymbol{x}), \boldsymbol{w})$ represents the importance of the average instance $\boldsymbol{A_k}(\boldsymbol{x})$ for the vector $\boldsymbol{x}$ and satisfies the following conditions:

$$\sum_{k=1}^{T}\alpha\left(\boldsymbol{x},\boldsymbol{A_k}(\boldsymbol{x}),\boldsymbol{w}\right)=1,\ \alpha\left(\boldsymbol{x},\boldsymbol{A_k}(\boldsymbol{x}),\boldsymbol{w}\right)\geq 0,\ k=1,\dots,T. \tag{13}$$

In [32], the authors used Huber's contamination model with weights of the following form:

$$\alpha\left(\boldsymbol{x},\boldsymbol{A_k}(\boldsymbol{x}),\boldsymbol{w}\right)=(1-\varepsilon)\cdot\sigma\left(-\frac{\left\|\boldsymbol{x}-\boldsymbol{A_k}(\boldsymbol{x})\right\|^2}{\omega}\right)+\varepsilon\cdot w_k, \tag{14}$$

where $\varepsilon \in \mathbb{R}$, $\omega \in \mathbb{R}$ and σ is a sigmoid function. This equation shows that the attention weight depends linearly on the trainable parameters $\boldsymbol{w} = (w_1, \dots, w_T)$ where T is the number of components. The softmax operation depends only on the hyperparameter ω. The trainable parameters w are restricted to the unit simplex $S(1, T)$, which means that the constraints on w are linear ($w_i \geq 0$ and $w_1 + \dots + w_T = 1$).

One drawback of this formulation is that there are no trainable parameters in the sigmoid function. Essentially, the expression inside the sigmoid function represents scoring functions:

$$W_{opt} = \arg\min_{w \in M} \sum_{S=1}^{n} \max\left(0,\ y_S\left(\sum_{k=1}^{T} \alpha\left(\boldsymbol{x}, \boldsymbol{A}_{\boldsymbol{k}}\left(\boldsymbol{x}\right), \boldsymbol{W}_{\boldsymbol{Q}}, \boldsymbol{W}_{\boldsymbol{X}}\right) \cdot h_k\left(\boldsymbol{x}_{\boldsymbol{S}}\right) - \gamma\right)\right). \tag{15}$$

By using these scoring functions inside the sigmoid without the Huber model, the attention mechanism can learn to assign higher weights to more relevant components in the input, leading to improved accuracy. The Huber model is designed to be more robust to outliers, but it may also smooth out the gradients and make the learning process slower. By using only the sigmoid function with the scoring functions the model can directly optimize the attention weights based on the relevance of the components in the input, without the additional smoothing effect of the Huber model. This can lead to faster convergence and better performance in some cases.

Scoring functions as attention weights

Scoring functions can be used as $\alpha(\boldsymbol{x}, \boldsymbol{A}_k(\boldsymbol{x}), \boldsymbol{w})$ using, for example, dot-scale and additive attention.

For dot-scale attention, $\alpha(\boldsymbol{x}, \boldsymbol{A}_k(\boldsymbol{x}), \boldsymbol{w})$ can be defined as follows:

$$\alpha\left(\boldsymbol{x}, \boldsymbol{A}_k\left(\boldsymbol{x}\right), \boldsymbol{W}_{\boldsymbol{Q}}, \boldsymbol{W}_{\boldsymbol{X}}\right) = \sigma\left(\frac{\boldsymbol{W}_{\boldsymbol{Q}}^{\boldsymbol{T}} \cdot \boldsymbol{x} \cdot \boldsymbol{A}_k\left(\boldsymbol{x}\right) \cdot \boldsymbol{W}_{\boldsymbol{X}}^{(\boldsymbol{k})\boldsymbol{T}}}{\sqrt{d}}\right), \tag{16}$$

where $\boldsymbol{W}_Q \in \mathbb{R}^d$, $\boldsymbol{W}_X^{(k)} \in \mathbb{R}^d$ are trainable parameter vectors ($\boldsymbol{W}_X \in \mathbb{R}^{d\times T}$), d is the dimensionality of vectors $\boldsymbol{x}$ and $\boldsymbol{A}_k(\boldsymbol{x})$, and σ is the softmax function.

