JCTA Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

A Graph-Augmented Isolation Forest Using Node2Vec and GraphSAGE for Mobile User Behavior Anomaly Detection

Amaka Patience Binitie [1,*], Sunny Innocent Onyemenem [1], Nneamaka Christiana Anujeonye [1], Arnold Adimabua Ojugo [2], Francesca Avwuru Egbokhare [3], and Tabitha Chukwudi Aghaunor [4]

[1] Department of Computer Science, School of Science, Federal College of Education (Technical), Asaba 320213, Nigeria; e-mail: amaka.binitie@fcetasaba.edu.ng; innocentsunnyonyemenem@gmail.com; anujeonyechristhy@gmail.com
[2] Department of Computer Science, Federal University of Petroleum Resources, Effurun 330102, Nigeria; e-mail: ojugo.arnold@fupre.edu.ng
[3] Department of Computer Science, Faculty of Physical Science, University of Benin, Benin 300283, Nigeria; e-mail: fegbokhare@uniben.edu
[4] School of Data Intelligence and Technology, Robert Morris University, Pittsburgh, PA 15108, United States of America; e-mail: tabitha.aghaunor@gmail.com
* Corresponding Author: Amaka Patience Binitie

Abstract: This study presents a Graph-Augmented Isolation Forest (GAIF), an unsupervised anomaly-detection framework for analyzing mobile user behavior. The proposed framework represents users and behavioral attributes as a user–feature bipartite graph, enabling the capture of relational dependencies that are not explicitly modeled in conventional vector-based approaches. Low-dimensional user representations are learned through Node2Vec and Graph Sample and Aggregate (GraphSAGE), and the resulting embeddings are subsequently processed by an Isolation Forest to produce anomaly scores. Experiments are conducted on a Mobile Device Usage and User Behavior dataset comprising 700 user profiles derived from application-level behavioral indicators. The dataset is treated as a behavioral abstraction rather than as a malware classification benchmark. A consistent 80:20 stratified train–test split is employed, with all learning-capable operations restricted to the training data to mitigate information leakage. Detection performance is evaluated post hoc using precision, recall, F1-score, and area under the curve (AUC) metrics. Under the evaluated setting, GAIF achieves an F1-score of 0.94 and an AUC of 0.97, demonstrating improved anomaly detection effectiveness relative to representative unsupervised baseline methods. These results are obtained on a static, proxy dataset and should not be interpreted as evidence of real-time deployment capability. Model interpretability is supported through post-hoc Uniform Manifold Approximation and Projection (UMAP) visualizations of the learned embeddings, providing structural insights into anomalous user behavior. Overall, the findings indicate that integrating graph-based representation learning with isolation-based anomaly scoring constitutes a computationally efficient approach for unsupervised mobile user behavior anomaly detection within the scope of this study.

Keywords: Bipartite graph modeling; Cybersecurity analytics; Graph-based learning; Isolation forest; Mobile user behavior analysis; Unsupervised anomaly detection; User behavior modeling; User–feature graph.

Received: January, 6th 2026
Revised: January, 30th 2026
Accepted: February, 2nd 2026
Published: February, 3rd 2026

1. Introduction

The rapid proliferation of mobile devices and applications has fundamentally transformed the way users interact with digital services. As mobile platforms increasingly manage sensitive personal, financial, and communication data, abnormal patterns of user behavior have emerged as potential indicators of cybersecurity threats, including fraud, identity theft, and account compromise [1]. Traditional detection mechanisms, which primarily rely on static signatures, permission analysis, or heuristic rules, are often insufficient for identifying sophisticated or previously unseen behavioral anomalies [2], [3]. This limitation has motivated

growing interest in machine learning and data-driven approaches for anomaly detection in mobile user behavior analytics [4], [5].

In this study, mobile user behavior anomaly refers to user-level deviations from established patterns in aggregated application usage, resource consumption, and device interaction profiles. The objective of mobile user behavior anomaly detection is to identify atypical user interaction patterns, such as unusual application usage behavior, network usage characteristics, session duration distributions, or device configuration profiles [1], [2]. These interactions inherently form complex dependencies among users and behavioral attributes, which are poorly captured by conventional vector-based representations that assume feature independence [6]–[8]. This structural limitation constrains the effectiveness of widely used unsupervised anomaly detection methods, such as Isolation Forest (IF), Local Outlier Factor (LOF), and One-Class Support Vector Machine (OC-SVM), particularly when anomalies manifest only within the relational context of user–feature interactions [9]–[12].

Recent advances in graph-based anomaly detection have demonstrated the advantages of explicitly modeling relational and structural dependencies for identifying atypical entities [13], [14]. In particular, bipartite graph representations have proven effective for modeling interactions between two distinct node types, such as users and features, enabling the detection of structural anomalies across heterogeneous data sources [15], [16]. By capturing node connectivity and relational context, graph-based embeddings often provide richer representations than flat feature vectors, thereby facilitating the detection of contextual and structural anomalies [17], [18]. However, these benefits frequently come at the cost of increased computational complexity, sensitivity to hyperparameter choices, and limited interpretability, which can hinder their suitability for computationally efficient and scalable anomaly detection under resource-constrained or label-scarce settings.

Parallel to these developments, graph representation learning (GRL) methods have been introduced to encode nodes, edges, and neighborhood structures into low-dimensional embeddings that preserve relational information [3], [4]. Such embeddings enable anomaly detection systems to exploit structural and topological characteristics beyond raw behavioral features. For example, group anomaly detection frameworks have employed spatiotemporal convex-hull models to identify irregular application usage patterns in mobile network environments [5]. Nevertheless, while graph-based models effectively capture relational context, they often require substantial computational resources and, in some cases, large labeled datasets for training [4]. In contrast, tree-based ensemble methods such as Isolation Forest offer scalability, robustness, and interpretability, but lack awareness of relational structure.

This contrast reveals a clear methodological gap between vector-based unsupervised anomaly detectors and graph-only anomaly detection approaches. Vector-based methods, including Isolation Forest, LOF, and OC-SVM, are computationally efficient and scalable, yet they fail to capture relational dependencies among behavioral features, limiting their ability to detect contextual and structural anomalies. Conversely, graph-based anomaly detection methods explicitly model relational structure but often incur higher computational costs, require careful parameter tuning, and may exhibit limited stability on mid-sized behavioral datasets. Graph-based modeling of relational contagion has also shown promise in behavioral analysis domains. For instance, CoSoGMIR, a social graph contagion diffusion framework incorporating movement–interaction–return dynamics, has been proposed to model anomalous propagation patterns, underscoring the importance of relational modeling in detecting behavioral deviations within networked systems [19]. Despite these advances, relatively few studies have explored the integration of relational modeling, anomaly isolation, and computational efficiency within a unified unsupervised framework for mobile user behavior analysis.

To address this gap, Graph-Augmented Isolation Forest (GAIF) is proposed as an unsupervised anomaly detection pipeline that integrates graph representation learning with isolation-based anomaly scoring. GAIF models mobile user behavior as a user–feature bipartite graph, enabling the explicit encoding of relational dependencies that are absent in conventional feature-vector representations. Low-dimensional embeddings are learned using Node2Vec [20], which captures global structural patterns, and Graph Sample and Aggregate (GraphSAGE) [21], which aggregates local neighborhood information in an inductive manner. These enriched representations are subsequently processed by an Isolation Forest to assign anomaly scores, combining relational awareness with computational efficiency.

The central hypothesis of this study is that integrating isolation-based anomaly detection with graph-derived embeddings that encode user–feature relational structures can improve

the identification of mobile user behavior anomalies, compared with approaches that rely solely on vector-based representations or graph modeling in isolation. This work is situated within a broader body of research on cybersecurity analytics, which emphasizes the need for adaptive and intelligent analytical methods [22], [23]. Prior studies have addressed challenges such as securing mobile transactions [24], mitigating interface-based attacks [25], and detecting phishing threats [26]. In contrast, GAIF is not intended as a deployed security mechanism, but rather as an offline analytical framework for highlighting suspicious user profiles that warrant further investigation, thereby complementing existing security and monitoring systems.

