Text Anomaly Detection with Simplified Isolation Kernel

Anonymous ACL submission

Abstract

Two-step approaches combining pre-trained large language model embeddings and anomaly detectors show good performance in text anomaly detection by leveraging rich semantic representations. However, high-dimensional dense embeddings extracted by large language models create challenges in substantial memory requirements and high computation time. To address this challenge, we introduce the Simplified Isolation Kernel (SIK), which maps high-dimensional dense embeddings to lower-dimensional sparse representations while preserving crucial anomaly characteristics. SIK has linear-time complexity and significantly reduces space complexity through its innovative boundary-focused feature mapping. Experiments across 7 datasets demonstrate that SIK achieves better detection performance than 11 SOTA anomaly detection algorithms while maintaining computational efficiency and low memory cost. All code and demonstrations are available at https://anonymous.4open.science/r/SIK-6577/.

1 Introduction

Text anomaly detection (TAD) plays a crucial role in many applications, including content moderation, fraud detection, and cybersecurity threat analysis (Pang et al., 2021). TAD involves identifying textual instances that significantly deviate from the norm, which could indicate potential security threats, novel information, or content requiring special attention (Cao et al., 2025b). With the exponential growth of digital text data, developing effective and efficient text anomaly detection methods has become increasingly important.

Text anomaly detection methods generally fall into two categories: end-to-end approaches and two-step approaches (Li et al., 2024). End-to-end approaches integrate representation learning and anomaly detection into unified frameworks. However, they require substantial data for each specific domain, demand complete retraining when deployed to new domains. Their poor generalization across different text corpora makes them impractical for many real-world scenarios where anomalies vary across contexts and domains (Malik et al., 2024).

Recent advances in large language models have created powerful embedding techniques that extract meaningful feature representations from various data types. These modular approaches to anomaly detection follow a two-step process (Li et al., 2024): 1) extracting dense vector embeddings that capture semantic relationships and contextual information from the raw data; 2) applying traditional anomaly detection algorithms on these embeddings. This approach leverages pre-trained models to directly extract features, eliminating the need for model retraining and significantly improving computational efficiency.

Isolation-based anomaly detection methods have demonstrated exceptional performance in text anomaly detection tasks (Cao et al., 2025b). The latest method Isolation Kernel (IK) (Ting et al., 2020) has also been widely applied to anomaly detection in time series, streaming data, and graph domains due to its data-dependent characteristics (Cao et al., 2024). However, IK requires mapping data to a high-dimensional space before performing anomaly detection, which significantly increases computational time and memory cost. The limitations become particularly problematic in the context of modern large language models (LLMs), where the embeddings extracted by LLMs already possess inherently high dimensionality, often reaching several hundred or thousand dimensions (Devlin et al., 2019).

To address these limitations, we propose the Simplified Isolation Kernel (SIK). SIK effectively maps high-dimensional dense embeddings to a lower-dimensional sparse representation while preserving crucial information for anomaly detection. The

core intuition of SIK is that for text anomaly detection tasks, we only need to focus on the dissimilarity between normal and anomalous samples, while the dissimilarity among normal samples can be ignored. The key contributions of our work are:

- Proposed Simplified Isolation Kernel (SIK), which has linear time complexity and is able to handle a training set with anomalies.
- A novel boundary-focused feature mapping that transforms high-dimensional dense text embeddings to a sparse, low-dimensional representation by capturing only essential boundary relationships, reduce memory cost significantly.
- Empirical evaluation demonstrates that SIK has better detection performance than existing methods across multiple domains and embeddings.

2 Related Work

2.1 Text Representations

The evolution of text representation techniques has been pivotal in advancing natural language processing. Early approaches such as TF-IDF (Salton and Buckley, 1988) created sparse vector representations that, while computationally efficient, failed to capture semantic relationships between words. This limitation was partially addressed by Word2Vec (Mikolov, 2013) and GloVe (Pennington et al., 2014), which generated dense continuous vector spaces based on word co-occurrence patterns, though they still assigned static representations regardless of contextual usage. The field subsequently progressed toward contextualized embeddings with ELMo (Peters et al., 2018) and transformer-based architectures like BERT (Devlin et al., 2019), which revolutionized NLP by using bidirectional attention mechanisms to produce context-sensitive representations.

The landscape of text representation was further transformed by the emergence of large language models (LLMs) exemplified by GPT (Brown et al., 2020). LLMs are trained on vast and diverse corpora, generating remarkably expressive embeddings that capture deep semantic relationships and generative text properties.

2.2 End-to-end TAD Approaches

End-to-end approaches integrate representation learning and anomaly detection into unified frameworks. Early neural methods primarily relied on autoencoder architectures to model normal text patterns, identifying anomalies through reconstruction errors (Manevitz and Yousef, 2007).