For additive attention, $\alpha(\boldsymbol{x}, \boldsymbol{A}_k(\boldsymbol{x}), \boldsymbol{w})$ can be defined as follows:

$$\alpha\left(\boldsymbol{x}, \boldsymbol{A}_k\left(\boldsymbol{x}\right), \boldsymbol{W}_{\boldsymbol{Q}}, \boldsymbol{W}_{\boldsymbol{X}}\right) = \sigma\left(\tanh\left(\boldsymbol{W}_{\boldsymbol{Q}}^{\boldsymbol{T}} \cdot \boldsymbol{x} + \boldsymbol{W}_{\boldsymbol{X}}^{(\boldsymbol{k})\boldsymbol{T}} \cdot \boldsymbol{A}_{\boldsymbol{k}}\left(\boldsymbol{x}\right)\right)\right). \tag{17}$$

The final form of computing $E[h(\boldsymbol{x})]$ for additive attention can be written as:

$$E\left[h\left(\boldsymbol{x}\right)\right] = \sum_{k=1}^{T} \sigma\left(\tanh\left(\boldsymbol{W}_{\boldsymbol{Q}}^{\boldsymbol{T}} \cdot \boldsymbol{x} + \boldsymbol{W}_{\boldsymbol{X}}^{(\boldsymbol{k})\boldsymbol{T}} \cdot \boldsymbol{A}_{\boldsymbol{k}}\left(\boldsymbol{x}\right)\right)\right) \cdot h_k\left(\boldsymbol{x}\right), \tag{18}$$

and for dot-scale:

$$E\left[h\left(\boldsymbol{x}\right)\right] = \sum_{k=1}^{T} \sigma\left(\frac{\boldsymbol{W}_{\boldsymbol{Q}}^{\boldsymbol{T}} \cdot \boldsymbol{x} \cdot \boldsymbol{A}_{\boldsymbol{k}}\left(\boldsymbol{x}\right) \cdot \boldsymbol{W}_{\boldsymbol{X}}^{(\boldsymbol{k})\boldsymbol{T}}}{\sqrt{d}}\right) \cdot h_k\left(\boldsymbol{x}\right). \tag{19}$$

In both cases, trainable parameters are included in the expression through $\boldsymbol{W}_Q$, $\boldsymbol{W}_X^{(k)}$.

To determine whether an object is an anomaly, a reformulation of the decision making from the classic isolation forest ($h(\boldsymbol{x}, F) < \tau$) should be used to make a decision based on $E[h(\boldsymbol{x})]$ [34].

$$y\left(\boldsymbol{x}\right) = \{\text{anomaly},\ \ E\left[h\left(\boldsymbol{x}\right)\right] \leq \gamma,\ \text{otherwise}. \tag{20}$$

Training attention weights allows the iForest models to better consider relationships between instances and each tree, which can overall improve the quality of anomaly detection.

Loss function

Standard optimization methods such as gradient descent or its variants can be used to train the parameters $\boldsymbol{W}_Q$ and $\boldsymbol{W}_X^{(k)}$.

To use optimization methods, it is necessary to define a loss function between the model prediction expressed through the expression $E[h(\boldsymbol{x_S})] - \gamma$ and the label of the data y_S, where the index S indicates the objects from the training dataset.

The loss function L has the form:

$$L\left(E\left[h\left(\boldsymbol{x_S}\right)\right]-\gamma, y_S\right) = \max\left(0, y_S\left(E\left[h\left(\boldsymbol{x_S}\right)\right]-\gamma\right)\right), \tag{21}$$

where y_S is the label of the instances (the label of instance y_S is 1 if $\boldsymbol{x_S}$ is anomalous and –1 if it is normal).

In [32], γ was calculated as follows:

$$\gamma = -c\left(n\right)\cdot \log_2\left(\tau\right). \tag{22}$$

We propose to include γ as a trainable parameter along with $\boldsymbol{W_Q}$ and $\boldsymbol{W_X^{(k)}}$.

The general form of the minimization problem can be written as follows:

$$W_{opt} = \arg\min_{w\in M} \sum_{S=1}^{n} \max\left(0, y_S\left(\sum_{k=1}^{T} \alpha\left(\boldsymbol{x}, \boldsymbol{A_k}\left(\boldsymbol{x}\right), \boldsymbol{W_Q}, \boldsymbol{W_X}\right)\cdot h_k\left(\boldsymbol{x_S}\right)-\gamma\right)\right), \tag{23}$$

where M is the space of trainable parameters for $\boldsymbol{W_Q}$, $\boldsymbol{W_X}$ and γ, s is the index of training instances, of which there are n instances.

When using gradient descent, gradients with respect to the trainable parameters are used. The general parameter optimization step is classical for gradient descent algorithms and its modifications.

Numerical experiments

The aim of this chapter is to provide a comprehensive evaluation of the proposed method using numerical experiments. The experiments are designed to demonstrate the effectiveness of the method in comparison to the other described in this article approaches, and to show the impact of various parameters on the performance of the method. In the experiments, we will compare the performance of the three models on a variety of datasets and use standard evaluation metrics such as F1-score to assess the performance of each model. The results will be presented in the form of tables and graphs to allow for a clear and comprehensive comparison of the models.

