IJITCE

ISSN 2347- 3657

# International Journal of Information Technology & Computer Engineering


Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 11, Issue 4, Dec 2023

# A Deep Learning Method for Transformer-Based, Robust Bot Detection on Twitter

**Mr.G JACOB JAYARAJ[1] , Mrs.Geddam Prashanthi[2], Mrs.N.SOWJANYA[3], Mrs.K.ANUSHA[4],**

***Abstract—***

Due to the large number of bots on Twitter, there has to be a way to reliably and accurately identify bots, both lawful and malevolent. These approaches worked, but they didn't solve the following problems: (1) the impossibility of obtaining ground truth real-world datasets due to the large datasets needed to train a model to detect bots; (2) the difficulty of learning representations of a diverse attributed network such as Twitter; and (3) the ongoing evolution of bot accounts to avoid automatic detection. In this study, we provide ADNET, a new framework for anomaly detection in networks ascribed to Twitter with little labeled data. Our proposed topology-based active learning framework, which trains the model using a deep autoencoder and outperforms prior techniques in handling huge graphs, is an attempt to remedy the shortcomings of earlier approaches. While reducing the annotation cost in Twitter attribution networks, our experimental findings show that the suggested strategy outperforms state-of-the-art approaches in identifying anomalous bot accounts.

***Index Terms*—Twitter bot detection, Automated accounts, So- cial Media**

## I. Introduction

A Twitter bot is an automated account that uses the Twitter API to either completely or partly manage the account's behavior. As a result of their special software management, such accounts may quickly produce a flood of material. Bots may legally use Twitter according to the rules of service, provided they declare their bot status in section 1 of their profile. This benefit has been well-received by news agency accounts, which are able to consistently post a large amount of news with their followers. On Twitter, malicious bots have used this capability to disseminate spam [3], objectionable material [4], and false news [1], [2].

Associate Professor[1], Asst.Professor[2,3,4]

CSE(DS)

DRK INSTITUTE OF SCIENCE & TECHNOLOGY

ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

These bots do not reveal their bot identity on their accounts. Because harmful bots are complex, detecting them has proved difficult. The software that controls these accounts is turned on and off sporadically, and their behavior patterns are continuously altering to simulate human activity, in an effort to elude automatic detection [5]. Due to the large number of bots on Twitter, there has to be a way to reliably and accurately identify bots, both lawful and malevolent.

The current state of bot identification on Twitter is centered upon manually constructed features that use typical classification algorithms to identify bots based on profile- and tweet-related characteristics [6]-[10]. Those classification algorithms don't work consistently since bots on Twitter are always changing [11]. This is particularly true when new characteristics are required to identify the better bot behavior. It was suggested that deep learning techniques such as graph convolutional networks [15], recurrent neural networks [12], and graph neural networks [13, 14] may be used to accurately detect bot accounts. These approaches worked, but they didn't solve the following problems:

(1) the massive datasets needed to train a bot detection model, (2) the fact that bot accounts are always changing to avoid detection, which means there aren't enough ground truth real-world datasets, and (3) the difficulty of learning representations of a diverse attributed network like Twitter.

The bot identification issue is approached as an anomaly detection problem in order to address these problems. A new methodology, ADNET, is proposed to find outliers in Twitter attributed networks using little labeled data. In particular, ADNET employs a topological partitioning of the network to choose the most informative nodes inside each partition. The learner is then trained using this subset of the network, which is then inputted into the autoencoder to identify outliers. A graph transformer is used as an encoder in the autoencoder to learn graph representations, and related decoders are used to recover the topological structure and node properties. Nodes are ranked and scored using the mistakes produced by the autoencoders. After then, the labeled portion of the network is expanded with the newly created nodes until the halting requirement is satisfied. Here are the key points of our contributions:

1) building an innovative active learning framework based on attributed network topology: We provide a new active learning querying approach for attributed networks that uses community structural features to choose the most informative nodes to label and logically divides the network topology. In attributed networks, our strategy lowers the cost of annotation. We believe our framework trains the first graph transformer for anomalous user identification in Twitter attributed networks using a network partitioning active learning strategy.

Developing a system for anomaly detection using active learning

In attributed networks, anomaly detection is accomplished by training a graph transformer-based autoencoder using a small subset of labeled nodes from each partition. This subset is selected depending on the network's informativeness.

Thirdly, we conducted a thorough performance study and experimental assessment of our technique using three datasets: one containing real-world Twitter data and two datasets including

ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

benchmark attributed networks. Results showed that it was efficient and effective when compared to baselines and cutting-edge approaches.

