Progress in Computing Applications

DLINE JOURNALS

Print ISSN: 2278 – 6465
Online ISSN: 2278 – 6473

PCA 2026: 15 (1)
https://doi.org/10.6025/pca/2026/15/1/30-43

Comparative Unsupervised Anomaly Detection in Mixed Code - Text Corpora Using Isolation Forest and PCA Autoencoders

Jun Wang
HONG KONG Chu Hai College
80 Castle Peak Bay Section of Castle Peak Road
Tuen Mun, New Territories. Hong Kong
Llger23423@aim.com

ABSTRACT

Anomaly detection (AD) in textual data remains challenging due to semantic complexity and the scarcity of labeled anomalous examples. This study proposes a comparative unsupervised framework to identify irregularities within mixed code text corpora, specifically addressing the heterogeneity of technical data from developer forums. We evaluate two distinct paradigms Isolation Forest (density-based isolation) and PCA Autoencoder (reconstruction based error) on a stratified subset of 10,000 Stack Overflow samples, derived from 52,270 total entries and represented via 500 dimensional TF–IDF features. Results indicate that while both methods identified 500 anomalies (5% contamination rate), they exhibited markedly different detection patterns. Isolation Forest showed a strong bias toward natural language, flagging 9.54% of text samples versus 0.08% of code samples. Conversely, the PCA Autoencoder detected a more balanced distribution, identifying 3.15% of code and 6.71% of text samples as anomalous. Agreement between methods was low (~3%), with only 15 consensus anomalies, suggesting complementary notions of irregularity rather than redundant signals. Isolation Forest captured global outliers, such as verbose questions, while the autoencoder detected local irregularities, such as unusual code constructs. These findings underscore the value of multi-perspective anomaly detection strategies for comprehensive coverage of irregular patterns. Ultimately, systematic anomaly analysis supports dataset cleaning, model robustness evaluation, and improved preprocessing for downstream tasks, including code understanding and automated question answering systems in software engineering applications

Keywords: Anomaly Detection, Unsupervised Learning, Isolation Forest, PCA Autoencoder, Mixed Code–Text Corpora, TF–IDF Vectorization, Consensus Evaluation, Software Engineering Analytics

Received: 5 October 2025, Revised 16 December 2025, Accepted 24 December 2025

Copyright: DLINE

1. Introduction

Anomaly detection (AD) is commonly formulated as an unsupervised learning task in which a model trained

on historical data is used to identify data instances that deviate significantly from expected patterns. AD has been successfully applied across a wide range of domains, including fraud detection [1], disease diagnosis [2] and intrusion detection [3]. In recent years, detecting anomalies in textual data has become increasingly important in applications such as content moderation, risk monitoring, and fraud analysis. However, textual anomaly detection remains particularly challenging due to the semantic richness of language and the limited availability of labeled anomalous examples in real world datasets.

Depending on the availability of labelled data, anomaly detection methods can be broadly categorised into supervised, semi-supervised (or weakly supervised), and unsupervised approaches. Supervised methods rely on labeled examples of both normal and anomalous data, which are often impractical to obtain. In contrast, unsupervised anomaly detection assumes no prior labeling information and aims to identify deviant samples solely based on intrinsic data characteristics, making it suitable for large, real-world corpora [4]. Nevertheless, unsupervised AD remains difficult when the majority class itself is highly heterogeneous, as is often the case in natural language data.

2. Review of Related Work

In the domain of anomaly detection, the one class classification paradigm utilizes a training dataset comprised exclusively of normal instances [5-13]. Conversely, weakly supervised approaches augment standard training data with abnormal samples [14, 15, 16, 17, 18, 19, 20]. Meanwhile, unsupervised methods operate on unlabeled datasets, typically predicated on the assumption that the majority of the data represents normal conditions [21, 22]. A critical limitation shared by both one class classification and unsupervised frameworks is their reliance on the assumption that the training data is predominantly normal; this limits their ability to discriminate against infrequent or novel scenarios. Although weakly supervised learning attempts to mitigate this issue by incorporating abnormal clips, the model remains vulnerable to anomalies not represented in the training distribution [23]. Regarding specific methodologies, Ait-Saada employs a Mixture Model approach to identify samples that deviate significantly from the corpus's underlying distributions. However, in contrast to domains such as computer vision, time series analysis, and numerical data processing, research on anomaly detection for textual data remains relatively scarce. Despite extensive research on anomaly detection in structured and numerical data, its application to textual data remains comparatively under explored. Early work by Cichosz [24] highlighted the potential of anomaly detection for text, while Maia [25] presented a comprehensive benchmarking study spanning representation strategies, learning paradigms, and linguistic diversity. More recent studies emphasize two fundamental challenges in text anomaly detection: manifold collapse in high dimensional embedding spaces and semantic entanglement, which obscures contextual irregularities and limits the separability of anomalous instances [26].