Gradient descent is used for optimization with the following parameters: learning rate is 0.001, optimizer – ADAM.

The experiments utilized both real-world and synthetic datasets spanning anomaly detection challenges across different domains:

- Arrhythmia[1] – electrocardiogram (ECG) slices from the Kaggle repository;
- Credit[2] – credit card transaction dataset from the Kaggle repository;
- Pima[3] – Pima Indians Diabetes Database from the NIDDK;
- EEG Eye[4] – electroencephalogram (EEG) eye state samples from the Kaggle repository;
- Haberman[5] – Haberman’s survival dataset from the Kaggle repository;

[1] Tavares M. Binary classification on arrhythmia dataset. Kaggle, 2023. Available: https://www.kaggle.com/code/mtavares51/binary-classification-on-arrhythmia-dataset (Accessed 29.08.2024)

[2] Sekra S. Credit card fraud detection – EDA & Isolation Forest. Kaggle, 2023. Available: https://www.kaggle.com/code/shivamsekra/credit-card-fraud-detection-eda-isolation-forest (Accessed 29.08.2024)

[3] Ramadan H. Data science project III. Kaggle, 2023. Available: https://www.kaggle.com/code/hafizramadan/data-science-project-iii (Accessed 29.08.2024)

[4] Scube R. Eye state classification EEG dataset. Kaggle, 2023. Available: https://www.kaggle.com/datasets/robikscube/eye-state-classification-eeg-dataset (Accessed 29.08.2024)

[5] Sousa G. Haberman's survival data set. Kaggle, 2023. Available: https://www.kaggle.com/datasets/gilsousa/habermans-survival-data-set (Accessed 29.08.2024)

- HTTP[6] – HTTP network intrusion dataset from the OpenML repository;
- Ionosphere[7] – radar returns from the ionosphere dataset in the Kaggle repository;
- Mulcross[8] – synthetically generated multivariate normal distribution with anomaly clusters from the OpenML repository.

Table 1

A brief introduction of the datasets

Dataset	Normal	Abnormal	Number of features
Arrhythmia	386	66	17
Credit	1500	400	30
Pima	500	268	8
EEG Eye	847	653	11
Haberman	225	81	3
HTTP	500	50	3
Ionosphere	225	126	33
Mulcross	1800	400	4

To facilitate computational efficiency, smaller excerpted samples rather than full dataset volumes were utilized for larger real-world sources (Table 1). Certain distributions also underwent preprocessing including normalization and feature selection to conform inputs to model assumptions, with code available in the ABIF-SF repository (https://github.com/AndreyAgeev/abif-sf).

In the experiments, we use the following evaluation metrics to assess the performance of the method:

- F1-score: The harmonic mean of precision and recall.

The proposed method was implemented using the programming language Python and the library PyTorch.

The method was compared with the following approaches:

- IForest;
- ABIF.

Experimental Results

Comparison between iForest, ABIF and ABIF-SF

To measure the performance, we use the F1-score, which is a commonly used metric in anomaly detection. We compare the F1-score dependence on the number of epochs on several datasets. To evaluate the F1-score, 66.7% of the data were randomly selected for training and 33.3% were randomly selected for testing.

The performance of the proposed method was compared with iForest and ABIF.

The results are shown in Table 2.

For these experiments, 5000 training epochs were carried out, the best weights was taken from the minimum error value on the training set. This approach was used to obtain a result from the point of view of a practitioner who could use a similar approach to quickly obtain a result without setting parameters, validation dataset and other parameters.

For the models, the number of trees 150 was chosen.

[6] HTTP. OpenML. 2023. Available: https://www.openml.org/search?type=data&sort=runs&id=40897&status=active (Accessed 29.08.2024)

[7] Zymzym. Classification of the Ionosphere dataset by KNN. Kaggle, 2023. Available: https://www.kaggle.com/code/zymzym/classification-of-the-ionosphere-dataset-by-knn (Accessed 29.08.2024)

[8] Mulcross. OpenML. 2023. Available: https://www.openml.org/search?type=data&sort=runs&id=40897&status=active (Accessed 29.08.2024)

Table 2

Comparison of algorithms on different data sets

	ABIF				iForest		Additive	Dot-scale
Dataset	ε_{opt}	τ	$\tau_{softmax}$	F1	τ	F1	F1	F1
Arrythmia	1.0	0.45	–	0.472	0.45	0.484	**0.853**	0.849
Credit	0.5	0.55	0.1	0.862	0.45	0.798	0.930	**0.932**
Pima	0.75	0.45	10	0.555	0.4	0.532	**0.667**	0.648
EEG Eye	1.0	0.45	–	**0.724**	0.35	**0.724**	0.5	0.543
Haberman	1.0	0.45	–	0.486	0.45	0.473	**0.732**	0.728
HTTP	0.0	0.55	0.1	0.739	0.5	0.628	**0.901**	0.880
Ionosphere	1.0	0.45	–	0.649	0.45	0.652	0.679	**0.686**
Mullcross	0.0	0.6	0.1	0.525	0.5	0.538	0.852	**0.897**