4) Making use of Twitter data to expand three already attribution networks for the purpose of anomaly detection: Twitter bots with suspicious activity were identified using the suggested technique. Using pre-existing datasets is inadequate for building our attributed networks for every dataset. Bot accounts were included in three of the Twitter datasets that were gathered. Each accessible user account's following and followers were then added to these databases. Anomaly detection using ADNET was then applied to these Twitter associated networks in order to identify bots.

## II. Related Work

### A. *Bot Detection in Twitter*

The automated detection of bots in Twitter datasets has been approached in many ways. In order to distinguish between real and fake user accounts, supervised methods that pull features from posts and accounts depend significantly on annotated datasets [17]. On the other hand, the datasets need regular updates with new kinds of bots to be discovered, as bots are always changing their behavior to avoid automated detection [5]. Recent proposals for Twitter bot identification have made use of graph approaches. SATAR [18] detects bots by using user data and elements of Twitter's network graph structure. A GCN-based approach to bot detection was suggested by AlHosseini et al. [19]. This technique makes use of node characteristics and neighborhood information. Also, in order to identify bots, BotRGCN [20] makes use of relational GCNs to depict the Twitter network and user attributes. Despite these approaches' comparative success, their training datasets and exposure to different kinds of bots greatly affect their performance. A more accurate bot detection that is independent of data or bot type is achieved by seeing bot detection as an anomaly detection problem. This approach helps to overcome this issue.

### B. Identifying Abnormalities in Attributed Networks

Previous research in graph anomaly identification relied on feature engineering, which is effective only on labeled datasets, to identify abnormalities [21]-[23]. While there are approaches that rely on statistics, they may be rather resource and time intensive [24], [25]. Deep learning methods such as graph attention networks (GATs), generative adversarial networks (GANs), and reinforcement learning (RL) have recently emerged.

Anomaly detection has made use of adversarial networks (GANs) and graph convolutional networks (GCNs) [26]- [31]. The results of graph anomaly detection employing deep learning approaches are superior, according to a recent survey [32]. Researchers have begun to place a greater emphasis on anomaly identification on attributed networks due to the proliferation of attributed network datasets. When it comes to detecting anomalies in attributed networks, newer solutions use deep learning algorithms. AMEN [33] uses the ego networks of each node to identify anomalous areas in attributed networks. The residuals of attribute information and their coherence with network information are characterized by radar [24] for anomaly detection. ANOMALOUS [34] uses cut matrix decomposition and residual analysis to perform attribute selection and anomaly identification. The attribute is reconstructed using a GCN autoencoder in DOMINANT [27], and anomalies are ranked using the aggregated error from the adjacency matrix. These techniques are effective at finding outliers in attributed networks, but they are quite resource intensive while learning and

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

perform poorly on bigger graphs. Our suggested methodology outperforms the state-of-the-art, particularly on large attributed networks, by using topology-based active learning to detect abnormalities.

C. Learning Through Doing

In order to learn the right prediction for a specific issue, active learning algorithms iteratively choose data points to annotate by asking an oracle [35]. The accuracy of an AL model improves with the exposure to more data samples. For AL sampling methods, there are a few options: either querying diverse instances one at a time, which can lead to overfitting [36], or using a combination of the two, where a budget is assigned to the querying algorithm and then used to choose a batch of nodes for labeling [37]. This approach yields better results for deep learning models. Anomalies involving arrhythmias have been detected using AL in medical datasets for anomaly identification [38]. In order to find outliers in environmental datasets, Russo et al. [39] use active learning techniques with machine learning algorithms. Each of these publications demonstrates how their respective areas dealt with the label sparsity problem by using AL. The models used in earlier studies that used AL on attributed networks overfit since they only questioned individual nodes [40], [41]. By establishing a budget and then selecting a batch of nodes using the combined querying approach, this study was able to reduce training resource consumption and avoid overfitting. Anomaly detection utilizing attributed networks has not been the subject of any research, despite the fact that AL has been the subject of various investigations.

III. PROPOSEDSYSTEM

The rising use of huge attributed networks as a datasource is driving the need for this framework, since training models utilizing them may be rather expensive. We provide a system that uses the associated network graph to create divisions, selects determines which nodes are most informative for querying and incorporates them into model training. This new approach uses structural graph representations for anomaly detection in the autoencoder and decreases the amount of labeled data needed to train models.