The mobile user behavior data considered in this study is inherently relational. The focus is on identifying contextual and structural anomalies at the user level, such as users whose behavioral profiles exhibit atypical relational patterns within a network of users and features, rather than point anomalies in individual network packets or application-specific malware signatures [22], [27]. Accordingly, this study does not address malware classification, intrusion detection, or real-time event monitoring. Instead, it targets the problem of profiling users based on holistic behavioral abstractions to flag deviations from learned normative patterns. The proposed framework is evaluated using the publicly available Mobile Device Usage and User Behavior Dataset [23], which is derived from the Drebin Android malware corpus. The dataset has been curated to represent user-level behavioral abstractions rather than malware signatures, comprising 700 synthetic user profiles characterized by 11 application-level behavioral indicators, such as application usage intensity and data consumption patterns.

The main contributions of this paper are summarized as follows:

- An unsupervised, graph-aware anomaly detection framework that integrates relational user–feature structures with isolation-based scoring for mobile user behavior analysis.
- A unified pipeline combining offline chi-square feature filtering, user–feature bipartite graph modeling, dual graph embeddings using Node2Vec and GraphSAGE, and isolation-based anomaly scoring.
- An empirical evaluation of the proposed framework against representative unsupervised anomaly detection baselines on a public mobile user behavior dataset.
- An embedding-level interpretability analysis based on Uniform Manifold Approximation and Projection (UMAP) to provide structural insights into anomalous user behavior.

The remainder of this paper is organized as follows. Section 2 reviews related work in anomaly detection and graph learning. Section 3 describes the GAIF methodology, data preprocessing, and experimental setup. Section 4 presents experimental results, ablation studies, and comparative analyses. Section 5 concludes the paper by summarizing key findings, discussing limitations, and outlining directions for future research.

2. Related Works

The rapid expansion of mobile applications and digital services has positioned smartphones as central components of modern digital ecosystems, generating large volumes of behavioral data that reflect user interactions, contextual usage patterns, and system states. Detecting anomalies within such data remains a challenging task due to high feature heterogeneity, dynamic behavioral contexts, and the limited availability of labeled data in real-world environments [3], [6]. Mobile user behavior anomaly detection seeks to identify deviations from expected usage patterns, such as abnormal battery consumption, excessive application activity, or unusual network behavior, which may indicate security threats or system misuse [13], [14].

Early studies in mobile anomaly detection primarily relied on conventional machine learning techniques, including K-Means clustering, LOF, and Gaussian Mixture Models (GMM), to identify irregular user behavior patterns [14], [22], [28]–[30]. While these methods offered computational simplicity, they often struggled with high-dimensional data and failed to capture inter-feature dependencies that are critical for modeling complex user behavior. Subsequent approaches introduced more advanced models, such as the OC-SVM [31] and Random Forest-based anomaly detectors, which improved detection sensitivity but required extensive hyperparameter tuning and exhibited scalability limitations in dynamic mobile environments [32].

The adoption of deep learning techniques further advanced mobile anomaly detection research. Models based on Autoencoders (AEs), Variational Autoencoders (VAEs), and

Recurrent Neural Networks (RNNs) were applied to learn latent representations of normal user behavior and identify deviations as anomalies [10], [33]. Although these approaches improved detection accuracy and enabled temporal modeling, they were often computationally expensive and lacked interpretability, particularly when applied to mid-sized datasets such as Drebin [2], [34], [35]. Moreover, most deep learning-based methods continued to rely on vector-based representations, implicitly assuming feature independence and overlooking relational dependencies among user behavior attributes.

Recent work has demonstrated the effectiveness of graph-based learning in addressing these limitations by explicitly modeling relationships among users and behavioral features. Graph embedding techniques have been applied to user interaction networks for anomaly detection, enabling the capture of both structural and attribute information [36]. Surveys of graph-based anomaly detection methods emphasize the importance of integrating topological structure with node attributes to identify complex anomalous patterns in graph data [18], [37]. Representative embedding methods, including Node2Vec [20], GraphSAGE [21], and Deep Graph Infomax (DGI) [38], have been shown to encode local and global structural information into low-dimensional representations, facilitating more effective anomaly detection [6], [34]. Applications of such methods include temporal graph-based detection in mobile networks [27] and mobility-trajectory modeling for behavioral anomaly identification [22]. Despite these advances, most graph-based approaches remain computationally intensive, sensitive to hyperparameter choices, or dependent on labeled data, limiting their applicability in scalable and unsupervised mobile behavior analysis settings [36], [39]–[42].

In parallel, the Isolation Forest (IF) algorithm has been widely adopted as an efficient and scalable unsupervised anomaly detection method [9], [43]. IF identifies anomalies through recursive random partitioning rather than explicit modeling of normal behavior, making it well suited for high-dimensional datasets such as Drebin. Its non-parametric structure and relatively low computational cost contribute to its robustness and practical appeal. However, conventional IF operates on flat feature vectors and treats behavioral attributes as independent, thereby failing to capture relational dependencies among user behaviors and limiting its interpretability in relational settings [9], [44].

Despite substantial progress across these research directions, existing anomaly detection systems continue to face persistent challenges, including limited interpretability in deep learning models, insufficient modeling of relational dependencies, and reduced robustness in high-dimensional or weakly imbalanced data scenarios [45]–[49]. These challenges are particularly pronounced in mobile user behavior analysis, where user actions are inherently relational and context dependent.

More recent graph neural network-based anomaly detection frameworks, such as Anomal-E [50] and hybrid graph neural network (GNN) transformation models [51], further demonstrate the importance of graph structures for capturing complex relational patterns. For instance, GNN-assisted anomaly detection has been shown to improve real-time fraud detection in fintech systems [52], while lightweight GNN-based models have been proposed for anomaly detection in cognitive wireless sensor networks [53]. Hybrid node representation learning approaches have also been shown to enhance graph anomaly detection by capturing complementary structural signals [17]. Comprehensive surveys highlight the increasing adoption of hybrid graph-based anomaly detection techniques across both static and temporal domains [54]. Additionally, graph clustering frameworks have been proposed to expose anomalous relational patterns in user interaction networks that are not detectable using traditional vector-based methods [55]. Nevertheless, most existing approaches focus on homogeneous or domain-specific interaction graphs, such as transaction or sensor networks, and do not explicitly model bipartite user–feature relationships, leaving a gap in unsupervised user-level mobile behavioral anomaly detection.

The proposed GAIF framework is designed to address this methodological gap by integrating relational modeling and isolation-based anomaly detection within a unified unsupervised pipeline. GAIF encodes mobile user behavior as a user–feature bipartite graph, enabling relational dependencies to be incorporated into anomaly scoring. By combining chi-square-based offline feature filtering, graph-based representation learning, and Isolation Forest detection, GAIF aims to balance representational richness, computational efficiency, and interpretability. This integrated design supports dimensionality reduction, context-aware behavioral representation, and scalable anomaly detection, positioning GAIF as a methodologically grounded approach for mobile user behavior anomaly analysis.

3. Materials and Methods

This study proposes the GAIF framework, which enhances isolation-based anomaly detection through graph-based representation learning. GAIF is designed to operate in an unsupervised setting, explicitly capture relational structure in mobile user behavior data, and enforce a strict separation between training and evaluation stages to prevent information leakage. The framework comprises six sequential stages: data abstraction, preprocessing, offline feature relevance assessment, train–test splitting, graph construction and representation learning, and isolation-based anomaly scoring. An overview of the complete pipeline is shown in Figure 1. Each subsection below describes a corresponding stage of the GAIF pipeline, from data preparation to anomaly interpretation.