A variety of representation and modeling strategies have been explored to address these challenges. Traditional document term matrix representations, such as sparse bag of words models, have been employed with techniques like Non negative Matrix Factorization to isolate outlier components [27]. Sparse representations have also been used as inputs to One Class Support Vector Machines [28, 29] and shallow autoencoders. With the advent of distributed representations, word embeddings such as word2vec have been integrated with probabilistic models, including von Mises Fisher mixture models, to compute document level anomaly scores while penalizing overly general terms [30]. Similarly, pre trained fastText embeddings have been used within attention based neural architectures for one class anomaly detection [31].

More recently, deep end to end approaches have emerged that leverage transformer architectures without relying on external knowledge transfer. Manolache et al. [32] proposed an anomaly detection framework based on the ELECTRA architecture [33], employing adversarial training through generator and discriminator components and optimizing a token replacement objective. In parallel, Bakumenko et al. [34] introduced a novel approach for anomaly detection in financial datasets using Large Language Model (LLM) embeddings, demonstrating that pretrained sentence transformer models can effectively encode even non-semantic categorical attributes for downstream anomaly detection tasks.

3. Proposed Framework for Comparative Anomaly Detection

To address the challenges of anomaly detection (AD) in mixed code text corpora outlined in Section 2, this study proposes a comparative unsupervised framework. The primary objective is to evaluate the efficacy of distinct detection paradigms specifically density based isolation versus reconstruction based error in identifying irregularities within heterogeneous technical data. Given the scarcity of labeled anomalous examples in real world textual datasets [4], the framework operates under an unsupervised premise, relying solely on intrinsic data characteristics to distinguish deviant samples.

The proposed methodology consists of four sequential stages: data preprocessing and representation, model architecture selection, anomaly scoring and thresholding, and consensus based evaluation. Figure 1 illustrates the overall workflow of the proposed framework.