*A. Problem Statement*

**Active Learning on Attributed Networks Definition** An attributed network is typically represented as $G = \{V, E, T\}$ with its adjacency matrix $A \in R^{N\times N}$ and node labels $Y$, where $V$ denotes the set of nodes, $E$ denotes the set of edges, $T = [t_1, t_2, \ldots, t_M] \in R^{N\times M}$ represents the attributes matrix, and each vector $t_i \in R^N$ represents the nodal

The structure reconstruction decoder uses the representa- tions to predict if a link exists between pairs of

nodes by training a

link prediction layer of the learned representations and its transpose shown in Equation 4.

$$H'^{(k)} = \text{TransformerSelfAttention}(H^k) \quad (2)$$

$$H^{k+1} = \text{GCN}(A, H'^{(k)}) \quad (3)$$

$$\hat{A} = \text{sigmoid}\ \mathbf{H^{k+1}H'^{(k+1)}} \quad (4)$$

Nodal connectivity patterns are used as an indicator of the node being anomalous by calculating the reconstruction error $E_s = A - \hat{A}$ where $\hat{A}$ is the estimated adjacency matrix. A higher norm value for $E_s$ means that the node has a higher probability of being an anomaly with regard to the network structure. The attribute reconstruction decoder takes the learned representations $H^{k+1}$ from the encoder to approximate nodal attributes information by computing reconstruction errors. To predict the original nodal attributes, we leverage a GCN denoted as in equation 5,

$$\mathbf{X} = f_{Relu}(\mathbf{H^{k+1}}, \mathbf{A} \mid \mathbf{W}^{(m)}) \quad (5)$$

V , and an initial labeled set of nodes $S_0$, the goal

ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

is to determine which unlabeled nodes should be selected to label when given a fixed budget b that produces a model M with
the lowest loss:

$$\arg\min_{S_1 \subset V} L_\theta(Y_{S0} \cup S_1)$$
$$\text{s.t.} \quad n(S_1) \leq b \tag{1}$$

where $W^{(m)}$ is a trainable layer weight matrix needed to learn the network representation. The GCN is then used in computing the reconstruction errors $\mathbf{E_A} = \mathbf{T} - \mathbf{T}$ to detect anomalies with regards to nodal attributes.

(1)The reconstruction errors calculated in the structure recon- struction decoder and the attribute reconstruction decoder are used to detect anomalies in the attributed network. In order to account for both nodal attributes and graph structure in our

where $Y_{S0}$ is the set of existing pre-labeled nodes, $S_1$ is the set of unlabeled nodes we want to label, $n(S_1)$ is the cardinality of the set $S_1$, b is the budget for the labeling, θ is the parameter of the anomaly detection model learned from the labeled set
(the union of $S_1$ and $Y_{S0}$ ), and L is the loss of the anomaly detection problem conditioned on model parameters θ and labeled data.

### B. *Preliminaries*

Our ADNET framework shown in Figure 1, leverages a deep autoencoder with a graph self-attention encoder to enhance AL anomaly detection results in large attributed networks.

**Self-Attention Based Anomaly Detection for Graph Data** Unlike previous works that use GCNs [27] to encode graph data into representations, this method adopted a graph transformer [42] as a self-attention encoder that learns graph representations for anomaly detection in our AL framework. Due to the heterogeneous nature of the attributed network data, it is essential to preserve both node embeddings and graph structures to be able to identify anomalous nodes.

The T ransformerSelf Attention() in Equation 2 is uti- lized to learn vector representations of all nodes for the given graph G, then outputs $H'^{(k)}$ which is used in the GCN model to improve the vector representations of nodes by adding the structure of the graph G and produces a graph embedding as the output of the encoder. attributed network, the model jointly learns the reconstruction errors by minimizing the deep autoencoder objective function:

$$L = (1-\alpha)\mathbf{E}_S + \alpha\mathbf{E}_A \tag{6}$$

where the controlling parameter α bal^ances the reconstruction impacts. Consecutively, the approximation of the attributed network is iteratively calculated until the objective function converges and the reconstruction errors are calculated to rank nodal abnormalities. The anomaly score for each node can be computed as

$$\text{ascore}(\mathbf{v}_i) = (1-\alpha)\mathbf{es} + \alpha\mathbf{e_A} \tag{7}$$

Our work utilized the deep autoencoder structure and was developed with a graph-based self-attention encoder which allowed us to incorporate graph structures and nodal attributes in our model properly. The graph-based self-attention used in the encoder learns complex graph representations and preserves both node embeddings and graph structures better than using only GCNs or only graph transformers to identify anomalous nodes.