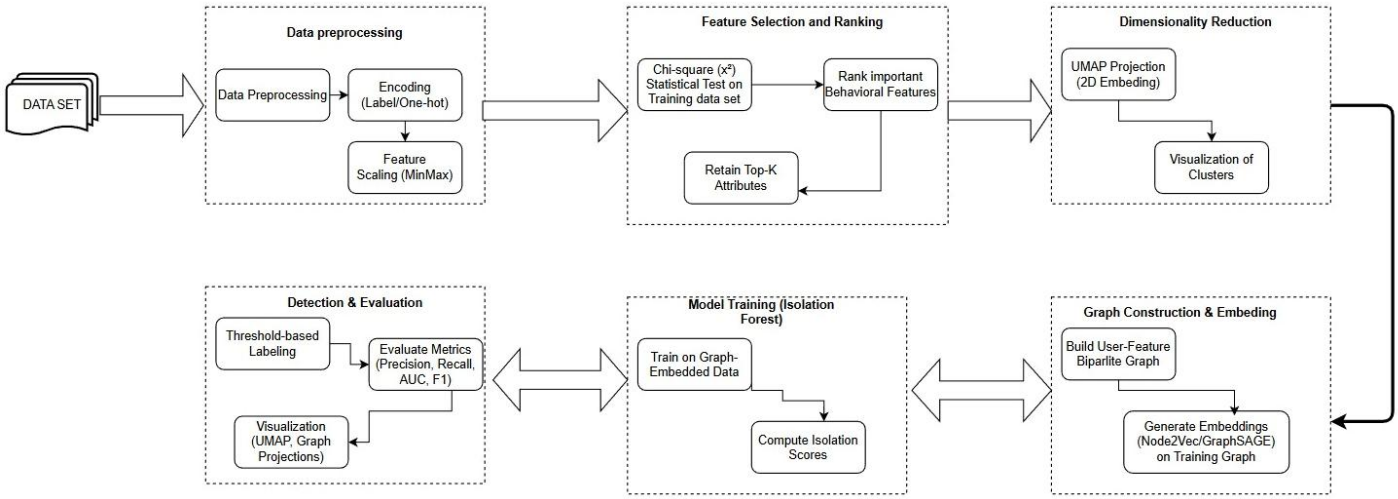

Figure 1. Overview of the proposed GAIF framework

3.1. Data Collection

This study uses the Mobile Device Usage and User Behavior Dataset, a publicly available, preprocessed derivative of the Drebin Android corpus. The original Drebin dataset is not used directly. Instead, the curated dataset reorganizes application-level indicators into user-level behavioral profiles, enabling the analysis of mobile usage behavior rather than malware classification.

In this dataset, a user represents an abstract behavioral profile constructed by aggregating multiple Android applications and their associated attributes. The resulting user profiles are synthetic behavioral abstractions derived from application-level data and are not balanced across behavioral categories, reflecting the heterogeneous nature of the source corpus. Consequently, anomaly detection is performed at the user level, consistent with the evaluation and discussion presented in Section 4.

Each user profile is described using a fixed-dimensional set of behavioral attributes derived from application-level indicators. These attributes capture permission access frequency, proxies for application usage intensity, and resource-related capabilities such as network access, background execution, and storage usage. Conceptually, these features characterize behavioral tendencies rather than malicious intent, allowing the identification of users whose interaction patterns deviate structurally or contextually from the normative population.

3.1.1. Dataset Limitations

The dataset represents static behavioral snapshots and does not capture temporal evolution or real-time user interactions. Furthermore, as a proxy dataset derived from application indicators, it abstracts user behavior rather than recording direct human interaction logs. These limitations are explicitly acknowledged in the interpretation of experimental results in Section 4 and motivate future validation using longitudinal and real-world mobile usage data.

3.2. Data Preprocessing

Aggregation and initial preprocessing steps, including feature consolidation, normalization, and the removal of application identifiers, were performed by the dataset creators prior

to this study. The resulting dataset consists of 700 user-level behavioral profiles, each treated as a static snapshot of mobile usage behavior.

In this work, these profiles are used exclusively to construct user–feature graphs and to evaluate unsupervised anomaly detection performance, as reported in Section 4. Importantly, this study does not perform malware classification, application-level detection, or packet-level intrusion analysis. The dataset is used strictly as a proxy representation of mobile user behavior, enabling controlled evaluation of user-level behavioral anomalies through post-hoc performance metrics and embedding-based interpretability analyses.

3.3. Feature Relevance Analysis

To reduce dimensionality and suppress statistically insignificant predictors, a Chi-square (χ^2) relevance analysis is performed exclusively on the training data. The Chi-square test evaluates the statistical dependency between each behavioral attribute and the available class labels and is used solely as an offline feature filtering mechanism, not as a learning signal.

Although GAIF performs anomaly detection in an unsupervised manner, this limited use of labels is restricted to identifying behaviorally informative attributes prior to graph construction. Attributes with negligible χ^2 scores are discarded to improve computational efficiency and model interpretability without influencing the unsupervised detection process. The Chi-square statistic is defined as:

$$\chi^2 = \sum_{i=1}^{K} \frac{(O_i - E_i)^2}{E_i} \tag{1}$$

where O_i denotes the observed frequency of a feature within a given class, E_i represents the expected frequency under the assumption of independence, and K is the number of feature–class combinations.

Higher χ^2values indicate stronger statistical association between a feature and user behavior, implying greater relevance. The resulting scores are used to rank behavioral attributes, retaining only the top-ranked features for subsequent graph construction. Attributes such as gender, age, and device model exhibit low χ^2 values, confirming their limited influence on behavioral variation and anomaly likelihood.

Class labels are never used during graph construction, embedding learning, anomaly scoring, or threshold estimation. Their role is confined strictly to offline statistical filtering and post-hoc evaluation. While the use of Chi-square introduces a form of weak supervision, potential bias is mitigated by restricting feature selection to the training partition and ensuring that labels are not exposed to the detection model. As a result, the anomaly detection pipeline remains fully unsupervised during both training and inference.

Alternative unsupervised filtering methods, such as variance thresholding or Laplacian score, could be applied in this stage. However, Chi-square is selected for its simplicity, interpretability, and effectiveness in suppressing noise in sparse mobile behavioral data. Graph-based variable selection methods also provide principled alternatives. For example, [55] proposed a causal graph-based feature selection approach for BERT-based survival prediction from clinical discharge data, demonstrating how relational structure can guide the identification of contextually relevant predictors—principles that complement GAIF's offline statistical filtering and subsequent graph embedding pipeline

3.4. Data Split and Leakage Control

To ensure unbiased evaluation and prevent information leakage, the optimized user behavior dataset is partitioned into training and testing subsets using an 80:20 stratified split, following standard machine learning practice [9], [13]. The training set is used exclusively for all learning-dependent operations, including feature relevance analysis, graph construction, and embedding learning.

To preserve a strict separation between training and evaluation, test instances are embedded inductively using parameters learned from the training graph. No test data are involved in feature selection, graph topology construction, or embedding optimization. This protocol ensures that evaluation results reflect genuine model generalization rather than memorization or contamination.

3.5. Graph Construction and Representation Learning

GAIF models mobile user behavior using a user–feature bipartite graph $G = (V, E)$ where one set of nodes represents users and the other represents behavioral attributes. Edges encode observed user–feature interactions, with weights proportional to normalized feature values. Bipartite modeling explicitly preserves relational dependencies between users and behavioral attributes, enabling the detection of anomalies arising from atypical relational patterns rather than deviations in individual features alone [15], [16]. An illustration of the constructed bipartite graph is shown in Figure 2, where nodes correspond to users and features, and edges represent interaction strength.