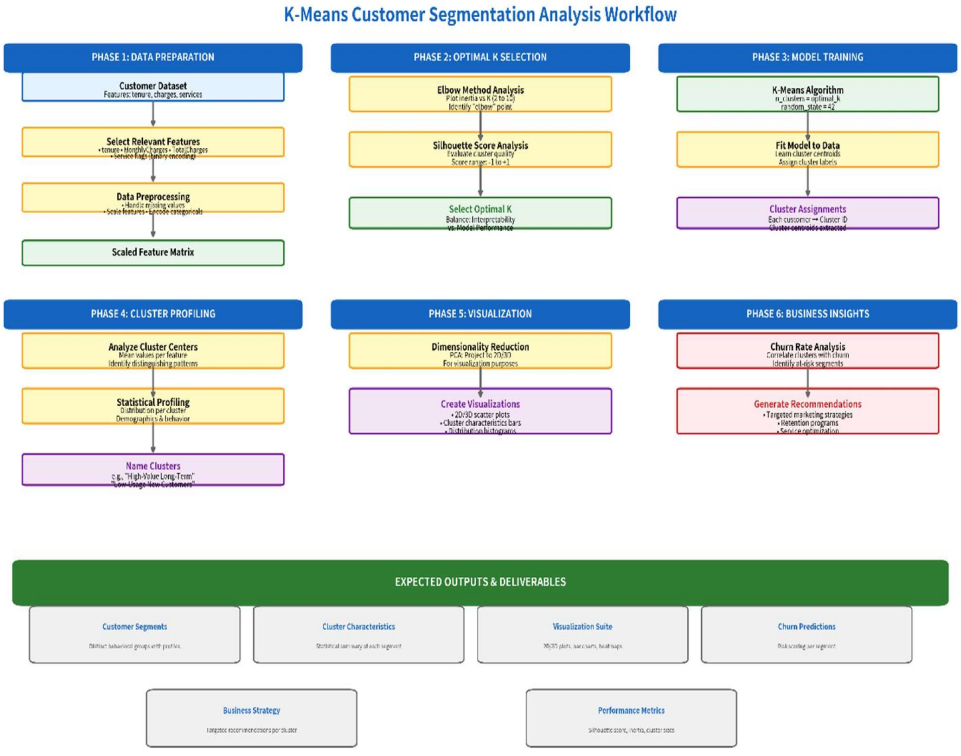

Figure 1. Workflow

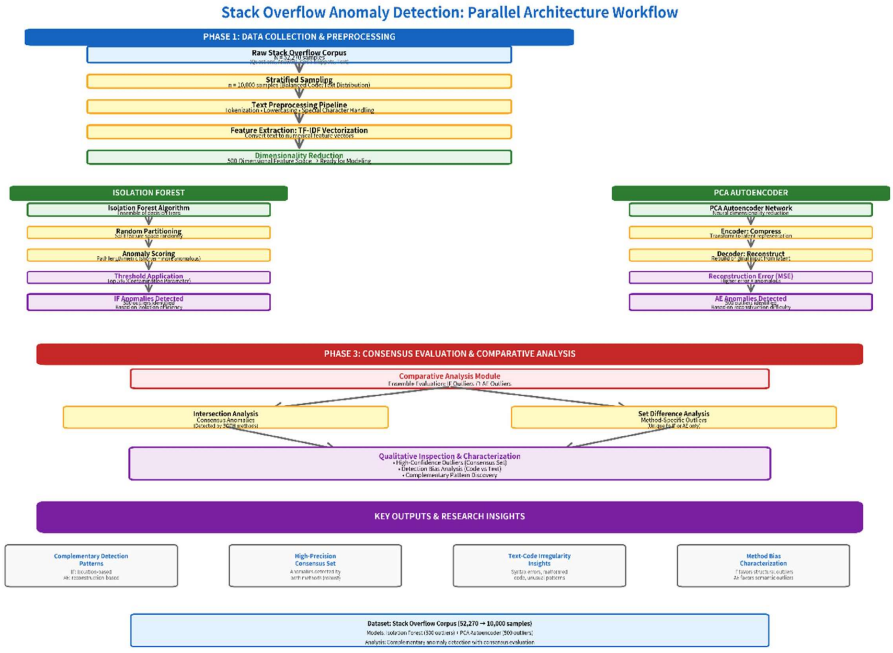

Figure 2. Parallel Architecture Workflow

3.1 Feature Representation

Effective anomaly detection in textual domains requires robust feature extraction that captures semantic and syntactic variations. While deep learning embeddings have gained prominence [30, 31], traditional statistical methods remain competitive for outlier detection in high dimensional sparse spaces. Accordingly, this study employs Term Frequency Inverse Document Frequency (TF–IDF) vectorization to transform raw text and code snippets into numerical representations. To mitigate the curse of dimensionality while retaining significant variance, the feature space is reduced to a 500-dimensional vector space. This representation balances computational efficiency with the preservation of lexical diversity necessary to distinguish between structured code and natural language text.

3.2 Model Architecture

Two distinct unsupervised algorithms are selected to provide complementary perspectives on data irregularity:

1. Isolation Forest (IF): As a density based approach, Isolation Forest isolates anomalies by randomly partitioning the feature space. It operates on the principle that anomalies are fewer and different, making them easier to isolate than normal instances [28]. This method is particularly effective at identifying global outliers that reside in sparse regions of the embedding space.

2. PCA Autoencoder: Conversely, the Principal Component Analysis (PCA) Autoencoder utilizes a reconstruction based approach. By compressing input data into a lower dimensional latent space and attempting to reconstruct the original input, the model learns the dominant patterns of the majority class. Anomalies are identified based on high reconstruction error, indicating that the sample deviates from the learned manifold [27]. This approach is sensitive to local irregularities and unusual feature combinations that may not necessarily be global outliers.

3.3 Anomaly Scoring and Thresholding

To ensure a fair comparative analysis, both models are configured to identify the top 5% of the dataset as anomalous. For the Isolation Forest, this is managed via a contamination parameter set to 0.05. For the PCA Autoencoder, anomalies are defined as samples with reconstruction errors in the top 5% of the distribution. This fixed contamination rate allows for a direct comparison of the *types* of instances flagged by each method, rather than discrepancies arising from varying sensitivity thresholds.

3.4 Consensus Evaluation Strategy

A critical component of this framework is the analysis of agreement between the two methods. Given that unsupervised AD lacks ground truth during training, validating detected anomalies is challenging. By examining the intersection of anomalies identified by both the Isolation Forest and the PCA Autoencoder, we establish a set of "consensus anomalies." These high-confidence outliers represent instances that are both statistically isolated and difficult to reconstruct, suggesting significant deviation from the underlying data distribution. This multi perspective strategy aims to overcome the limitations of single model approaches, which may fail to detect novel anomalies not represented in the training distribution [Kim, J.].

The following section details the specific dataset used to validate this framework, followed by an empirical analysis of the detection performance and the characteristics of the identified anomalies.