Hyperparameters for the isolation forest and attention-based models were selected through a grid search over reasonable values, following a procedure like that used by the authors of the original ABIF paper. Specifically, we predefined grids of potential hyperparameters, including contamination model epsilon values and anomaly thresholds. Models were trained and evaluated on a test set across the grid space. The best performing hyperparameter configuration on the test data was then selected and used to produce the primary results and comparisons between ABIF, ABIF-SF, and isolation forest reported in this work. We use 10 different seeds when building trees, and 10 times shuffle train/test dataset, and then average the results of the metrics.

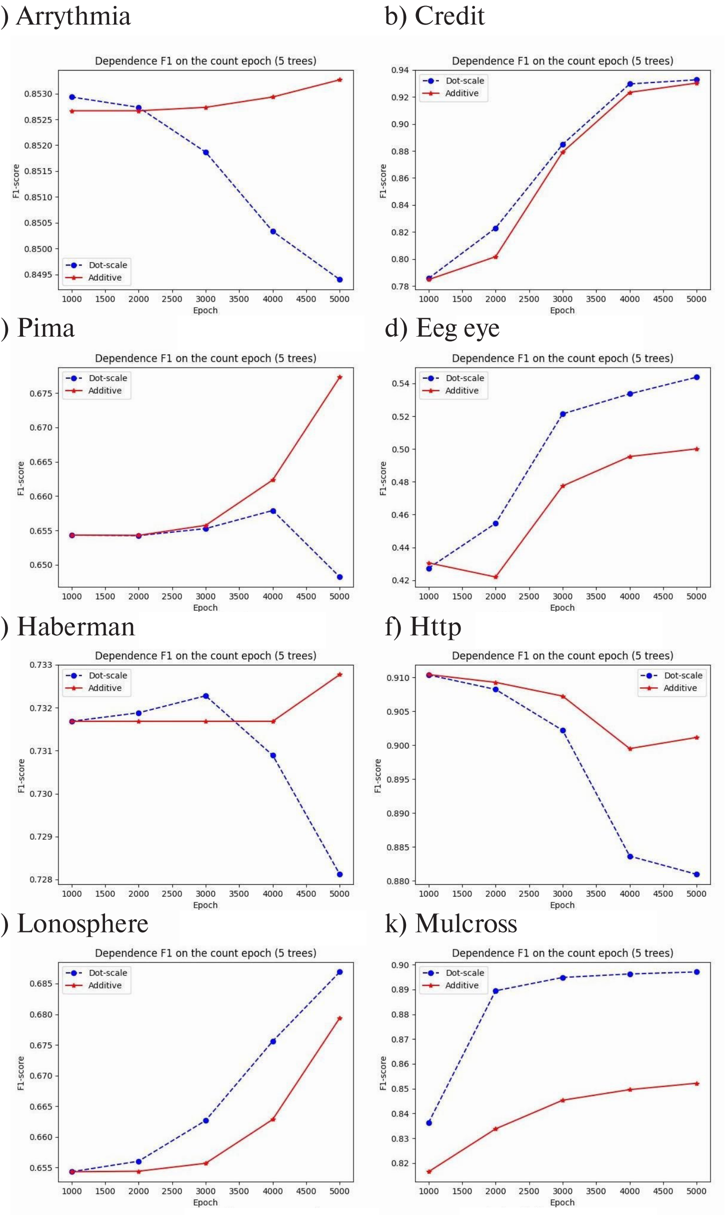
In the experiments, we used a smaller number of dataset partitions and different seeds due to the addition of new algorithms when comparing, and therefore, on average, the best results could be obtained with ϵ equal to 0 or 1, which does not coincide with the results of the author of the article on ABIF. When using more seeds and experiments on average on datasets, it is preferable to choose ϵ not equal to 0.

Analysis of learning dynamics

Monitoring model performance across training epochs provides insight into learning dynamics – identifying overfitting, suitable regularization, optimal timing to stop training, etc. Here we track the F1 score after each epoch on the test set to assess ABIF-SF's resilience to overfitting as additional iterations may better fit the training distribution without improving generalization. Ideally, test set metrics should steadily improve before plateauing once the intrinsic complexity is reached. Declining scores indicates overfitting – losing generalization due to redundant adaptation on noise or spurious patterns. The scoring functions contain little explicit regularization, hence the trends characterize inherent resistance to overlearning.