To generate low-dimensional representations suitable for anomaly detection, two complementary graph representation learning methods are applied. Node2Vec [20] captures both local and global structural relationships through biased random walks, while GraphSAGE [17] performs inductive neighborhood aggregation, enabling generalization to unseen users. Together, these methods produce embeddings that preserve structural topology and local semantic context [17], [18].

The embeddings learned by Node2Vec and GraphSAGE are fused through vector concatenation, forming a unified representation for each user. Let

$$Z = [z_1, z_2, \dots, z_n] \in \mathbb{R}^{n \times d} \tag{2}$$

denote the resulting embedding matrix, where n is the number of users and d is the embedding dimension. This fused representation combines global structural information with local neighborhood context and serves as input to the Isolation Forest for anomaly scoring.

For GraphSAGE, node embeddings are computed using the mean-aggregation formulation:

$$h_i^{(k)} = \sigma\left(W^{(k)}.\,\text{AGGREGATE}\left(\left\{h_j^{(k-1)} | j \in \mathcal{N}(i)\right\}\right) + b^{(k)}\right) \tag{3}$$

where $h_i^{(k)}$ denotes the representation of node i at at layer k, $\mathcal{N}(i)$ is the neighborhood of node i, $W^{(k)}$ and $b^{(k)}$ are learnable parameters, and σ is a nonlinear activation function (ReLU in this study).

The integration of Node2Vec, GraphSAGE, and Isolation Forest allows GAIF to exploit their complementary strengths. Node2Vec captures global structural irregularities, GraphSAGE preserves local contextual dependencies and supports inductive inference, and Isolation Forest efficiently isolates sparse and atypical regions in the embedding space. Individually, these methods are insufficient: vector-based Isolation Forest ignores relational structure, Node2Vec alone lacks fine-grained neighborhood context, and GraphSAGE alone does not encode global topology. Their integration enables effective anomaly isolation while maintaining computational efficiency.

Table 1. Hyperparameter settings.

Component	Parameter	Value
Node2Vec	Embedding dimension	64
	Walk length	10
	Walks per node	50
	Return parameter (p)	1.0
	In–out parameter (q)	1.0
GraphSAGE	Number of layers	2
	Hidden dimension	64
	Aggregation function	Mean
	Activation function	ReLU
Isolation Forest	Number of trees	200
	Contamination rate	0.07
	Max samples	Auto
	Random state	42

In the GAIF framework, the contamination threshold of the Isolation Forest is set to 0.07, selected empirically from the training data using silhouette analysis to balance false positives and detection sensitivity. This value deviates from the default setting (0.1) to better reflect dataset-specific sparsity. All hyperparameters are held constant across experiments to ensure fair and reproducible comparisons. The complete configuration is summarized in Table 1.

3.6. Isolation Forest–Based Anomaly Detection

Following the graph construction and representation learning stage described in Section 3.5, the user–feature bipartite graph illustrated in Figure 2 serves as the structural foundation for the final anomaly detection process. Graph embeddings learned from this structure encode both global connectivity patterns and local neighborhood context, which are subsequently leveraged by the Isolation Forest for unsupervised anomaly scoring.

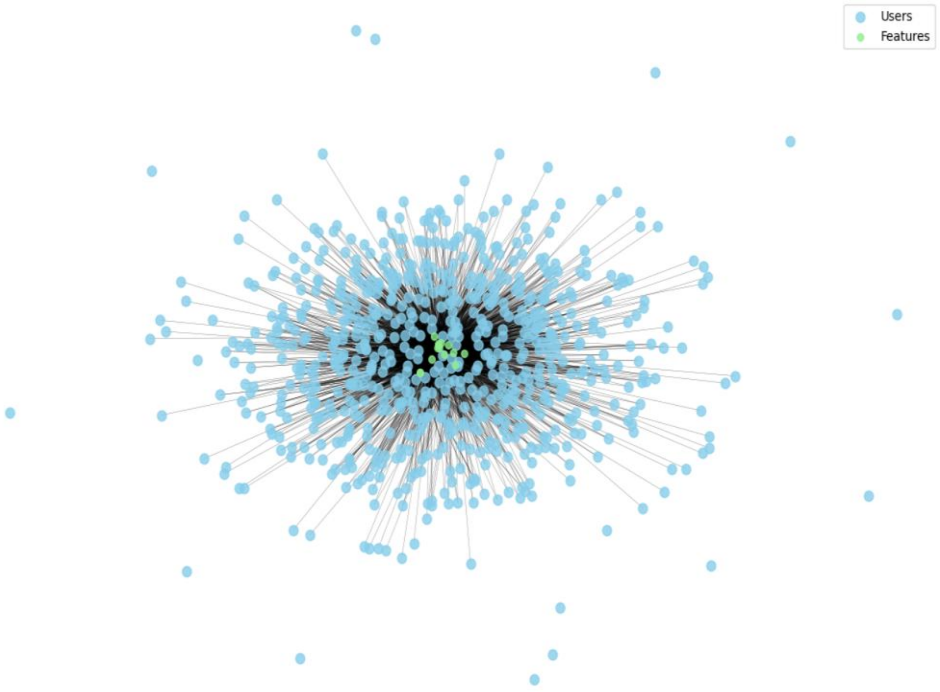

Figure 2. User–Feature Bipartite Graph used for GAIF

Algorithm 1. Isolation Forest–Based Anomaly Detection in GAIF

INPUT: $Z \in \mathbb{R}^{n \times d}$: graph-embedded user representations, t: number of isolation trees; ψ: subsampling size per tree; τ: anomaly score threshold
OUTPUT: A: binary anomaly labels (0 = normal, 1 = anomalous); S: anomaly scores for all user instances

1: Initialize the Isolation Forest with t trees and subsampling size ψ.
2: For each isolation tree T_i, randomly sample a subset $Z_\psi \subset Z$
3: Recursively partition Z_ψ by randomly selecting a feature dimension and split value until instances are isolated.
4: Compute the expected path length $E\left(h(z_j)\right)$ for each instance (z_j).
5: Calculate anomaly scores: $S(z_j)$ according to Eq. (4).
6: Assign anomaly labels based on threshold τ.
7: Return anomaly scores S and labels A for all users.

Specifically, the fused embedding matrix obtained from the concatenation of Node2Vec and GraphSAGE representations (defined in Eq. (2)) is used as the sole input to the Isolation Forest. No raw behavioral features or class labels are involved at this stage, ensuring that

anomaly detection operates purely on graph-informed representations and preserves the unsupervised nature of the GAIF framework.

The Isolation Forest identifies anomalous users by recursively partitioning the embedding space and isolating instances that require fewer splits to separate from the majority of the data. Users whose embeddings occupy sparse or structurally isolated regions—often arising from atypical relational patterns encoded in the bipartite graph shown in Figure 2—are assigned higher anomaly scores [9], [44], [56]. The complete anomaly detection procedure is formally described in Algorithm 1.

Let z_j denote the fused embedding of user j. As detailed in Algorithm 1, the anomaly score is computed based on the expected path length $E\left(h(z_j)\right)$ across an ensemble of isolation trees:

$$S(z_j) = 2^{-\frac{E\left(h(z_j)\right)}{c(\psi)}} \tag{4}$$

where $c(\psi) = 2H(\psi - 1) - \frac{2(\psi-1)}{\psi}$ is the normalization constant and $H(\cdot)$ denotes the harmonic number. Higher anomaly scores indicate instances that are more easily isolated in the embedding space.

To obtain binary anomaly labels, a contamination threshold of 0.07 is applied, classifying users with anomaly scores exceeding this threshold as anomalous. This threshold is selected empirically using silhouette analysis on the training data to balance detection sensitivity and false positive rates and differs from the default Isolation Forest setting to reflect dataset-specific sparsity. Threshold selection is performed exclusively on the training partition, and the same threshold is applied unchanged during testing to ensure unbiased evaluation.