More recent innovations have shifted toward transformer-based architectures for text anomaly detection. CVDD (Ruff et al., 2019) detects anomalies by learning multiple context vectors through self-attention mechanisms on word embeddings, then identifying outliers based on the distance between text representations and these context vectors. DATE (Manolache et al., 2021) identifies replaced tokens and recognizes which masking pattern was applied to normal text, then scoring anomalies based on the model's uncertainty when processing unfamiliar patterns. FATE (Das et al., 2023) leverages a small number of labeled anomalous examples along with a deviation learning approach, where normal texts are pushed to match reference scores from a prior distribution while anomalous texts are forced to deviate significantly.

2.3 Two-step Approaches

Based on the text embeddings, traditional anomaly detection methods can be applied and they are classified into several distinct approaches, each with specific strengths for different data distributions and anomaly types.

Density-based methods like LOF (Breunig et al., 2000) identify outliers by measuring local density deviations relative to neighboring points. Isolation techniques, including iForest (Liu et al., 2008, 2012) and iNNE (Bandaragoda et al., 2018), operate on the principle that anomalies are sparse and distinctive, using space partitioning strategies where anomalous points require fewer partitions or are assigned to larger or out of hyperspheres.

Statistical approaches detect anomalies through their deviation from established data distributions, with ECOD (Li et al., 2022) utilizing cumulative distribution functions for efficient scoring and COPOD (Li et al., 2020) employing copulas to effectively model dependencies in multivariate scenarios. Meanwhile, deep learning methods have emerged as powerful tools for capturing complex, nonlinear patterns in data. Models such as Deep SVDD (Ruff et al., 2018) and LUNAR (Goodge et al., 2022) learn representations from normal instances and identify anomalies as significant deviations from these learned patterns, though they typically demand substantial training data and computational resources to achieve optimal performance.

3 Preliminaries

Table 1 presents the key symbols and notations used in this paper.

Table 1: Key symbols and notations

κ_I	Isolation Kernel
$\widehat{\mathcal{K}}_I$	Isolation Distributional Kernel
Φ	Feature map of Isolation Kernel
$\widehat{\Phi}$	Kernel mean map of Isolation Distributional Kernel
S_{IK}	Anomaly scores of Isolation Kernel
κ_S	Simplified Isolation Kernel
ϕ	Feature map of Simplified Isolation Kernel
S_{SIK}	Anomaly scores of Simplified Isolation Kernel

3.1 Problem Definition

Text anomalies are instances that significantly deviate from established patterns within a document collection. These anomalies may manifest as unusual topics, atypical linguistic structures, domain-specific terminology, or deliberately manipulated content such as spam, misinformation, or hate speech. Detecting such anomalies serves valuable purposes in content moderation, deception detection, and security surveillance.

Let $D = \{x_1, x_2, \ldots, x_N\}$ represent a collection of N text documents, where each document x_i is a sequence of lexical elements: $x_i = \{token_1, token_2, \ldots, token_{L_i}\}$, with L_i denoting the document's length in tokens.

The core objective in text anomaly detection is distinguishing D into two disjoint subsets: D_{normal} and $D_{\text{anomalous}}$, where $D_{\text{anomalous}}$ contains documents that substantially differ from the dominant patterns exhibited by $D_{\text{normal}} = D \setminus D_{\text{anomalous}}$.

3.2 Isolation Kernel (IK)

Isolation Kernel (IK) (Ting et al., 2018) is a data-driven kernel that derives directly from the dataset without a learning process. It has been used in many different anomaly detection application scenarios, including time series (Ting et al., 2022, 2024), streaming data (Cao et al., 2025a) and graphs (Zhuang et al., 2023), etc. The fundamental principle behind IK involves estimating the probability that two points will be assigned to the same partition through a data space partitioning strategy. Previous implementations of IK have utilized various partitioning mechanisms, including iForest (Ting et al., 2018), hypersphere (Ting et al., 2020), and Voronoi diagram (Qin et al., 2019) approaches. In this paper, we specifically employ the hypersphere partitioning strategy, with detailed methodological explanations provided in Section 4.1.

Let $D \subset \mathcal{X} \subseteq \mathbb{R}^d$ be a dataset sampled from an unknown distribution P_D, and $\mathbb{H}_\psi(D)$ denote the set of all partitionings H that are admissible from $\mathcal{D} \subset D$, where each sample point $z \in \mathcal{D}$ has an equal probability of being selected from $\mathcal{D}$, and $|\mathcal{D}| = \psi$.

The key idea of IK is to use ψ random sample points z to partition the data space, and the detailed partitioning strategy is provided in Section 4.1. The similarity between two points x and y is the times that both of them fall into the same partition $\theta[z]$ across t partitionings.

Definition 1 *(Ting et al., 2018; Qin et al., 2019) For any two points $x, y \in \mathbb{R}^d$, Isolation Kernel of x and y is defined to be the expectation taken over the probability distribution on all partitionings $H \in \mathbb{H}_\psi(D)$ that both x and y fall into the same isolating partition $\theta[z] \in H$, $z \in \mathcal{D} \subset D$:*

$$\begin{aligned}\kappa_I(x, y \mid D) &= \mathbb{E}_{\mathbb{H}_\psi(D)}[\mathbb{1}(x, y \in \theta[z] \mid \theta[z] \in H)] \\ &= \frac{1}{t}\langle \Phi(x), \Phi(y) \rangle, \end{aligned} \quad (1)$$

where $\mathbb{1}(\cdot)$ be an indicator function.