4. Dataset Description

The dataset used in this study is a labeled corpus derived from Stack Overflow posts, designed to distinguish between programming code snippets and natural language text. It consists of 52,270 individual samples, each representing a single line or segment extracted from Stack Overflow content.

Each sample is annotated with a binary class label, where label = 1 denotes code-related content and label = 0 denotes natural language text. The dataset is approximately balanced, with both classes represented in comparable proportions, enabling unbiased evaluation of analytical and machine learning methods.

4.1 Data Structure and Attributes

The dataset comprises three primary attributes:

- Text Content (line):

A raw textual field containing either programming code (e.g., function definitions, error traces, configuration snippets) or natural language text (e.g., questions, explanations, descriptive comments). The content varies significantly in length and structure, ranging from short phrases to long, multi line technical descriptions.

- Class Label (label):

A binary indicator specifying whether the text corresponds to code (1) or non code textual content (0). These labels enable supervised evaluation while also supporting unsupervised and exploratory analyses.

- Source Identifier (Source):

A categorical attribute indicating the origin of the data. All entries in the dataset originate from Stack Overflow, reflecting real world developer generated content.

4.2 Data Characteristics (Table 2)

The dataset exhibits substantial heterogeneity in vocabulary, syntax, and structure. Natural language samples demonstrate high lexical diversity and variable sentence structure, while code samples show more rigid syntactic patterns driven by programming language grammar. This contrast makes the dataset particularly suitable for studying text code discrimination, representation learning, and anomaly detection.

The absence of temporal ordering and the presence of mixed format text make the dataset representative of real world technical corpora used in software engineering research and machine learning applications.

4.3 Suitability for Analysis

Given its size, balance, and diversity, the choosn dataset we found that it is well suited for:

- Text and code classification tasks
- Embedding based representation learning
- Unsupervised anomaly and outlier detection
- Dataset quality assessment and noise analysis
- Robustness evaluation of NLP and code intelligence models

Overall, the dataset provides a realistic and challenging benchmark for analyzing structural and semantic differences between programming code and natural language text in developer forums.

5. Results

Embedding-based anomaly detection was performed on a stratified subset of 10,000 samples drawn from the full dataset of 52,270 Stack Overflow entries. The subset comprised 4,799 code samples (48.0%) and 5,201 natural language text samples (52.0%), represented using a 500-dimensional TF–IDF feature space.

5.1 Isolation Forest Results

The Isolation Forest model, configured with a contamination rate of 5%, identified 500 anomalous instances. A pronounced imbalance was observed between data types: only four anomalies were detected among code samples (0.08%), whereas 496 anomalies were identified in text samples (9.54%). The mean anomaly score across detected outliers was -0.3511.

Table 2 summarizes the comparative performance of Isolation Forest and a PCA-based autoencoder for unsupervised anomaly detection on a stratified subset of 10,000 Stack Overflow samples represented using 500-dimensional TF–IDF features. Both methods identify an equal number of anomalies (5%), but exhibit markedly different detection behavior. Isolation Forest is strongly biased toward natural language text, flagging only a negligible fraction of code samples, whereas the autoencoder detects a substantially higher proportion of anomalous code instances. The low overlap between methods (≈3%) indicates that they capture complementary notions of irregularity, with the small set of consensus anomalies representing high confidence

Attribute	Description
Data Source	Stack Overflow posts
Total Number of Samples	52,270
Unit of Analysis	Individual text line or segment
Classes	Code (label = 1), Natural Language Text (label = 0)
Class Distribution	Approximately balanced
Text Content	Mixed-format data including programming code snippets, error traces, questions, and explanations
Source Attribute	All samples originate from Stack Overflow
Temporal Information	Not available
Data Characteristics	High lexical and structural variability in text; syntactically constrained patterns in code
Primary Use Cases	Text–code classification, representation learning, anomaly detection, dataset quality assessment

Table 1. Dataset Description

outliers.

Method 1: Isolation Forest (contamination=5%)

Visualisation of anomaly score distributions shows clear separation between normal and anomalous samples, with outliers concentrated in sparse regions of the embedding space. The corresponding two-dimensional PCA projection highlights clusters of text anomalies located at the periphery of the dominant data manifold. Qualitative inspection reveals that these anomalies predominantly consist of long, verbose technical questions containing rare terminology and complex linguistic structures.

5.2 PCA Autoencoder Results

Using a PCA-based autoencoder, anomalies were defined as samples with reconstruction errors in the top 5% of the error distribution. This approach also identified 500 anomalous instances, but with a more balanced distribution: 151 anomalies in code samples (3.15%) and 349 in text samples (6.71%).