By integrating graph-derived embeddings with isolation-based anomaly scoring, as formalized in Algorithm 1 and Eq. (4), the GAIF framework enables the detection of user-level behavioral anomalies that arise from atypical relational structures rather than isolated feature deviations. This stage completes the GAIF pipeline by transforming relational patterns encoded in the bipartite graph (Figure 2) into interpretable anomaly scores suitable for downstream analysis.

4. Results and Discussion

This section presents the experimental evaluation of the proposed GAIF framework on the curated Drebin-derived mobile user behavior dataset. All experiments follow the methodological protocol described in Section 3, including strict separation between training and testing data, inductive embedding of test instances, and fully unsupervised anomaly scoring. The evaluation focuses on overall detection effectiveness, computational efficiency, and the contribution of individual pipeline components.

4.1. Experimental Setup and Evaluation Overview

All experiments were conducted in a controlled CPU-based environment to ensure reproducibility and fair runtime comparison. The implementation was developed in Python (version 3.10) and executed on a system equipped with an Intel Xeon processor operating at 2.3 GHz and 16 GB of RAM, running Windows 10.

Graph representation learning was implemented using PyTorch and PyTorch Geometric, while anomaly detection and evaluation metrics were computed using scikit-learn. Data handling was performed with pandas, and UMAP was used exclusively for post-hoc visualization and interpretability analysis.

The dataset was split using the 80:20 stratified protocol described in Section 3, yielding 560 training instances and 140 test instances. Runtime was measured as the total elapsed time covering graph construction, embedding generation, and Isolation Forest–based anomaly scoring, excluding offline preprocessing steps. All experiments were executed in a single-threaded CPU setting and repeated five times, with mean runtime reported to reduce variability. As GAIF operates in an unsupervised manner, anomaly scores were computed without using class labels and converted into binary predictions using a contamination threshold of 0.07, selected from the training data as described in Section 3.6. Performance was assessed

using precision, recall, F1-score, and area under the ROC curve (AUC), enabling direct comparison with baseline unsupervised anomaly detection methods.

4.2. Quantitative Results

Table 2 summarizes the quantitative performance of GAIF under the evaluation setting described above. The results indicate that GAIF achieves strong detection effectiveness, characterized by high precision and recall, resulting in an F1-score of 0.94 and an AUC of 0.97. These findings suggest that incorporating graph-based representations enhances the ability of isolation-based detection to distinguish anomalous user behavior profiles within the evaluated dataset.

The reported runtime reflects the combined cost of embedding generation and anomaly scoring, demonstrating that the performance gains achieved by GAIF do not require prohibitive computational overhead. This balance between detection effectiveness and efficiency supports the practical feasibility of the proposed framework for offline and near-real-time analytical scenarios, rather than deployed real-time systems.

Table 2. Quantitative performance of GAIF.

Metric	Value
Precision	0.96
Recall	0.94
F1-score	0.94
AUC	0.97
Runtime (s)	33.8

4.3. Ablation Studies

To assess the contribution of individual components within the GAIF pipeline, an ablation study was conducted by progressively enabling graph-based modeling and embedding mechanisms. The evaluated configurations, summarized in Table 3, range from a vector-based baseline to the full GAIF framework. This staged design allows the isolated impact of feature filtering, graph embedding, and embedding fusion to be examined.

The baseline configuration (C_1) applies Isolation Forest directly to the original feature space. Configuration C_2 introduces offline Chi-square feature filtering prior to anomaly detection. Configuration C_3 incorporates Node2Vec-based graph embeddings to encode global relational structure. The final configuration (C_4) corresponds to the complete GAIF framework, combining Node2Vec and GraphSAGE embeddings with Isolation Forest. UMAP is used solely for visualization and does not constitute a model configuration.

Table 3. Model configurations for ablation analysis.

Case	Description	Components
C_1	Baseline Isolation Forest	Isolation Forest
C_2	Feature-filtered IF	Chi-square filtering + IF
C_3	Graph-embedded IF	Node2Vec + IF
C_4	GAIF (full model)	Node2Vec + GraphSAGE + IF

Table 4 reports the comparative performance of these configurations. The baseline Isolation Forest (C_1) provides a reference point for vector-only anomaly detection. Introducing Chi-square feature filtering (C_2) yields moderate improvements, indicating the benefit of suppressing weak or noisy attributes. A more substantial performance gain is observed with Node2Vec-based embeddings (C_3), highlighting the importance of capturing global relational structure. The full GAIF framework (C_4) achieves the strongest overall performance, demonstrating that combining global structural encoding with local neighborhood aggregation produces a more discriminative embedding space for isolation-based anomaly detection. Notably, these gains are achieved with only a marginal increase in runtime, underscoring the practicality of the proposed framework within the evaluated offline analytical setting.

Table 4. Performance comparison of ablation configurations.

Model	Precision	Recall	F1-score	AUC	Runtime (s)
C_1	0.89	0.85	0.86	0.90	28.4
C_2	0.91	0.87	0.88	0.92	27.6
C_3	0.94	0.92	0.93	0.95	33.2
C_4	0.96	0.94	0.94	0.97	33.8

4.4. Interpretability Analysis

Figure 3 presents a side-by-side comparison of two-dimensional UMAP visualizations across the GAIF ablation stages, illustrating how graph-based representation learning progressively improves the separability of normal and anomalous user profiles. The figure compares embeddings derived from raw features, GraphSAGE, Node2Vec, and the fused GAIF representation, using a consistent color scheme to support visual interpretability.

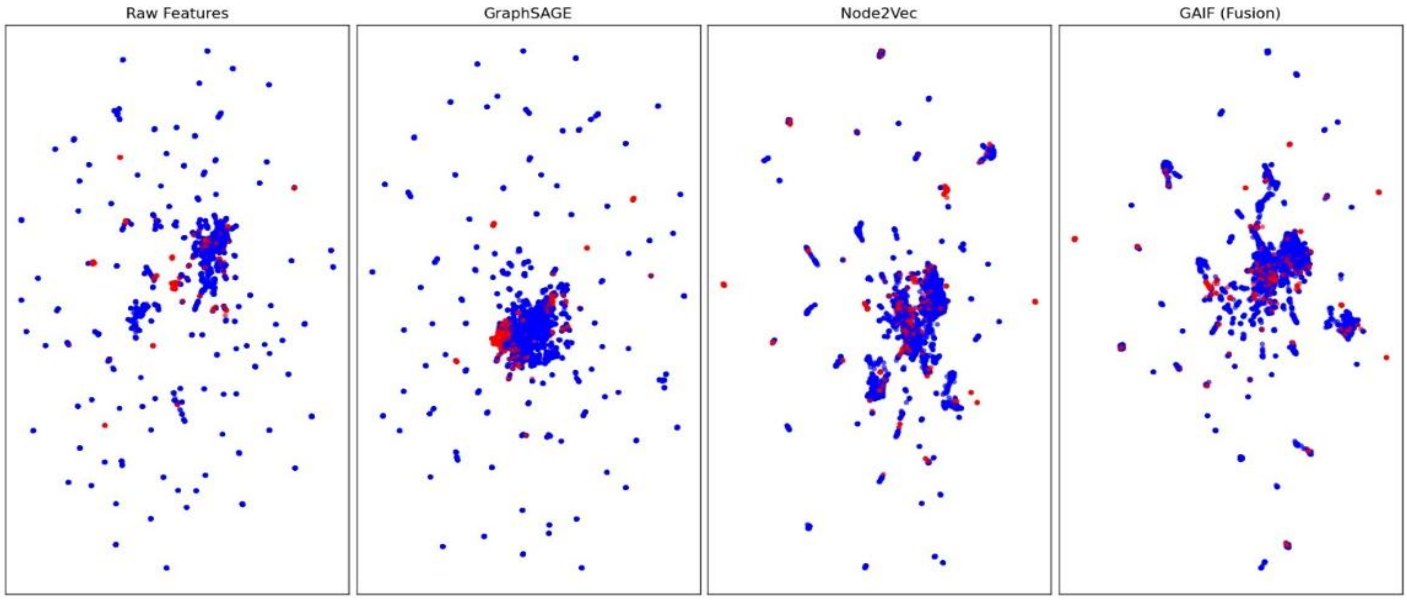

Figure 3. UMAP projections of user embeddings across ablation stages, including raw features, GraphSAGE, Node2Vec, and GAIF fusion. Normal user profiles are shown in blue (RGB: 31, 119, 180), and anomalous user profiles are shown in red (RGB: 214, 39, 40), with consistent coloring across all subplots.