Definition 2 (Feature Map of IK) *(Ting et al., 2020) IK maps each point x to a $t \times \psi$ dimensions binary feature map $\Phi : x \mapsto \{0,1\}^{t\times\psi}$. Specifically, given t partitionings, for each partitioning H_i ($i = 1, \ldots, t$), IK creates a ψ-dimensional binary column vector $\Phi_i(x)$ where each dimension corresponds to one of the ψ partitions in H_i. The j-th component of this vector is:*

$$\Phi_{i,j}(x) = \mathbb{1}(x \in \theta_j \mid \theta_j \in H_i), \quad (2)$$

where $j = 1, \ldots, \psi$. This indicates whether point x falls inside partition θ_j in partitioning H_i. The final representation $\Phi(x)$ is the concatenation of all vectors: $\Phi_1(x), \ldots, \Phi_t(x)$.

3.3 Isolation Distributioal Kernel (IDK)

Based on the same framework of Kernel Mean Embedding (KME) (Muandet et al., 2017), IK has been used as the foundation to develop a distributional kernel called Isolation Distributional Kernel (IDK) (Ting et al., 2020). IDK specifically measures the similarity between two distributions rather than just between individual points. For text

anomaly detection, the anomaly score of each point can be computed by measuring the similarity between each point and the whole data distribution.

Definition 3 *Isolation Distributional Kernel of a point distribution $\mathcal{P}_x$ and a distribution $\mathcal{P}_Y$ is:*

$$\widehat{\mathcal{K}}_I(\mathcal{P}_x, \mathcal{P}_Y \mid D) = \frac{1}{t}\left\langle \Phi(\mathcal{P}_x|D), \widehat{\Phi}(\mathcal{P}_Y|D) \right\rangle, \tag{3}$$

where $\widehat{\Phi}(\mathcal{P}_Y|D) = \frac{1}{|Y|}\sum_{y\in Y}\Phi(y|D)$ is the kernel mean map.

4 Methodology

For text anomaly detection, we follow a two-step approach: text documents are first transformed into dense vector embeddings that capture semantic relationships. These embeddings can be generated using pre-trained language models, which encode contextual information and linguistic patterns. The quality of embeddings significantly influences downstream anomaly detection performance. Once documents are embedded in a high-dimensional space, traditional anomaly detection algorithms can be applied to identify outliers. However, these embeddings typically exist in high-dimensional spaces and applying anomaly detection algorithms to such high-dimensional data creates substantial computational challenges. We introduce Simplified Isolation Kernel (SIK) to address these challenges, the key steps are shown in the following subsections.

4.1 Space Partitioning

The proposed SIK employs a hypersphere-based space partitioning mechanism, following the same approach as used in iNNE (Bandaragoda et al., 2018) and IDK (Ting et al., 2020). The fundamental idea is to create a collection of hyperspheres that adapt to the local density of the data, which enables effective anomaly detection across regions of various densities.

Definition 4 (Hypersphere Partionings) *Each point $z \in \mathcal{D}$ is isolated from the rest of the points in $\mathcal{D}$ by building hyperspheres $\theta[z] \in H$ centered at z. The radius of this hypersphere is determined by the distance between z and its nearest neighbor in $\mathcal{D} \setminus \{z\}$. Each partitioning H consists of ψ hyperspheres and the region that is not covered by these hyperspheres. For stability, t different partitionings H_i, $i = 1 \dots t$ are generated, each based on a different random subset $\mathcal{D}_i \subset D$.*

This partitioning mechanism naturally adapts to the underlying data distribution. In dense regions, the resulting hyperspheres have short radii, since nearest neighbors are typically close. Conversely, in sparse regions, the hyperspheres have long radii because nearest neighbors are farther apart. This data-dependent property is crucial for effective anomaly detection, as it creates adaptive partitionings that can appropriately in both dense and sparse regions.

Unlike fixed-radii approaches that may struggle with varying data densities, this adaptive mechanism provides appropriate coverage across the entire feature space. It avoids overfitting in dense regions (where a fixed small radius would create too many partitions) and underfitting in sparse regions (where a fixed large radius might miss important structural details).

The first difference between IK and SIK lies in their fundamental approaches, despite using the same hypersphere partitioning mechanism. Both methods agree that points falling outside hyperspheres are more likely to be abnormal. However, IK focuses on pairwise similarity measurement, calculating how many times two points fall into the same specific hypersphere to determine their similarity. In contrast, SIK adopts a boundary-based perspective, constructing a decision boundary using multiple hyperspheres. In SIK, we assume that the higher the frequency of a point falling outside these boundaries, the more likely it is to be abnormal. SIK deliberately ignores the specific position of points within the boundaries, as this information is less relevant for anomaly detection purposes.