Metric	**Isolation Forest**	**PCA Autoencoder**
Subset Size Used	10,000 samples	
Feature Representation	500-dimensional TF–IDF vectors	
Detection Criterion error	Contamination = 5%	Top 5% reconstruction
Total Anomalies Detected	500	500
Code Anomalies	4 (0.08%)	151 (3.15%)
Text Anomalies	496 (9.54%)	349 (6.71%)
Mean Anomaly / Reconstruction Score	-0.3511 (anomaly score)	2.368 (anomalous samples)
Mean Normal Reconstruction Error	—	0.713
Consensus Anomalies (Both Methods)	\ multicolumn{2}{c}{15 (all text samples)}	
Agreement Rate	≈3%	

Table 2. Anomaly Detection Results Summary

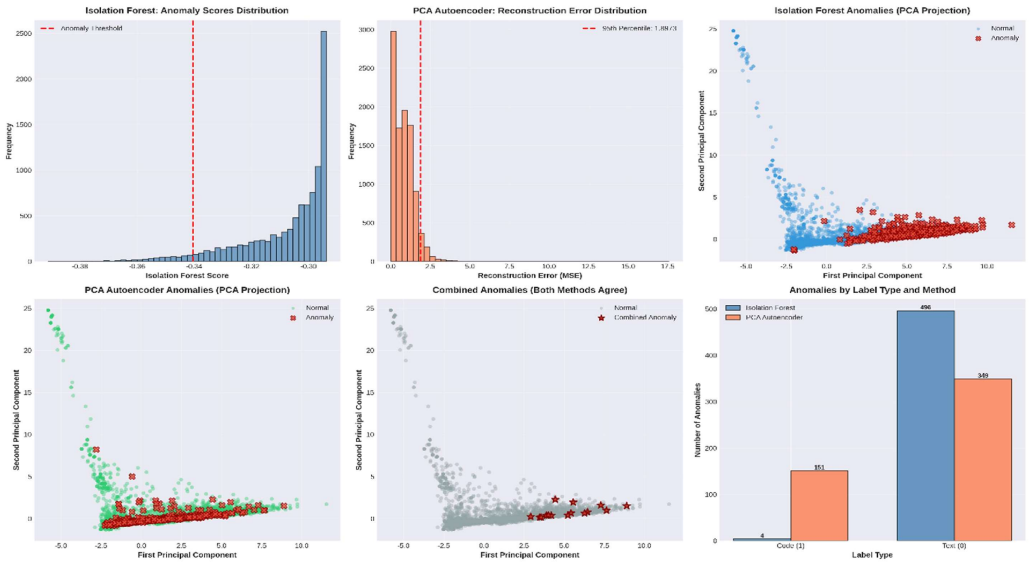

Figure 3. Isolation Forest

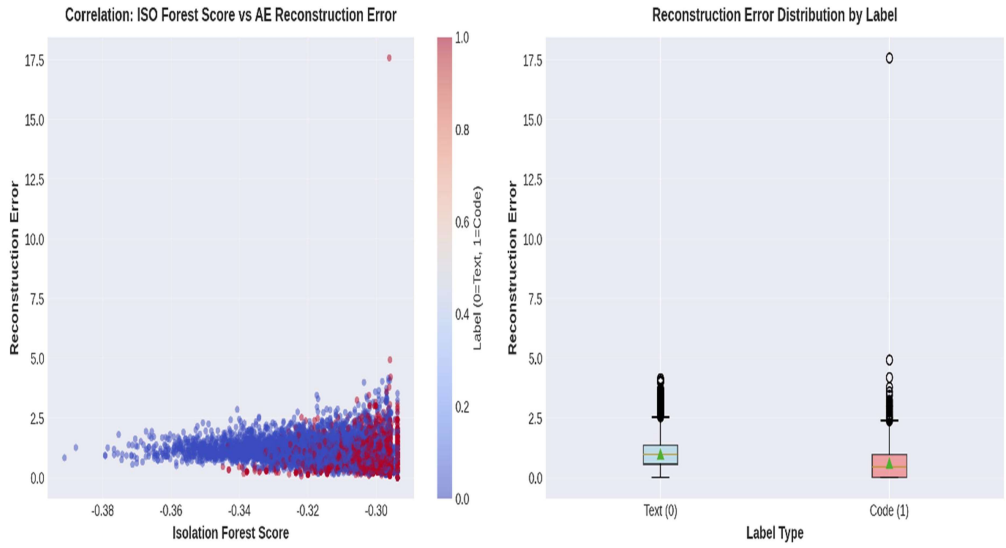

Figure 4. PCA Autoencoder

Method 2: PCA Autoencoder (top 5% reconstruction error)

The mean reconstruction error for anomalous samples was 2.368, compared to 0.713 for normal samples, yielding an error ratio of approximately 3.3. Reconstruction error distributions exhibit heavy tails, indicating the presence of rare feature combinations that deviate substantially from the dominant patterns. Autoencoder-detected anomalies include extremely short text fragments, irregular code constructs, and unusually long or repetitive code lines.