In the raw feature space, normal and anomalous users exhibit substantial overlap, indicating that independent feature vectors provide limited structural discrimination for isolation-based anomaly detection. Without explicit relational modeling, the resulting embedding geometry is diffuse, and anomalous users are not clearly separated from the majority population.

The GraphSAGE embedding introduces improved local organization by aggregating neighborhood information from the user–feature bipartite graph. This leads to a denser central cluster of normal users; however, noticeable overlap with anomalous users remains, suggesting that local neighborhood aggregation alone is insufficient to capture broader relational irregularities.

The Node2Vec embedding further enhances separability by encoding global structural relationships through biased random walks on the bipartite graph. In this representation, anomalous users are more frequently displaced toward peripheral regions of the embedding space, reflecting the role of global topological context in highlighting atypical relational patterns.

The GAIF fusion embedding exhibits the most pronounced separation, characterized by a compact manifold of normal users and more distinctly isolated anomalous points. This outcome arises from the complementary integration of Node2Vec's global topology preservation and GraphSAGE's local neighborhood aggregation, yielding an embedding space that more effectively amplifies structural deviations relevant to anomaly detection.

From an isolation-based perspective, the geometry of these embeddings directly influences Isolation Forest behavior. Users embedded in sparse or weakly connected regions are isolated with fewer random splits, resulting in shorter average path lengths, whereas users embedded within dense clusters require deeper partitioning. This interaction between graph topology, embedding geometry, and isolation dynamics explains the quantitative performance improvements reported in Table 4, without modifying the underlying anomaly detection algorithm. Overall, the visual patterns observed in Figure 3 demonstrate that GAIF improves anomaly detection performance primarily through representation enhancement driven by graph structure, rather than through changes to the isolation mechanism itself. This reinforces the interpretation of GAIF as a graph-aware representation pipeline for unsupervised mobile user behavior anomaly detection, consistent with the methodological scope of this study.

4.5. Comparative Performance Evaluation

This section compares the performance of the proposed GAIF with representative unsupervised anomaly detection baselines that were reimplemented and evaluated under the same dataset, preprocessing pipeline, and evaluation protocol described in Sections 3 and 4. The comparative results are summarized in Table 5, which reports precision, recall, F1-score, AUC, and runtime for each method. To ensure experimental fairness and reproducibility, all baseline methods—including IF, LOF, OC-SVM, and Autoencoder-based anomaly detection—were evaluated on the same Mobile Device Usage and User Behavior dataset, using identical train–test splits, preprocessing steps, and evaluation metrics. Prior studies are referenced only for methodological context in the related work section and are not used for direct numerical comparison.

Table 5. Comparative performance of GAIF and reimplemented unsupervised anomaly detection baselines evaluated under the same dataset and experimental protocol.

Model	Precision	Recall	F1-Score	AUC	Runtime (s)	Observation
IF	0.89	0.85	0.86	0.90	28.4	Efficient, but ignores relational structure
LOF	0.88	0.84	0.85	0.89	31.6	Sensitive to neighborhood size
OC-SVM	0.90	0.86	0.88	0.91	42.3	Kernel-sensitive, higher cost
Autoencoder	0.91	0.88	0.89	0.92	39.8	Captures nonlinearity, limited interpretability
GAIF (Ours)	0.96	0.94	0.94	0.97	33.8	Best accuracy with efficient runtime

As shown in Table 5, GAIF achieves the strongest overall performance across all evaluation metrics. In particular, GAIF attains the highest F1-score (0.94) and AUC (0.97), indicating improved detection effectiveness and discrimination capability compared with the baseline methods evaluated under the same conditions. Despite incorporating graph-based representation learning, GAIF maintains competitive computational efficiency, with runtime comparable to or lower than several vector-based baselines.

Among the reimplemented baselines, the standard Isolation Forest exhibits strong efficiency but lower detection accuracy due to its inability to model relational dependencies among behavioral attributes. Local Outlier Factor and One-Class SVM achieve moderate performance but are sensitive to neighborhood size and kernel selection, respectively, which can limit robustness in heterogeneous behavioral spaces. Autoencoder-based anomaly detection captures nonlinear feature interactions but incurs higher computational cost and offers limited interpretability.

By contrast, GAIF benefits from integrating graph-based representation learning with isolation-based anomaly scoring. Modeling user behavior as a user–feature bipartite graph enables the framework to encode relational dependencies that are not accessible to vector-based methods. The resulting graph-informed embeddings reshape the data geometry in a manner that enhances the effectiveness of Isolation Forest without increasing algorithmic complexity. Overall, this comparative evaluation demonstrates that GAIF provides a favorable balance between detection effectiveness and computational efficiency within the evaluated

offline analytical setting, outperforming conventional unsupervised anomaly detection baselines when assessed under identical experimental conditions..

5. Conclusions

This study proposed GAIF, an unsupervised framework for mobile user behavior anomaly detection that integrates graph-based representation learning with isolation-based anomaly scoring. GAIF models user behavior through a user–feature bipartite graph and combines Node2Vec and GraphSAGE embeddings to capture both global structural topology and local contextual relationships, which are subsequently exploited by an Isolation Forest for anomaly detection. Experimental results on the Mobile Device Usage and User Behavior Dataset, a curated proxy dataset derived from the Drebin Android corpus, demonstrate that GAIF achieves improved detection effectiveness compared with representative vector-based and graph-only unsupervised baselines under the evaluated setting. Quantitative gains in F1-score and AUC are complemented by embedding-level interpretability, where UMAP visualizations reveal progressive structural separation between normal and anomalous user profiles. These findings support the central hypothesis that enriching isolation-based anomaly detection with graph-informed representations enhances the identification of user-level behavioral anomalies driven by relational and contextual deviations.

From a methodological perspective, this work contributes a reproducible and computationally efficient pipeline that bridges the gap between vector-based anomaly detectors, which lack relational awareness, and graph-only approaches, which often incur high computational cost. By demonstrating that performance improvements arise primarily from representation enhancement rather than algorithmic modification, GAIF clarifies the role of graph-based embeddings as a complementary mechanism for isolation-based detection rather than as a standalone detection model.

Several limitations of this study should be acknowledged. The evaluated dataset represents static behavioral snapshots rather than continuous user activity, aggregates multiple applications into synthetic user profiles, and does not capture real-time interactions. As a result, the findings should not be interpreted as evidence of real-time deployment capability. Future work should focus on validating the proposed framework on longitudinal and real-world mobile usage data, extending the model to temporal and streaming settings, and exploring alternative unsupervised feature selection and graph construction strategies to further improve robustness and generalizability. Overall, GAIF provides a principled and interpretable approach for unsupervised mobile user behavior anomaly detection, contributing to ongoing research on graph-aware behavioral analytics and offering a foundation for future extensions in real-world mobile security and monitoring applications..