4.2 Feature Map

The second key difference between IK and our proposed SIK lies in their feature representation approaches. Figure 1 illustrates how both methods map data points differently in the Reproducing Kernel Hilbert Space (RKHS) $\mathscr{H}$.

While IK achieves linear time complexity, it creates high-dimensional feature representations by tracking exactly which hypersphere contains each point. This approach becomes problematic for large-scale text anomaly detection, where input data (e.g., 768-dimensional BERT embeddings) is already high-dimensional. IK feature mapping further expands this dimensionality, creating prohibitive memory requirements. Figure 1 left shows that IK maps all points to 3-dimensions ($\psi = 3$) space in one partitioning.

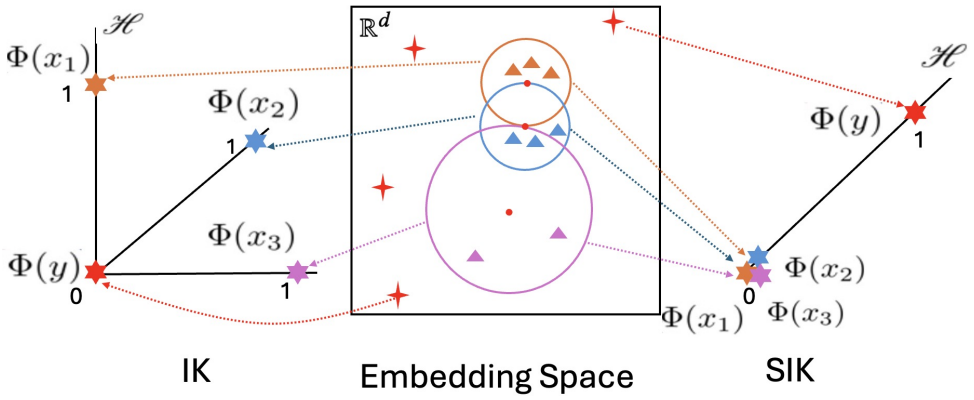


Figure 1: An illustration of feature maps of IK and SIK with one partitioning ($t = 1$) of 3 hyperspheres. Each center of a hypersphere is at a point $z \in D$ where $\psi = 3$ are randomly selected from the given dataset D. When a point x falls into an overlapping region, x is regarded as being in the hypersphere whose center is closer to x. On the left IK feature space, x has a 3-dimensional feature vector. On the right SIK feature space, x has only a 1-dimensional feature vector.

SIK addresses this limitation through a more compact feature representation, and the key insight is that anomaly detection doesn't require determining that a point falls into which specific hypersphere, but is sufficient to know whether a point falls into the boundary. This simplification significantly reduces the feature dimension while preserving the critical information needed for anomaly detection. Figure 1 right shows that all points are mapped to a 1-dimensional space, points in the boundary are mapped to the original of RKHS, and others are mapped to 1.

Definition 5 (Feature Map of SIK) *Given a point $x \in \mathbb{R}^d$, the feature map $\phi : x \mapsto \{0,1\}^t$ of SIK is a t-dimensional binary column vector, where each H_i ($i = 1, \ldots, t$) indicates whether x falls outside all hyperspheres $\theta \in H_i$:*

$$\phi_i(x) = \mathbb{1}(x \notin \theta | \theta \in H_i). \tag{4}$$

The SIK kernel function between two points x and y can be formally defined as:

$$\begin{aligned}\kappa_S(x,y) &= \mathbb{E}_{\mathbb{H}_\psi(D)}[\mathbb{1}(x,y \notin \theta \mid \theta \in H_i)] \\ &= \frac{1}{t}\sum_{i=1}^{t}[\mathbb{1}(x,y \notin \theta \mid \theta \in H_i)] \\ &= \frac{1}{t}\langle \phi(x), \phi(y)\rangle. \end{aligned} \tag{5}$$

Unlike traditional kernel functions that typically measure similarity between points, SIK quantifies how many times two points simultaneously fall outside all hyperspheres across multiple partitionings. When both points consistently fall outside all hyperspheres, they have a high SIK value. This characteristic allows us to compute anomaly scores by measuring the similarity between each point's feature vector and a reference anomaly vector (consisting of all ones), which forms the basis of our scoring method described in the next subsection.

4.3 Anomaly Scores Calculation

Based on the feature map of SIK, an ideal anomaly point $\mathcal{A}$ should fall outside all hyperspheres in all partitionings, where its vector will be $[1, \ldots, 1]$. Thus, the anomaly score of each point x can be defined as the similarity between the point x and the ideal reference anomaly point $\mathcal{A}$.

Definition 6 (Anomaly Score) *Given the binary feature representation $\phi(x) \in \{0,1\}^t$ for a point $x \in \mathbb{R}^d$, and let $\mathcal{A}$ be an reference anomaly point with $\phi(\mathcal{A}) = [1, \ldots, 1]$, the anomaly score is defined as:*

$$S_{SIK}(x) = \frac{1}{t}\langle \phi(x), \phi(\mathcal{A})\rangle, \tag{6}$$

the range of S_{SIK} is $[0,1]$ since $0 \le S_{SIK}(x) \le t$.