We trained dot-product and additive models for 5000 epochs on the Arrhythmia, Credit, EEG Eye, Haberman, HTTP, Ionosphere, and Mullcross datasets. The number of trees was fixed at five to better stress test potential overfitting. At each epoch, the parameter set minimizing training error was evaluated on the test data.

Fig. 1 shows the F1-score learning curves on the test datasets over training progression. The dot-product scoring consistently demonstrates stable or gradually improving F1 while not overfitting even after thousands of iterations. The additive attention exhibits more volatility, with drops in some datasets. For example, in the Arrhythmia set, additive scoring peaks at epoch 1000 before declining by nearly 3% in F1-score. However, dot-product matches best performance around epoch 4000 and smoothly converges thereafter. The EEG Eye dataset proves challenging for both formulations, plateauing below 60% F1-score. Still dot-product dominates additively, backed by superior Arrhythmia and Haberman results. The trends indicate inherent regularization properties differentiate the scoring mechanisms. Dot-product generalizes


a) Arrythmia

b) Credit

c) Pima

d) Eeg eye

e) Haberman

f) Http

g) Lonosphere

k) Mulcross

Fig. 1. Comparison ABIF-SF scoring function

reliably with extensive epochs while additive learning can become unstable. Sensitivity to initial conditions, co-adaptation of weights, and disparate gradient behaviors likely explain discrepancies. The results also highlight the harder EEG Eye distribution where better feature extraction is essential. In summary, tracking F1-score across training epochs reveals additive attention more vulnerable to overfitting than dot-product formulations. This highlights the greater regularization of dot mechanisms, also backed by consistently good performance into thousands of iterations. The analysis also identifies limitations modeling certain distributions and suggests enhancements like constrained optimization, dropout, or batch normalization to further boost robustness.

Impact of training set size

In real-world scenarios, the volume of quality training data available can vary significantly across anomaly detection tasks. To characterize the data efficiency and generalization capability of the proposed ABIF-SF model, we investigated performance with enlarged and reduced dataset sizes. Intuitively, additional high-quality examples should enable better learning of normal vs anomalous patterns. However, insufficient or redundant data could respectively lead to underfitting or overfitting. We evaluated the F1 score on the test set for the additive and dot-product scoring functions using 80%, 100%, and 120% of the original training set sizes. The Credit, HTTP, and Mulcross datasets were employed for these experiments with 2000 training epochs. At each epoch, the best-performing model parameterization on the training set was selected for final evaluation on the test data. Results are shown in Table 3.

Table 3

F1-Score vs training set size

	Additive			**Dot-Product**		
Dataset	80%	100%	120%	80%	100%	120%
Credit	0.794	0.801	0.834	0.814	0.822	0.853
HTTP	0.907	0.909	0.903	0.907	0.908	0.908
Mulcross	0.824	0.833	0.850	0.869	0.889	0.894

The Credit and Mulcross datasets exhibit consistent improvement in anomaly detection accuracy (F1-score) as more training examples are provided, plateauing at the maximum 120% volume. This demonstrates both scoring functions can effectively leverage additional representative data to better learn normal vs anomalous patterns in these distributions. However, the story differs markedly on the HTTP dataset. Surprisingly, the additive scoring function shows a decline in accuracy from 0.909 to 0.903 when switching from 100% to 120% training data volumes. At the same time, the dot-product scoring function remains stable at 0.908 F1-score despite variations in data amount. Decreasing the dataset size also only causes minor performance changes for both approaches. This reversal in trends for the HTTP dataset suggests differing generalization capabilities between the scoring formulations. The additive model appears to overfit on the augmented 120% training set – overly adapting to patterns that do not transfer to the test data. Meanwhile, the performance consistency of dot-product scoring implies it has saturated learning from this distribution once 100% examples are available. Additional data volume provides redundancy rather than meaningful new information. Furthermore, both functions achieve their maximal accuracy with only 80% subset, confirming enough representative information was intrinsically available in the original dataset. In conclusion, while ABIF-SF can leverage increased training data for some distributions, performance plateaus or drops past distribution-dependent optimal training set sizes. Choosing appropriate volumes with sufficient but concise representative examples is vital for efficiently learning anomalies, avoiding under- or over-fitting tendencies. Our experiments also highlight distinctions between the scoring formulations – additive functions may better model some distributions but are more prone to overfitting compared to more robust dot-product attention.

Conclusion

This article presents a novel anomaly detection model, the attention-based isolation forest with scoring function (ABIF-SF), which is an improvement of the original iForest. The proposed model utilizes attention weights, which are determined by scoring functions, to enhance its performance. The experiments conducted using real datasets demonstrate the superiority of the proposed model compared to the original isolation forest and the attention-based isolation forest eps-contamination model. The source code for this algorithm has been made publicly available for further research and development.