Author Contributions: Conceptualization: F.A.E. and A.P.B.; Methodology: A.A.O., S.I.O., and T.C.A.; Software: S.I.O. and A.A.O.; Validation: A.P.B., F.A.E., and A.A.O.; Formal Analysis: S.I.O. and A.A.O.; Investigation: A.A.O., F.A.E., N.C.A., and A.P.B.; Data Curation: T.C.A. and A.P.B.; Writing—original draft: A.P.B. and S.I.O.; Writing—review and editing: A.P.B., N.C.A., and A.A.O.; Visualization: T.C.A. and F.A.E.; N.C.A; Supervision: A.A.O. and F.A.E.; Project administration: S.I.O. and T.C.A.; Funding acquisition: All. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is available online [web]: https://github.com/Gayatri-Subramanium/Mobile-Device-Usage-and-User-Behavior-Dataset.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] M. Ashawa and S. Morris, "Analysis of Mobile Malware: A Systematic Review of Evolution and Infection Strategies," *J. Inf. Secur. Cybercrimes Res.*, vol. 4, no. 2, pp. 103–131, Dec. 2021, doi: 10.26735/KRVI8434.

[2] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, Jan. 2016, doi: 10.1016/j.jnca.2015.11.016.

[3] D. Samariya and A. Thakkar, "A Comprehensive Survey of Anomaly Detection Algorithms," *Ann. Data Sci.*, vol. 10, pp. 829–850, Nov. 2023, doi: 10.1007/s40745-021-00362-9.

[4] K. G. Mehrotra, C. K. Mohan, and H. Huang, "Anomaly Detection," in *Anomaly Detection Principles and Algorithms*, 2017, pp. 21–32. doi: 10.1007/978-3-319-67526-8_2.
[5] R. A. Manzano Sanchez, K. Naik, A. Albasir, M. Zaman, and N. Goel, "Detection of Anomalous Behavior of Smartphone Devices using Changepoint Analysis and Machine Learning Techniques," *Digit. Threat. Res. Pract.*, vol. 4, no. 1, pp. 1–28, Mar. 2023, doi: 10.1145/3492327.
[6] A. Blaise, M. Bouet, V. Conan, and S. Secci, "Group anomaly detection in mobile app usages: A spatiotemporal convex hull methodology," *Comput. Networks*, vol. 216, p. 109277, Oct. 2022, doi: 10.1016/j.comnet.2022.109277.
[7] A. Pathak, U. Barman, and T. S. Kumar, "Machine learning approach to detect android malware using feature-selection based on feature importance score," *J. Eng. Res.*, vol. 13, no. 2, pp. 712–720, Jun. 2025, doi: 10.1016/j.jer.2024.04.008.
[8] A. Nematzadeh, S. C. Meylan, and T. L. Griffiths, "Evaluating Vector-Space Models of Word Representation, or,The Unreasonable Effectiveness of Counting Words Near Other Words," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2017. [Online]. Available: https://escholarship.org/uc/item/1kh9p4gj
[9] W. S. Al Farizi, I. Hidayah, and M. N. Rizal, "Isolation Forest Based Anomaly Detection: A Systematic Literature Review," in *2021 8th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, Sep. 2021, pp. 118–122. doi: 10.1109/ICITACEE53184.2021.9617498.
[10] M. Yamauchi, Y. Ohsita, M. Murata, K. Ueda, and Y. Kato, "Anomaly Detection in Smart Home Operation From User Behaviors and Home Conditions," *IEEE Trans. Consum. Electron.*, vol. 66, no. 2, pp. 183–192, May 2020, doi: 10.1109/TCE.2020.2981636.
[11] M. Zhang, B. Xu, and J. Gong, "An Anomaly Detection Model Based on One-Class SVM to Detect Network Intrusions," in *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, Dec. 2015, pp. 102–107. doi: 10.1109/MSN.2015.40.
[12] O. Alghushairy, R. Alsini, T. Soule, and X. Ma, "A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams," *Big Data Cogn. Comput.*, vol. 5, no. 1, p. 1, Dec. 2020, doi: 10.3390/bdcc5010001.
[13] M. S. Parwez, D. B. Rawat, and M. Garuba, "Big Data Analytics for User-Activity Analysis and User-Anomaly Detection in Mobile Wireless Network," *IEEE Trans. Ind. Informatics*, vol. 13, no. 4, pp. 2058–2065, Aug. 2017, doi: 10.1109/TII.2017.2650206.
[14] R. Meddeb, F. Jemili, B. Triki, and O. Korbaa, "Anomaly-based Behavioral Detection in Mobile Ad-Hoc Networks," *Procedia Comput. Sci.*, vol. 159, pp. 77–86, 2019, doi: 10.1016/j.procs.2019.09.162.
[15] H. Li, C. Zhao, Y. Liu, and X. Zhang, "Anomaly detection by discovering bipartite structure on complex networks," *Comput. Networks*, vol. 190, p. 107899, May 2021, doi: 10.1016/j.comnet.2021.107899.
[16] Z. Chen and A. Sun, "Anomaly Detection on Dynamic Bipartite Graph with Burstiness," in *2020 IEEE International Conference on Data Mining (ICDM)*, Nov. 2020, pp. 966–971. doi: 10.1109/ICDM50108.2020.00110.
[17] X. Wang, H. Dou, D. Dong, and Z. Meng, "Graph anomaly detection based on hybrid node representation learning," *Neural Networks*, vol. 185, p. 107169, May 2025, doi: 10.1016/j.neunet.2025.107169.
[18] X. Ma *et al.*, "A Comprehensive Survey on Graph Anomaly Detection With Deep Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12012–12038, Dec. 2023, doi: 10.1109/TKDE.2021.3118815.
[19] A. A. Ojugo *et al.*, "CoSoGMIR: A Social Graph Contagion Diffusion Framework using the Movement-Interaction-Return Technique," *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 163–173, Dec. 2023, doi: 10.33633/jcta.v1i2.9355.
[20] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 855–864. doi: 10.1145/2939672.2939754.
[21] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, Sep. 2017, vol. 2017-Decem, no. Nips, pp. 1025–1035. [Online]. Available: http://arxiv.org/abs/1706.02216
[22] H. Zhang, Y. Luo, Q. Yu, L. Sun, X. Li, and Z. Sun, "A Framework of Abnormal Behavior Detection and Classification Based on Big Trajectory Data for Mobile Networks," *Secur. Commun. Networks*, vol. 2020, pp. 1–15, Dec. 2020, doi: 10.1155/2020/8858444.
[23] A. P. Ferreira, C. Gupta, P. R. M. Inacio, and M. M. Freire, "Behaviour-based Malware Detection in Mobile Android Platforms Using Machine Learning Algorithms," *J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl.*, vol. 12, no. 4, pp. 62–88, 2021, doi: 10.22667/JOWUA.2021.12.31.062.
[24] A. P. Binitie *et al.*, "MoBiSafe: an obfuscated single factor authentication mode to enhance secured USSD channel transaction in Nigeria," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 40, no. 1, p. 426, Oct. 2025, doi: 10.11591/ijeecs.v40.i1.pp426-436.
[25] B. A. P. and B. J. O., "Adapting User Interface Design to Mitigate Shoulder Surfing Attacks in USSD Channel," *African J. Environ. Nat. Sci. Res.*, vol. 7, no. 1, pp. 13–27, Jan. 2024, doi: 10.52589/AJENSR-DPCGWN0X.
[26] E. V. Ugbotu *et al.*, "Investigating a SMOTE-Tomek Boosted Stacked Learning Scheme for Phishing Website Detection: A Pilot Study," *J. Comput. Theor. Appl.*, vol. 3, no. 2, pp. 145–159, Oct. 2025, doi: 10.62411/jcta.14472.
[27] J. Burgueño, I. De-la-Bandera, J. Mendoza, D. Palacios, C. Morillas, and R. Barco, "Online Anomaly Detection System for Mobile Networks," *Sensors*, vol. 20, no. 24, p. 7232, Dec. 2020, doi: 10.3390/s20247232.
[28] G. Suarez-Tangil, S. K. Dash, M. Ahmadi, J. Kinder, G. Giacinto, and L. Cavallaro, "DroidSieve," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, Mar. 2017, pp. 309–320. doi: 10.1145/3029806.3029825.
[29] D. R. I. M. Setiadi, A. R. Muslikh, S. W. Irianananda, W. Warto, J. Gondohanindijo, and A. A. Ojugo, "Outlier Detection Using Gaussian Mixture Model Clustering to Optimize XGBoost for Credit Approval Prediction," *J. Comput. Theor. Appl.*, vol. 2, no. 2, pp. 244–255, Nov. 2024, doi: 10.62411/jcta.11638.
[30] A. Adesh, S. G, J. Shetty, and L. Xu, "Local outlier factor for anomaly detection in HPCC systems," *J. Parallel Distrib. Comput.*, vol. 192, p. 104923, Oct. 2024, doi: 10.1016/j.jpdc.2024.104923.
[31] P. Bountzis, D. Kavallieros, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "A deep one-class classifier for network anomaly detection using autoencoders and one-class support vector machines," *Front. Comput. Sci.*, vol. 7, Oct. 2025, doi: 10.3389/fcomp.2025.1646679.
[32] J. Liu, H. Wang, H. Hang, S. Ma, X. Shen, and Y. Shi, "Self-Supervised Random Forest on Transformed Distribution for Anomaly Detection," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 36, no. 2, pp. 2675–2689, Feb. 2025, doi: 10.1109/TNNLS.2023.3348833.