Under this formulation, points with scores approaching 1 after normalization) are more likely to be anomalies as they have higher similarity to the ideal anomaly, while normal points typically have scores closer to 0.

Since SIK feature map consists of 0 and 1, the anomaly score of point x are equivalent to the Hamming distance between $\phi(x)$ and the origin $[0, \ldots, 0]$, which quantifies the degree that a point is isolated from regions of normal data. In addition,

the anomaly scores can be equivalently expressed in terms of L_0 and L_1 norm as well.

The score calculation method can also be applied to the IK feature map:

$$S_{IK}(x) = 1 - \frac{1}{t} \parallel \Phi(x) \parallel, \tag{7}$$

where $\| \cdot \|$ can be either L0 or L1 norm.

It is worth noting that $S_{IK}(x)$ is equal to $S_{SIK}(x)$ since both methods essentially count how many times a point falls outside all hyperspheres across the t partitionings, directly measuring its degree of isolation from normal data regions.

4.4 Is SIK a Valid Kernel?

According to Mercer's theorem, a symmetric function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a valid kernel only if it is positive semi-definite and symmetric (Christmann and Steinwart, 2008). We demonstrate that SIK satisfies both requirements.

Based on Equation 5, for symmetry, we observe that:

$$\begin{aligned} \kappa_S(x, y) &= \frac{1}{t}\langle \phi(x), \phi(y) \rangle \\ &= \frac{1}{t}\langle \phi(y), \phi(x) \rangle \\ &= \kappa_S(y, x) \end{aligned} \tag{8}$$

This confirms SIK satisfies symmetry.

For positive semi-definiteness, Mercer's theorem requires that for any data points $x_1, ..., x_n \in \mathbb{R}^d$ and any real coefficients $\alpha_1, ..., \alpha_n \in \mathbb{R}$:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j \kappa_S(x_i, x_j) \geq 0 \tag{9}$$

By the properties of inner products and the fact that $\kappa_S(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, rewritten Equation 9 as:

$$\left\langle \sum_{i=1}^{n} \alpha_i \phi_S(x_i), \sum_{j=1}^{n} \alpha_j \phi(x_j) \right\rangle \geq 0 \tag{10}$$

Since both summations represent the same vector in feature space, this simplifies to:

$$\left\| \sum_{i=1}^{n} \alpha_i \phi(x_i) \right\|^2 \geq 0 \tag{11}$$

This inequality always holds since a squared norm is non negative. It is important to note that the feature map ϕ maps input points to binary vectors, which further supports the positive semi-definiteness property. Therefore, by Mercer's theorem, SIK is a valid kernel function.

5 Experiments

5.1 Experimental Setups

The experiments are conducted using the same benchmark datasets and embeddings from the benchmark NLP-ADBench (Li et al., 2024). The embeddings are extracted via BERT (Devlin et al., 2019) and OpenAI (text-embedding-3-large) (OpenAI, 2024) models as specified in NLP-ADBench. Statistical information of the experiment datasets is summarized in Table 2. AUROC (Area Under the Receiver Operating Characteristic Curve) is adopted as the evaluation metric. Each experiment is repeated 5 times with average results reported to mitigate randomness.

We utilized 8 traditional methods (LOF, iForest, ECOD, DeepSVDD, Autoencoder, LUNAR, INNE and IDK) sourced from the *PyOD* library (Zhao et al., 2019) on the extracted embeddings. The hyperparameter of nearest neighbors for LOF and LUNAR is searched in $\{5, 10, 20, 40\}$. For iForest, iNNE, IDK and SIK, ψ is searched in $\{32, 64, 128, 256, 512\}$ and with defalt $t = 200$. For Autoencoder and DeepSVDD, the hyperparameter of hidden neuron is searched in $\{[128, 64], [64, 32], [32, 16]\}$.

For comparative analysis, 3 end-to-end methods (CVDD, DATE and FATE) are included in this paper, and their performance is directly referenced from the NLP-ADBench (Li et al., 2024) due to the same datasets.

Table 2: Statistical information of datasets

Dataset	# Samples	# Ano.	% Ano.	Train	Test
EmailSpam	3578	146	4.08	2402	1176
SMSSpam	4969	144	2.89	3162	1510
BBCNews	1785	62	3.47	1206	579
AGNews	98207	3780	3.85	66098	32109
N24News	59822	1828	3.06	40595	19227
MovieReview	26369	1487	5.64	17417	8952
YelpReview	316924	17938	5.66	209290	107634

5.2 Empirical Evaluation

Table 3 presents the AUROC scores of all baseline methods across the 7 datasets.

With BERT embeddings, SIK performs best on Email_Spam, SMS_Spam, BBC_News and Movie_Review datasets. A Friedman-Nemenyi test (Demšar, 2006) in Figure 2a shows that SIK is top-ranked, only SIK and IDK are significantly better than DeepSVDD, ECOD and Forest, but SIK is much faster than IDK.

Table 3: Evaluation results across 7 datasets in terms of AUROC. SIK results are shown with a shadow background, and the best result on each dataset is in bold.