REFERENCES

1. **Chandola V., Banerjee A., Kumar V.** Anomaly detection: A survey. *ACM Computing Surveys* (*CSUR*), 2009, Vol. 41, No. 3, Pp. 1–58. DOI: 10.1145/1541880.1541882

2. **Barnett V., Lewis T.** *Outliers in Statistical Data*, 3rd ed. Chichester: Wiley. 1994. 584 p.

3. **Foorthuis R.** On the nature and types of anomalies: a review of deviations in data. *International Journal of Data Science and Analytics*, 2021, Vol. 12, Pp. 297–331. DOI: 10.1007/s41060-021-00265-1

4. **Liao Y., Bartler A., Yang B.** Anomaly detection based on selection and weighting in latent space. *arXiv:2103.04662*, 2021. DOI: 10.48550/arXiv.2103.04662

5. **Xu J., Wu H., Wang J., Long M.** Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv:2110.02642*, 2022. DOI: 10.48550/arXiv.2110.02642

6. **Li Z., Xiang Z., Gong W., Wang H.** Unified model for collective and point anomaly detection using stacked temporal convolution networks. *Applied Intelligence*, 2022, Vol. 52, Pp. 3118–3131. DOI: 10.1007/s10489-021-02559-0

7. **Chatterjee A., Ahmed B.S.** IoT anomaly detection methods and applications: A survey. *Internet of Things*, 2022, Vol. 19, Art. no. 100568. DOI: 10.1016/j.iot.2022.100568

8. **Wu T., Wang Y.** Locally interpretable one-class anomaly detection for credit card fraud detection. *arXiv:2108.02501*, 2021. DOI: 10.48550/arXiv.2108.02501

9. **Bierbrauer D.A., Chang A., Kritzer W., Bastian N.D.** Cybersecurity anomaly detection in adversarial environments. *arXiv:2105.06742*, 2021. DOI: 10.48550/arXiv.2105.06742

10. **Fisch A.T.M., Eckley I.A., Fearnhead P.** A linear time method for the detection of point and collective anomalies. *arXiv:1806.01947*, 2018. DOI: 10.48550/arXiv.1806.01947

11. **Li Z., van Leeuwen M.** Robust and explainable contextual anomaly detection using quantile regression forests. *arXiv:2302.11239v1*, 2023.

12. **Madhuri G.S., Rani M.U.** Anomaly detection techniques. *2018 IADS International Conference on Computing, Communications & Data Engineering* (*CCODE*), 2018. DOI: 10.2139/ssrn.3167172

13. **Gafni T., Wolff B., Revach G., Shlezinger N., Cohen K.** Anomaly search over discrete composite hypotheses in hierarchical statistical models. *IEEE Transactions on Signal Processing*, 2023, Vol. 71, Pp. 202–217. DOI: 10.1109/TSP.2023.3242074

14. **Kandanaarachchi S., Hyndman R.J.** Anomaly detection in dynamic networks. *arXiv:2210.07407*, 2022. DOI: 10.48550/arXiv.2210.07407

15. **Hou Y., Chen Z., Wu M., Foo C.-S., Li X., Shubair R.M.** Mahalanobis distance based adversarial network for anomaly detection. *2020 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2020, Pp. 3192–3196. DOI: 10.1109/ICASSP40776.2020.9053206

16. **Sarmadi H., Karamodin A.** A novel anomaly detection method based on adaptive Mahalanobis-squared distance and one-class kNN rule for structural health monitoring under environmental effects. *Mechanical Systems and Signal Processing*, 2020, Vol. 140, Art. no. 106495. DOI: 10.1016/j.ymssp.2019.106495

17. **Souto Arias L.A., Oosterlee C.W., Cirillo P.** AIDA: Analytic isolation and distance-based anomaly detection algorithm. *arXiv:2212.02645*, 2022. DOI: 10.48550/arXiv.2212.02645

18. **Wang W., Zhang B., Wang D., Jiang Y., Qin S., Xue L.** Anomaly detection based on probability density function with Kullback–Leibler divergence. *Signal Processing*, 2016, Vol. 126, Pp. 12–17. DOI: 10.1016/j.sigpro.2016.01.008

19. **Liu B., Tan P.-N., Zhou J.** Unsupervised anomaly detection by robust density estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, Vol. 36, No. 4, Pp. 4101–4108. DOI: 10.1609/aaai.v36i4.20328

20. **Le Lan C., Dinh L.** Perfect density models cannot guarantee anomaly detection. *arXiv:2012.03808*, 2020. DOI: 10.48550/arXiv.2012.03808

21. **Zimek A., Filzmoser P.** There and back again: Outlier detection between statistical reasoning and data mining algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2018, Vol. 8, No. 6, Art. no. e1280. DOI: 10.1002/widm.1280