5.3 Consensus Anomalies

Only 15 samples were simultaneously flagged as anomalous by both Isolation Forest and the autoencoder, corresponding to an agreement rate of approximately 3%. All consensus anomalies belonged to the text category. These instances represent the most extreme outliers in the dataset and are characterized by long, structurally complex technical descriptions with uncommon vocabulary.

6. Discussion

The results demonstrate that different unsupervised anomaly detection techniques capture distinct and complementary aspects of irregularity in mixed text–code corpora. The strong bias of Isolation Forest toward text anomalies reflects the high lexical and structural variability inherent in natural language questions, whereas programming code exhibits more constrained syntactic patterns that form denser regions in the embedding space.

In contrast, the PCA autoencoder exhibits greater sensitivity to atypical code samples, indicating its ability to detect subtle deviations from dominant reconstruction patterns rather than relying solely on global density. This explains the higher proportion of code anomalies identified by the autoencoder compared to Isolation Forest, despite the code's overall structural regularity.

The low overlap between the two methods highlights the importance of multi-perspective anomaly detection. While a low agreement rate might initially appear undesirable, it instead suggests that each method identifies different classes of anomalies. Isolation Forest emphasizes global outliers, such as verbose or semantically rare text, whereas the autoencoder captures local irregularities, including unusual combinations of otherwise common features.

The small set of consensus anomalies comprises high confidence outliers and is particularly valuable for applications that require precision, such as dataset cleaning, manual inspection, or benchmarking model robustness. From an applied perspective, these findings suggest that a combined anomaly detection strategy offers superior coverage of irregular patterns than any single method alone.

Overall, the analysis confirms that embedding-based unsupervised anomaly detection is an effective approach for identifying edge cases, noise, and atypical content in large scale technical datasets. Such anomalies are likely to influence downstream tasks, including text classification, code understanding, and automated question-answering systems, underscoring the importance of systematic anomaly analysis during dataset preparation and evaluation.

7. Summary

This study presents a comparative unsupervised framework for anomaly detection in mixed code text corpora, addressing the challenge of identifying irregularities in heterogeneous technical data where labeled anomalous examples are scarce. The proposed methodology evaluates two distinct paradigms Isolation Forest (density-based isolation) and PCA Autoencoder (reconstruction based error) to detect deviations within a stratified subset of 10,000 Stack Overflow samples represented via 500-dimensional TF–IDF features. The dataset comprises approximately balanced code snippets and natural language text, enabling unbiased assessment of detection behaviors across modalities.

Results reveal that while both methods identified 500 anomalies (5% contamination rate), they exhibited markedly different detection patterns. Isolation Forest showed a strong bias toward natural language, flagging only 0.08% of code samples versus 9.54% of text samples. Conversely, the PCA Autoencoder detected a more balanced distribution, identifying 3.15% of code and 6.71% of text samples as anomalous. Qualitative analysis indicates that Isolation Forest primarily captures global outliers verbose, semantically rare technical questions whereas the autoencoder is sensitive to local irregularities, such as unusual code constructs or atypical feature combinations.

Notably, agreement between methods was low (~3%), with only 15 consensus anomalies, all belonging to the text category. This limited overlap suggests the approaches capture complementary notions of irregularity rather than redundant signals. The consensus set represents high confidence outliers characterized by structurally complex descriptions with uncommon vocabulary. The findings underscore the value of multi-perspective anomaly detection strategies for comprehensive coverage of irregular patterns in technical corpora. From an applied standpoint, such systematic anomaly analysis supports dataset cleaning, model robustness evaluation, and improved preprocessing for downstream tasks, including code understanding and automated question answering systems. Overall, the study confirms that embedding based unsupervised methods effectively identify edge cases in large-scale, heterogeneous textual datasets.

8. Conclusion

This study established a comparative framework for unsupervised anomaly detection within mixed code text corpora, addressing the critical challenge of identifying irregularities where labeled anomalous examples are scarce. By evaluating Isolation Forest and PCA Autoencoders on a stratified Stack Overflow dataset, we demonstrated that distinct detection paradigms capture complementary notions of irregularity. Isolation Forest exhibited a strong bias toward natural language text, identifying global outliers characterized by verbose, semantically rare questions. Conversely, the PCA Autoencoder displayed greater sensitivity to local irregularities, detecting a significantly higher proportion of anomalous code constructs alongside text anomalies.