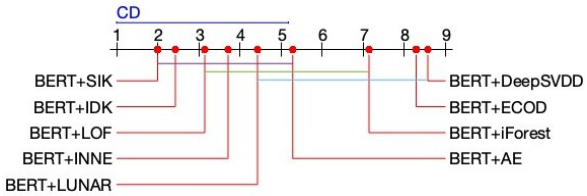
Algorithms	Email_Spam	SMS_Spam	BBC_News	AG_News	N24News	Movie	Yelp
CVDD	0.9340	0.4782	0.7221	0.6046	0.7507	0.4895	0.5345
DATE	0.9697	**0.9398**	0.9030	0.8120	0.7493	0.5185	0.6092
FATE	0.9061	0.6262	0.9310	0.7756	0.8073	0.5289	0.5945
BERT+LOF	0.7793	0.7642	0.9412	0.7643	0.6991	0.5253	0.6842
BERT+iForest	0.7599	0.6544	0.7394	0.6760	0.5804	0.4624	0.6222
BERT+ECOD	0.7427	0.6164	0.7302	0.6578	0.5363	0.4434	0.6204
BERT+DeepSVDD	0.6200	0.5765	0.6841	0.6290	0.5373	0.4732	0.6066
BERT+AE	0.8067	0.7526	0.9117	0.7491	0.6465	0.4975	0.6728
BERT+LUNAR	0.8340	0.7474	0.9404	0.7832	0.6589	0.4806	0.6721
BERT+INNE	0.8531	0.7528	0.9235	0.7761	0.6360	0.5125	0.6861
BERT+IDK	0.8649	0.7703	0.9473	0.7805	0.6625	0.5131	0.6829
BERT+SIK	0.8705	0.7719	0.9414	0.7755	0.6689	0.5264	0.6840
OpenAI+LOF	0.9726	0.9032	0.9671	0.9013	0.8160	0.6731	0.7694
OpenAI+iForest	0.5425	0.6131	0.6468	0.5364	0.5289	0.5886	0.5401
OpenAI+ECOD	0.8926	0.6155	0.7780	0.7260	0.6179	**0.6933**	0.7706
OpenAI+DeepSVDD	0.5291	0.5238	0.6010	0.5272	0.5885	0.5318	0.4808
OpenAI+AE	0.6826	0.7933	0.9645	0.8684	0.7504	0.6597	0.7568
OpenAI+LUNAR	0.9590	0.7855	0.9773	**0.9309**	0.8324	0.6781	**0.7984**
OpenAI+INNE	0.9727	0.8688	0.9833	0.8701	0.8067	0.6668	0.7367
OpenAI+IDK	0.9531	0.8615	0.9797	0.8855	0.8290	0.6290	0.6688
OpenAI+SIK	**0.9729**	0.8967	**0.9844**	0.8904	**0.8343**	0.6634	0.7345

SIK shows further performance improvements when applied to OpenAI embeddings, achieving the highest AUROC scores on several datasets, including Email_Spam, BBC_News, and N24News. The high-dimensional OpenAI embeddings contain more nuanced semantic information, which SIK successfully leverages for more accurate anomaly detection. Figure 2b shows that SIK is top-ranked and the performance of SIK has a critical difference from DeepSVDD and iForest, but IDK doesn't have.

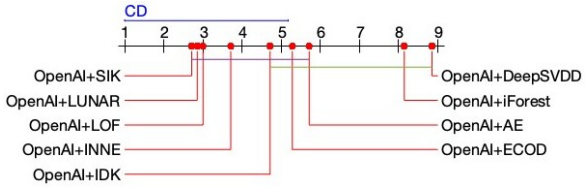
Compared with end-to-end approaches (CVDD, DATE, and FATE), the two-step approach with SIK usually demonstrates superior performance. For instance, on the BBC_News dataset, OpenAI+SIK significantly outperforms all 3 end-to-end methods. Figure 2c shows that SIK is the only detector that significantly better than CVDD.

Compared with other isolation-based approaches (iForest, iNNE and IDK), SIK maintains comparable or superior performance despite its reduced feature dimensionality. With OpenAI embeddings on the SMS_Spam dataset, SIK achieves higher AUROC than both iForest and IDK, indicating that the SIK preserves the essential discrimination information in the low-dimensional sparse map.

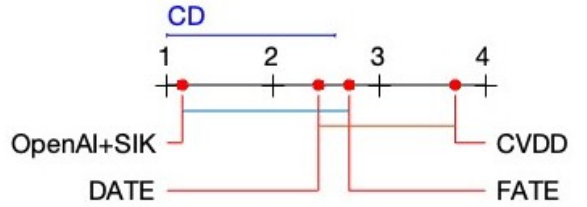
Since OpenAI-based methods perform better than end-to-end and BERT-based methods, we will focus on OpenAI-based methods in the following subsections.



(a) BERT-based



(b) OpenAI-based



(c) End-to-End

Figure 2: Friedman-Nemenyi test for the anomaly detection methods based on BERT, OpenAI embeddings and end-to-end at significance level 0.1 (the lower the better).