[33] T. Lu, Y. Du, L. Ouyang, Q. Chen, and X. Wang, "Android Malware Detection Based on a Hybrid Deep Learning Model," *Secur. Commun. Networks*, vol. 2020, pp. 1–11, Aug. 2020, doi: 10.1155/2020/8863617.
[34] W. Ju *et al.*, "A Comprehensive Survey on Deep Graph Representation Learning," *Neural Networks*, vol. 173, p. 106207, May 2024, doi: 10.1016/j.neunet.2024.106207.
[35] G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, "Deep Learning for Anomaly Detection," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2022, doi: 10.1145/3439950.
[36] A. Modell, J. Larson, M. Turcotte, and A. Bertiger, "A Graph Embedding Approach to User Behavior Anomaly Detection," in *2021 IEEE International Conference on Big Data (Big Data)*, Dec. 2021, pp. 2650–2655. doi: 10.1109/BigData52589.2021.9671423.
[37] L. Akoglu, H. Tong, and D. Koutra, "Graph-based Anomaly Detection and Description: A Survey," *arXiv*. Apr. 28, 2014. [Online]. Available: http://arxiv.org/abs/1404.4679
[38] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep Graph Infomax," *ArXiv*. Dec. 21, 2018. [Online]. Available: http://arxiv.org/abs/1809.10341
[39] H. Kim, B. S. Lee, W.-Y. Shin, and S. Lim, "Graph Anomaly Detection With Graph Neural Networks: Current Status and Challenges," *IEEE Access*, vol. 10, pp. 111820–111829, 2022, doi: 10.1109/ACCESS.2022.3211306.
[40] G. Li and J. J. Jung, "Dynamic graph embedding for outlier detection on multiple meteorological time series," *PLoS One*, vol. 16, no. 2, p. e0247119, Feb. 2021, doi: 10.1371/journal.pone.0247119.
[41] Bandyopadhyay Sambaran, Vishal Vivek Saley, and Murty M.N., "Integrating Network Embedding and Community Outlier Detection via Multiclass Graph Description," in *Frontiers in Artificial Intelligence and Applications*, 2020. doi: 10.3233/FAIA200191.
[42] K. Hoarau, P. U. Tournoux, and T. Razafindralambo, "Unsupervised representation learning for BGP anomaly detection using graph auto-encoders," *ITU J. Futur. Evol. Technol.*, vol. 5, no. 1, pp. 120–133, Mar. 2024, doi: 10.52953/CTFY7896.
[43] Purushottam Perapu, "Anomaly Detection in User Behaviour Using Machine Learning For Cloud Platforms," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 11, no. 3, pp. 805–809, May 2025, doi: 10.32628/CSEIT25113343.
[44] Y. Chabchoub, M. U. Togbe, A. Boly, and R. Chiky, "An In-Depth Study and Improvement of Isolation Forest," *IEEE Access*, vol. 10, pp. 10219–10237, 2022, doi: 10.1109/ACCESS.2022.3144425.
[45] X. Kong, J. Wang, Z. Hu, Y. He, X. Zhao, and G. Shen, "Mobile Trajectory Anomaly Detection: Taxonomy, Methodology, Challenges, and Directions," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19210–19231, Jun. 2024, doi: 10.1109/JIOT.2024.3376457.
[46] N. R. Palakurti, "Challenges and Future Directions in Anomaly Detection," in *Practical Applications of Data Processing, Algorithms, and Modeling*, 2024, pp. 269–284. doi: 10.4018/979-8-3693-2909-2.ch020.
[47] M. Yang and J. Zhang, "Data Anomaly Detection in the Internet of Things: A Review of Current Trends and Research Challenges," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 9, 2023, doi: 10.14569/IJACSA.2023.0140901.
[48] M. Bahri, F. Salutari, A. Putina, and M. Sozio, "AutoML: state of the art with a focus on anomaly detection, challenges, and research directions," *Int. J. Data Sci. Anal.*, vol. 14, no. 2, pp. 113–126, Aug. 2022, doi: 10.1007/s41060-022-00309-0.
[49] D. D. Yao, X. Shu, L. Cheng, and S. J. Stolfo, *Anomaly Detection as a Service*. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-031-02354-5.
[50] E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, "Anomal-E: A self-supervised network intrusion detection system based on graph neural networks," *Knowledge-Based Syst.*, vol. 258, p. 110030, Dec. 2022, doi: 10.1016/j.knosys.2022.110030.
[51] A. Zoubir and B. Missaoui, "Integrating Graph Neural Networks with Scattering Transform for Anomaly Detection," *ArXiv*. Apr. 24, 2024. [Online]. Available: http://arxiv.org/abs/2404.10800
[52] T. O. Jejeniwa, T. O. Jejeniwa, and O. S. Owolabi, "Leveraging Hybrid AI for Real-Time Fraud Detection: A Case Study On The Efficacy of Graph Neural Networks and Anomaly Detection In Nigerian Fintechs," *J. Digit. Secur. Forensics*, vol. 2, no. 2, pp. 137–143, Dec. 2025, doi: 10.29121/digisecforensics.v2.i2.2025.60.
[53] M. R. Kori, "IFSC-GNN: A Lightweight Graph Neural Framework for Intelligent Anomaly Detection in Cognitive Wireless Sensor Networks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 13, no. 11, pp. 2115–2120, Nov. 2025, doi: 10.22214/ijraset.2025.75558.
[54] T. K. K. Ho, A. Karami, and N. Armanfard, "Graph Anomaly Detection in Time Series: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 8, pp. 6990–7009, Aug. 2025, doi: 10.1109/TPAMI.2025.3566620.
[55] O. Okolo, B. Y. Baha, and M. D. Philemon, "Using Causal Graph Model variable selection for BERT models Prediction of Patient Survival in a Clinical Text Discharge Dataset," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 4, pp. 455–473, Mar. 2025, doi: 10.62411/faith.3048-3719-61.
[56] W. Chua *et al.*, "Web Traffic Anomaly Detection Using Isolation Forest," *Informatics*, vol. 11, no. 4, p. 83, Nov. 2024, doi: 10.3390/informatics11040083.