The low agreement rate between methods (~3%) underscores the limitation of relying on single model approaches for heterogeneous data. However, the small set of consensus anomalies represents high confidence outliers, which are valuable for precision critical applications such as dataset cleaning and robustness benchmarking. These findings confirm that embedding based unsupervised methods effectively identify edge cases that could otherwise compromise downstream tasks such as code understanding and automated question-answering. The divergence in detection behavior suggests that natural language variability differs fundamentally from syntactic code deviations, requiring tailored modeling strategies.

Ultimately, this research highlights the necessity of multi-perspective anomaly detection strategies to achieve comprehensive coverage of irregular patterns in technical corpora. A combined strategy offers superior coverage to any single method alone, ensuring both global and local deviations are captured. Future work should explore integrating deep semantic embeddings to further refine detection sensitivity and validate these unsupervised findings against expert annotated ground truth. By systematically analyzing structural and semantic deviations, developers can enhance data quality and model reliability in software engineering applications, ensuring robust performance in real world deployment scenarios.

References

[1] Fraud. com. (2025). Anomaly detection for fraud prevention Advanced strategies, *In:* https://www.fraud.com/post/anomaly detection.

[2] Han, Y., Li, W., Liu, M., Wu, Z., Zhang, F., Liu, X., Tao, L., Li, X., Guo, X. (2021). Application of an Anomaly Detection Model to Screen for Ocular Diseases Using Color Retinal Fundus Images: Design and Evaluation Study, *J Med Internet Res* 2021 23(7) e27822.

[3] Luo, Y., Cheng, S., Liu, C., Jiang, F. (2018). PU Learning in Payload-based Web Anomaly Detection, *In:* Third International Conference on Security of Smart Cities, *Industrial Control System and Communications (SSIC), Shanghai, China,* 2018, p. 1-5.

[4] Mira, Ait-Saada., Mohamed, Nadif. (2023). Unsupervised Anomaly Detection in Multi-Topic Short-Text Corpora. *In:*Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1392–1403, Dubrovnik, Croatia. *Association for Computational Linguistics*.

[5] Zhao, B., Fei-Fei, L., Xing, E. P. (2011). Online detection of unusual events in videos via dynamic sparse coding. *In:*Proceedings of the CVPR 2011, *Colorado Springs,* CO, USA, 20–25 June p. 3313–3320.

[6] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., Davis, L. S. (2016). Learning temporal regularity in video sequences. *In:Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA,* 27–30 June 2016 p. 733–742.

[7] Cheng, K. W., Chen, Y. T., Fang, W. H. (2015). Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression. *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA*, 7–12 June 2015;p. 2909–2917.

[8] Qiao, M., Wang, T., Li, J., Li, C., Lin, Z., Snoussi, H. (2017). Abnormal event detection based on deep autoencoder fusing optical flow. *In:Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China*, 26–28 July p. 11098–11103.

[9] Xu, D., Yan, Y., Ricci, E., Sebe, N. (2017). Detecting anomalous events in videos by learning deep representations of appearance and motion. *Comput. Vis. Image Underst.* 2017, 156, 117–127.

[10] Zaheer, M. Z., Lee, J. h., Astrid, M., Lee, S. I. (2020). Old is gold: Redefining the adversarially learned one-class classifier training paradigm. *In:Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA,* 13–19 June 2020; p. 14183–14193.

[11] Wang, X., Che, Z., Jiang, B., Xiao, N., Yang, K., Tang, J., Ye, J., Wang, J., Qi, Q. (2021). Robust unsupervised video anomaly detection by multipath frame prediction. *IEEE Trans. Neural Networks Learn. Syst.* 2021, 33, 2301–2312.

[12] Georgescu, M. I., Ionescu, R. T., Khan, F. S., Popescu, M., Shah, M. (2021). A background-agnostic framework with adversarial training for abnormal event detection in video. *IEEE Trans. Pattern Anal. Mach. Intell.* 2021, 44, 4505–4523.

[13] Reiss, T., Hoshen, Y. (2022). Attribute Based Representations for Accurate and Interpretable Video Anomaly Detection. arXiv 2022, arXiv:2212.00789.

[14] Sultani, W., Chen, C., Shah, M. (2018). Real world anomaly detection in surveillance videos. *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Munich, Germany*, 8–14 September 2018; p. 6479–6488.