5.3 Scalability Analysis

Memory complexity: SIK achieves space efficiency improvements by focusing solely on boundary information. During training, SIK only needs to store hypersphere information rather than mapping the entire training dataset to feature spaces as required by IK. During testing, SIK reduces the feature representation dimensionality from ψt to just t, representing a significant reduction in space complexity from $O(nt\psi)$ to $O(nt)$, where the training set has n points.

Time complexity: The fundamental difference between SIK and IK emerges in how they process data during both the training and testing phases. During training, SIK directly calculates anomaly scores via norm computations, whereas IK requires mapping the entire dataset to compute KME, making SIK substantially faster.

During testing, while both SIK and IK have the same mapping complexity of $O(nt\psi)$, SIK's feature map dimensionality is only t compared to IK's ψt dimensions. This dimensional reduction translates to a testing complexity of only $O(nt)$ for SIK versus $O(nt\psi)$ for IK when computing similarities, resulting in computational savings, particularly for larger values of ψ. The overall time complexity is linear because $t\psi$ are hyperparameters and $t\psi \ll n$ for large datasets.

Table 4 presents the runtime and memory costs of both IK and SIK on the SMS_Spam dataset with the same hyperparameters $\psi = 256, t = 200$. SIK completes training approximately 14 times faster than IK while requiring dramatically less memory. During testing, memory savings remain substantial while time differences are less pronounced.

Although LOF and LUNAR demonstrate good performance on OpenAI embeddings, LOF's quadratic time complexity and LUNAR's computationally intensive deep learning approach result in significantly slower runtime compared to SIK.

Table 4: Time and memory comparison on SMS_Spam where $\psi = 256, t = 200$.

	Time (CPU seconds)		**Memory** (MB)	
	IDK	SIK	IDK	SIK
Train	115.4	8.2	1235.2	0.5
Test	46.3	45.6	589.8	2.3

5.4 Training with impure data

This section examines how robust anomaly detectors are against contamination in training data. Figure 3 illustrates the performance of both SIK and IDK on the Email_Spam dataset with increasing anomaly ratios from 1% to 5%. SIK exhibits a gradual decline in AUROC performance, while IDK maintains more stable performance. Despite this slight downward trend, SIK consistently maintains excellent detection capabilities with all values remaining above 0.95 AUROC.

The performance difference occurs despite both methods using identical hypersphere construction. SIK's decline stems from its reliance on binary inside/outside boundary decisions; when anomalies become hypersphere centers, they create spheres that erroneously encompass other anomalies, misclassifying them as normal. In contrast, IDK demonstrates greater robustness because it goes beyond simple boundary decisions by utilizing kernel mean embedding (KME), which averages feature representations across all training samples. This ensemble effect mitigates the negative impact of individual anomalous sphere centers, allowing IDK's similarity measurement to remain highly effective even when hyperspheres are distorted by contamination.

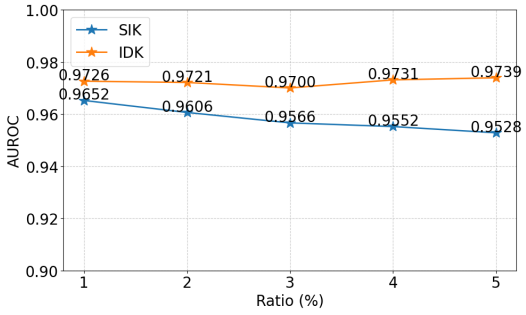


Figure 3: The performance of 5 runs of SIK on Email_Spam OpenAI embedding. The anomaly ratio is the ratio of the number of normal points and anomalies in the given dataset.

6 Conclusion

In this paper, we introduced the Simplified Isolation Kernel (SIK) for text anomaly detection. SIK effectively overcomes the computational and memory challenges posed by dense text embeddings from pre-trained LLMs by mapping high-dimensional dense embeddings to a low-dimensional sparse space that preserves boundary information. The key innovation of SIK lies in its boundary-focused feature mapping, which maintains a linear time complexity and significantly reduces the dimensionality of the feature representation.

Limitations

Although the proposed SIK has shown encouraging results in text anomaly detection, some issues remain for future consideration. While SIK was thoroughly compared with both end-to-end and two-step methods, we did not compare it with direct LLM reasoning approaches due to their significantly slower processing speed and output inconsistencies. Our attempts to use LLMs directly for anomaly detection produced results where the number of output labels frequently mismatched the test data quantity and could not be properly mapped to original text indices, preventing meaningful comparison.

Additionally, SIK demonstrates effective integration with LLM-generated embeddings, but its applicability to more nuanced domains such as legal, medical, or technical texts requires further investigation. Future work should also explore SIK's capability in detecting subtle anomalies that maintain similar semantic structures to normal text but contain misleading information or factual errors.

Ethic Statement

Data Sources and Usage: This study utilizes publicly available research datasets commonly referenced in NLP and anomaly detection literature. All datasets are properly cited throughout the paper. No private, proprietary, or personally identifiable information was included in our research.