22. **Knorr E.M., Ng R.T., Tucakov V.** Distance-based outliers: algorithms and applications. *The VLDB Journal*, 2000, Vol. 8, Pp. 237–253. DOI: 10.1007/s007780050006

23. **Liu F.T., Ting K.M., Zhou Z.-H.** Isolation forest. *2008 8th IEEE International Conference on Data Mining*, 2008, Pp. 413–422. DOI: 10.1109/ICDM.2008.17

24. **Fanaee-T H., Gama J.** Tensor-based anomaly detection: An interdisciplinary survey. *Knowledge-Based Systems*, 2016, Vol. 98, Pp. 130–147. DOI: 10.1016/j.knosys.2016.01.027

25. **Zimek A., Schubert E., Krieger H.-P.** A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 2012, Vol. 5, Pp. 363–387. DOI: 10.1002/sam.11161

26. **Nassif A.B., Talib M.A., Nasir Q., Dakalbab F.M.** Machine learning for anomaly detection: A systematic review. *IEEE Access*, 2021, Vol. 9, Pp. 78658–78700. DOI: 10.1109/ACCESS.2021.3083060

27. **Alloqmani A., Abushark Y.B., Khan A.I., Alsolami F.** Deep learning based anomaly detection in images: Insights, challenges and recommendations. *International Journal of Advanced Computer Science and Applications* (*IJACSA*), 2021, Vol. 12, No. 4, Pp. 205–215. DOI: 10.14569/IJACSA.2021.0120428

28. **Matsuo H., Nishio M., Kanda T., Kojita Y., Kono A.K., Hori M., Teshima M., Otsuki N., Nibu K.-i., Murakami T.** Diagnostic accuracy of deep-learning with anomaly detection for a small amount of imbalanced data: discriminating malignant parotid tumors in MRI. *Scientific Reports*, 2020, Vol. 10, Art. no. 19388. DOI: 10.1038/s41598-020-76389-4

29. **Kim M., Moon K.-R., Lee B.-D.** Unsupervised anomaly detection for posteroanterior chest -rays using multiresolution patch-based self-supervised learning. *Scientific Reports*, 2023, Vol. 13, Art. no. 3415. DOI: 10.1038/s41598-023-30589-w

30. **Maya S., Ueno K., Nishikawa T.** dLSTM: a new approach for anomaly detection using deep learning with delayed prediction. *International Journal of Data Science and Analytics*, 2019, Vol. 8, Pp. 137–164. DOI: 10.1007/s41060-019-00186-0

31. **Kurniabudi, Purnama B., Sharipuddin, Darmawijoyo, Stiawan D., Samsuryadi, Heryanto A., Budiarto R.** Network anomaly detection research: A survey. *Indonesian Journal of Electrical Engineering and Informatics* (*IJEEI*), 2019, Vol. 7, No. 1, Pp. 37–50. DOI: 10.11591/ijeei.v7i1.773

32. **Utkin L.V., Ageev A.Y., Konstantinov A.V., Muliukha V.A.** Improved anomaly detection by using the attention-based isolation forest. *Algorithms*, 2023, Vol. 16, No. 1, Art. no. 19. DOI: 10.3390/a16010019

33. **Takimoto H., Seki J., Situju S.F., Kanagawa A.** Anomaly detection using Siamese network with attention mechanism for few-shot learning. *Applied Artificial Intelligence*, 2022, Vol. 36, No. 1, Art. no. 2094885. DOI: 10.1080/08839514.2022.2094885

34. **Zhou H., Xia H., Zhan Y., Mao Q.** Salient attention model and classes imbalance remission for video anomaly analysis with weak label. *Human Centered Computing* (*HCC 2020*), 2021, Pp. 126–135. DOI: 10.1007/978-3-030-70626-5_13

35. **Dong F., Chen S., Demachi K., Yoshikawa M., Seki A., Takaya S.** Attention-based time series analysis for data-driven anomaly detection in nuclear power plants. *Nuclear Engineering and Design*, 2023, Vol. 404, Art. no. 112161. DOI: 10.1016/j.nucengdes.2023.112161

36. **Yu Y., Zha Z., Jin B., Wu G., Dong C.** Graph-based anomaly detection via attention mechanism. *Intelligent Computing Theories and Application* (*ICIC 2022*), 2022, Pp. 401–411. DOI: 10.1007/978-3-031-13870-6_33

37. **Zhu Y., Newsam S.** Motion-aware feature for improved video anomaly detection. *arXiv:1907.10211*, 2019. DOI: 10.48550/arXiv.1907.10211

38. **Hojjati H., Ho T.K.K., Armanfard N.** Self-supervised anomaly detection: A survey and outlook. *arXiv:2205.05173v5*, 2024.