[15] Tian, Y., Pang, G., Chen, Y., Singh, R., Verjans, J. W., Carneiro, G. (2021). Weakly supervised video anomaly detection with robust temporal feature magnitude learning. *In:Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual,* 11–17 October 2021; pp. 4975–4986.

[16] Sapkota, H., Yu, Q. (2022). Bayesian nonparametric submodular video partition for robust anomaly detection. *In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA,* 18–24 June 2022; p. 3212–3221.

[17] Wu, P., Liu, J. (2021). Learning causal temporal relation and feature discrimination for anomaly detection. *IEEE Trans. Image Process.* 2021, 30, 3513–3527.

[18] Zhang, C., Li, G., Qi, Y., Wang, S., Qing, L., Huang, Q., Yang, M. H. (2023). Exploiting Completeness and Uncertainty of Pseudo Labels for Weakly Supervised Video Anomaly Detection. *In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada*, 18–22 June 2023; p. 16271–16280.

[19] Lv, H., Yue, Z., Sun, Q., Luo, B., Cui, Z., Zhang, H. (2023). Unbiased Multiple Instance Learning for Weakly Supervised Video Anomaly Detection. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada,* 18–22 June p. 8022–8031.

[20] Cho, M., Kim, M., Hwang, S., Park, C., Lee, K., Lee, S. (2023). Look Around for Anomalies: Weakly-Supervised Anomaly Detection via Context-Motion Relational Learning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada,* 18–22 June p. 12137–12146.

[21] Yu, G., Wang, S., Cai, Z., Liu, X., Xu, C., Wu, C. (2022). Deep anomaly discovery from unlabeled videos via normality advantage and self-paced refinement. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans,* LA, USA, 18–24 June 2022; p. 13987–13998.

[22] Tur, A. O., Dall'Asen, N., Beyan, C., Ricci, E. (2023). Exploring Diffusion Models for Unsupervised Video Anomaly Detection. arXiv 2023, arXiv:2304.05841.

[23] Kim, J., Yoon, S., Choi, T., Sull, S. (2023). Unsupervised Video Anomaly Detection Based on Similarity with Predefined Text Descriptions. *Sensors* 23, 6256.

[24] Cichosz, P. (2020). Unsupervised modeling anomaly detection in discussion forums posts using global vectors for text representation. *Natural Language Engineering*. 26(5) 551-578.

[25] Maia, Fabio Masaracchia; COSTA, Anna Helena Reali., Learning with Few (2025). A Comparative Study of Multilingual Text Anomaly Detection. *In:* Simpósio Brasileiro De Tecnologia Da Informação E Da Linguagem Humana (Stil), 16. , 2025, Fortaleza/CE. Anais. Porto Alegre: *Sociedade Brasileira de Computação,* 2025. p. 233-246.

[26] Jeremie, Pantin., Christophe, Marsala. (2025). A Robust Autoencoder Ensemble-Based Approach for Anomaly Detection in Text, *Procedia Computer Science,* Volume 270,. Pages 1986-1995.

[27] Ramakrishnan, Kannan., Hyenkyun, Woo., Charu, C. Aggarwal, and Haesun Park. Outlier Detection for Text Data, pages 489–497.

[28] Larry, M., Manevitz, Malik., Yousef. (2001). One-class svms for document classification. *Journal of machine Learning research*, 2(Dec) 139–154.

[29] Bernhard, Schölkopf., John, C. (2001). Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the Support of a High Dimensional Distribution. *Neural Computation*, 13(7) 1443–1471.

[30] Honglei, Zhuang., Chi, Wang., Fangbo, Tao., Lance, Kaplan., Jiawei, Han. (2017). Identifying semantically deviating outlier documents. *In: Proceedings of the Conference on Empirical Methods in Natural Language Processing,* pages 2748–2757.

[31] Lukas, Ruff., Yury, Zemlyanskiy., Robert, Vandermeulen., Thomas, Schnake., Marius, Kloft. (2019). Selfattentive, 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.

[32] Andrei, Manolache., Florin, Brad., Elena, Burceanu. (2021). DATE: Detecting anomalies in text via selfsupervision of transformers. *In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pages 267–277.

[33] Kevin, Clark., Minh-Thang, Luong., Quoc, V., Le, Christopher, D., Manning. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *In*: *International Conference on Learning Representations*.

[34] Bakumenko, A., Hlaváèková-Schindler, K., Plant, C., Hubig, N. C. (2025). Advancing Anomaly Detection: Non-Semantic Financial Data Encoding With Large Language Models, *IEEE Access*, vol. 13, p. 146757-146771.