Risks and Responsible Use: While anomaly detection technologies offer valuable capabilities for content moderation and security applications, we recognize they could potentially be misused for surveillance, censorship, or discriminatory filtering. We emphasize that SIK should be deployed responsibly with clear guidelines that respect privacy rights and freedom of expression. The technology presented in this paper is intended for research purposes and legitimate applications such as spam detection, fraud prevention, and identification of harmful content, not for arbitrary surveillance or censorship activities.

Use of AI Assistance: We acknowledge the use of AI-based writing assistants for grammatical refinement, spelling correction, and improving the clarity of our manuscript. However, all intellectual contributions, experimental designs, analyses, and conclusions in this paper are solely the work of the authors. The development of SIK, its implementation, experimentation, and evaluation were conducted exclusively by the authors without automated generation of scientific content.

References

Tharindu R Bandaragoda, Kai Ming Ting, David Albrecht, Fei Tony Liu, Ye Zhu, and Jonathan R Wells. 2018. Isolation-based anomaly detection using nearest-neighbor ensembles. *Computational Intelligence*, 34(4):968–998.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yang Cao, Yixiao Ma, Ye Zhu, and Kai Ming Ting. 2025a. Revisiting streaming anomaly detection: benchmark and evaluation. *Artificial Intelligence Review*, 58(1):1–24.

Yang Cao, Haolong Xiang, Hang Zhang, Ye Zhu, and Kai Ming Ting. 2024. Anomaly detection based on isolation mechanisms: A survey. *arXiv preprint arXiv:2403.10802*.

Yang Cao, Sikun Yang, Chen Li, Haolong Xiang, Lianyong Qi, Bo Liu, Rongsheng Li, and Ming Liu. 2025b. Tad-bench: A comprehensive benchmark for embedding-based text anomaly detection. *arXiv preprint arXiv:2501.11960*.

Andreas Christmann and Ingo Steinwart. 2008. Support vector machines.

Anindya Sundar Das, Aravind Ajay, Sriparna Saha, and Monowar Bhuyan. 2023. Few-shot anomaly detection in text with deviation learning. In *International Conference on Neural Information Processing*, pages 425–438. Springer.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Adam Goodge, Bryan Hooi, See-Kiong Ng, and Wee Siong Ng. 2022. Lunar: Unifying local outlier detection methods via graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6737–6745.

Yuangang Li, Jiaqi Li, Zhuo Xiao, Tiankai Yang, Yi Nian, Xiyang Hu, and Yue Zhao. 2024. Nlp-adbench: Nlp anomaly detection benchmark. *arXiv preprint arXiv:2412.04784*.

Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. 2020. Copod: copula-based outlier detection. In *2020 IEEE international conference on data mining (ICDM)*, pages 1118–1123. IEEE.

Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George H Chen. 2022. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12181–12193.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2012. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(1):1–39.

Jitendra Singh Malik, Hezhe Qiao, Guansong Pang, and Anton van den Hengel. 2024. Deep learning for hate speech detection: a comparative study. *International Journal of Data Science and Analytics*, pages 1–16.

Larry Manevitz and Malik Yousef. 2007. One-class document classification via neural networks. *Neurocomputing*, 70(7-9):1466–1481.

Andrei Manolache, Florin Brad, and Elena Burceanu. 2021. Date: Detecting anomalies in text via self-supervision of transformers. *arXiv preprint arXiv:2104.05591*.

Tomas Mikolov. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, and 1 others. 2017. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141.

OpenAI. 2024. New embedding models and api updates.

Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2):1–38.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Xiaoyu Qin, Kai Ming Ting, Ye Zhu, and Vincent CS Lee. 2019. Nearest-neighbour-induced isolation similarity and its impact on density-based clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4755–4762.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR.

Lukas Ruff, Yury Zemlyanskiy, Robert Vandermeulen, Thomas Schnake, and Marius Kloft. 2019. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4061–4071.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Kai Ming Ting, Zongyou Liu, Lei Gong, Hang Zhang, and Ye Zhu. 2024. A new distributional treatment for time series anomaly detection. *The VLDB Journal*, 33(3):753–780.

Kai Ming Ting, Zongyou Liu, Hang Zhang, and Ye Zhu. 2022. A new distributional treatment for time series and an anomaly detection investigation. *Proceedings of the VLDB Endowment*, 15(11):2321–2333.

Kai Ming Ting, Bi-Cun Xu, Takashi Washio, and Zhi-Hua Zhou. 2020. Isolation distributional kernel: A new tool for kernel based anomaly detection. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 198–206.

Kai Ming Ting, Yue Zhu, and Zhi-Hua Zhou. 2018. Isolation kernel and its effect on svm. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2329–2337.

Yue Zhao, Zain Nasrullah, and Zheng Li. 2019. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7.

Zhong Zhuang, Kai Ming Ting, Guansong Pang, and Shuaibin Song. 2023. Subgraph centralization: a necessary step for graph anomaly detection. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 703–711. SIAM.