39. **Perera P., Oza P., Patel V.M.** One-class classification: A survey. *arXiv:2101.03064*, 2021. DOI: 10.48550/arXiv.2101.03064

40. **Darban Z.Z., Webb G.I., Pan S., Aggarwal C.C., Salehi M.** Deep learning for time series anomaly detection: A survey. *arXiv:2211.05244*, 2022. DOI: 10.48550/arXiv.2211.05244

41. **Chalapathy R., Chawla S.** Deep learning for anomaly detection: A survey. *arXiv:1901.03407*, 2019. DOI: 10.48550/arXiv.1901.03407

42. **Landauer M., Onder S., Skopik F., Wurzenberger M.** Deep learning for anomaly detection in log data: A survey. *arXiv:2207.03820*, 2022. DOI: 10.48550/arXiv.2207.03820

43. **Di Mattia F., Galeone P., De Simoni M., Ghelfi E.** A survey on GANs for anomaly detection. *arXiv:1906.11632*, 2019. DOI: 10.48550/arXiv.1906.11632

44. **Suarez J.J.P., Naval Jr. P.C.** A survey on deep learning techniques for video anomaly detection. *arXiv:2009.14146*, 2020. DOI: 10.48550/arXiv.2009.14146

45. **Tschuchnig M.E., Gadermayr M.** Anomaly detection in medical imaging – A mini review. *Data Science – Analytics and Applications* (*iDSC 2021*), 2022, Pp. 33–38. DOI: 10.1007/978-3-658-36295-9_5

46. **Hashimoto M., Ide Y., Aritsugi M.** Anomaly detection for sensor data of semiconductor manufacturing equipment using a GAN. *Procedia Computer Science*, 2021, Vol. 192, Pp. 873–882. DOI: 10.1016/j.procs.2021.08.090

47. **Wu X., Huang S., Li M., Deng Y.** Vector magnetic anomaly detection via an attention mechanism deep-learning model. *Applied Sciences*, 2021, Vol. 11, No. 23. Art. no. 11533. DOI: 10.3390/app112311533

48. **Cortes D.** Isolation forests: looking beyond tree depth. *arXiv:2111.11639*, 2021. DOI: 10.48550/arXiv.2111.11639

49. **Gao R., Zhang T., Sun S., Liu Z.** Research and improvement of isolation forest in detection of local anomaly points. *Journal of Physics: Conference Series*, 2019, Vol. 1237, Art. no. 052023. DOI: 10.1088/1742-6596/1237/5/052023

50. **Karczmarek P., Kiersztyn A., Pedrycz W., Al E.** K-Means-based isolation forest. *Knowledge-Based Systems*, 2020, Vol. 195, Art. no. 105659. DOI: 10.1016/j.knosys.2020.105659

51. **Gałka L., Karczmarek P., Tokovarov M.** Isolation forest based on minimal spanning tree. *IEEE Access*, 2022, Vol. 10, Pp. 74175–74186. DOI: 10.1109/ACCESS.2022.3190505

52. **Utkin L.V., Konstantinov A.V.** Attention-based random forest and contamination model. *arXiv:2201.02880*, 2022. DOI: 10.48550/arXiv.2201.02880

53. **Niu Z., Zhong G., Yu H.** A review on the attention mechanism of deep learning. *Neurocomputing*, 2021, Vol. 452, Pp. 48–62. DOI: 10.1016/j.neucom.2021.03.091

54. **Bahdanau D., Cho K., Bengio Y.** Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2014. DOI: 10.48550/arXiv.1409.0473

55. **Cai Z.** Weighted Nadaraya–Watson regression estimation. *Statistics & Probability Letters*, 2001, Vol. 51, No. 3, Pp. 307–318. DOI: 10.1016/S0167-7152(00)00172-3

56. **Chen X., Li D., Li Q., Li Z.** Nonparametric estimation of conditional quantile functions in the presence of irrelevant covariates. *Journal of Econometrics*, 2019, Vol. 212, No. 2, Pp. 433–450. DOI: 10.1016/j.jeconom.2019.04.037

57. **Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I.** Attention is all you need. *arXiv:1706.03762*, 2017. DOI: 10.48550/arXiv.1706.03762

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Ageev Andrey Yu.
Агеев Андрей Юрьевич
E-mail: andreyageev1@mail.ru

Utkin Lev V.
Уткин Лев Владимирович
E-mail: lev.utkin@gmail.com

Konstantinov Andrei V.
Константинов Андрей Владимирович
E-mail: andrue.konst@gmail.com

Submitted: 21.10.2024; Approved: 17.01.2025; Accepted: 12.03.2025.
Поступила: 21.10.2024; Одобрена: 17.01.2025; Принята: 12.03.2025.