### C. *ADNET Framework Description*

**ADNET** is an active learning anomaly detection frame- work for attributed networks. The architecture of our pro- posed framework is illustrated in Figure 1. The objective of ADNET is to select the most informative nodes from

large attributed networks to be labeled such that the anomaly detection performance is improved with minimal labeling cost. Therefore, topology-based AL was incorporated with the deep autoencoder, which maps the attributed network into a latent low-dimensional feature space, and then recovers the original data based on latent representations to detect anomalies based on the computed reconstruction errors. The workflow

ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

of the ADNET framework shown in Fig. 1 and Algorithm 1 can be described as follows:

1) Given a graph G, the initial labeled set L, the unlabeled set U, and the budget η as input. The topology-based attributed network sampling in algorithm2 is performed.
2) After the graph partitions, cluster centroids, and the most informative nodes are defined, we annotate the selected nodes based on their partition and merge them with the labeled set of nodes L. We also subtract the selected nodes from the unlabeled set U.
3) Finally, we train the autoencoder model with the labeled set of nodes L that are queried by algorithm 2. The graph-based self-attention encoder takes the attributed network as an input, and learns the graph representations by learning both node embeddings and graph structures. The output vector is then passed on to the decoders, where the structure reconstruction decoder reconstructs the graph topology, and the attribute reconstruction de- coder reconstructs the nodal attributes using the learned graph embeddings. The reconstruction errors would be used to rank the nodes based on their anomaly scores, where the top nodes are considered anomalous.

Using this approach, the strengths of active learning and deep autoencoders are combined to minimize the amount of labeling needed in large attributed networks and maximize the anomaly detection task performance.

**Algorithm 1** Active Learning Anomaly Detection for At- tributed Networks

**function** ACTIVEANOMALY(L, U, G)

Given : the initial labeled set L, the unlabeled set U, the graph G, the partition number K, budget η, trade-off parameter α :

S ←TopologyBasedANSampling(L, U, G, K, η, α) L ← L ∪ S

U ← U − S

lambda ←train(L,G) //train model M with labeled samples acquired from topology sampling

**Topology-based Attributed Network Sampling**

A novel query strategy algorithm is shown in algorithm 2, which conducts a topology-based attributed network sampling. In- spired by previous works that consider community detection to partition well-defined networks [37], [43], the quality of each partition was measured as a function of modularity and purity. Existing community detection methods are applied to fully labeled graphs; this challenge was addressed by assigning unlabeled nodes to communities according to the average

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

**Algorithm 2** Topology-based Attributed Network Sampling

**Input:** A graph G, the initial labeled set L, the unlabeled set U, budget η, partition number K, trade-off parameter α

**Output:** A subset of unlabeled nodes $S_1$ of size $\eta : S_1 \subseteq V \backslash S_0$

$H_K \leftarrow$ GraphPartition(K)
Set $S_1 = \emptyset$. D to hold the subset of unlabeled nodes
**for** $H_k \in H_K$ **do**
  $\eta_k \leftarrow \eta//K$
  $H_k \leftarrow H_k \backslash \{S_0 \cup S_1\}$ $E_k \leftarrow \{g(v_i)\}_{i \in H_k}$ $L_k \leftarrow L \cap H_k$
  $U_k \leftarrow U \cap H_k$
  $G_k$ = Generate($G$, $M_k$) D generating partitions
  $C \leftarrow$ Initialize($G_k$) $Z \leftarrow \alpha P + (1 - \alpha)Q$
  $Z_{prev} \leftarrow -\infty$
  **while** $Z > Z_{prev}$ **do**
    $C \leftarrow$ PartitionNodes($G_k$, C, α) D greedily identifies partitions by maximizing modularity and purity
    $G \leftarrow$ Aggregate($G_k$, C) D Network reconstruction and moving nodes to their partitions
    Compute P according to Eq. (10) and Q according to Eq. (8)
    $Z_{prev} \leftarrow Z$
    $Z \leftarrow \alpha P + (1 - \alpha)Q$
  **end while**
  **for** $v_i \in U_k$ **do**
    $v_i \leftarrow$ AssignCommunity($v_i$, C)
    $C \leftarrow \{C \cup v_i\}$ D Update the community
  **end for**
  **end**
  $CT \leftarrow$ FindCentroids(C) D Compute the centroids of the communities
  $S \leftarrow$ ClosestNodes(CT, $U_k$, $\eta_k$) D Find $\eta_k$ unlabeled nodes closest to the centroids.
  **if** $S \neq \emptyset$ **then**
    $S_1 = S_1 \cup S$
  **end if**
  **return** S

similarity between the unlabeled node and all nodes in a partition. The sampling strategy partitions the graph G into K-partitions following the method described below. Then, for each partition, topology-based community detection was conducted on labeled nodes. As for unlabeled nodes, they are assigned to their corresponding communities based on their similarity to a community calculated using equation
12. Finally, the unlabeled nodes closest to each community centroid were selected as the most informative nodes to use to train our model M in algorithm 1.

*1) Topology-based Attributed Network Partition Method:* In order to correctly identify the partitions in our attributed networks, the quality of the partition was calculated as a function of modularity and purity.

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

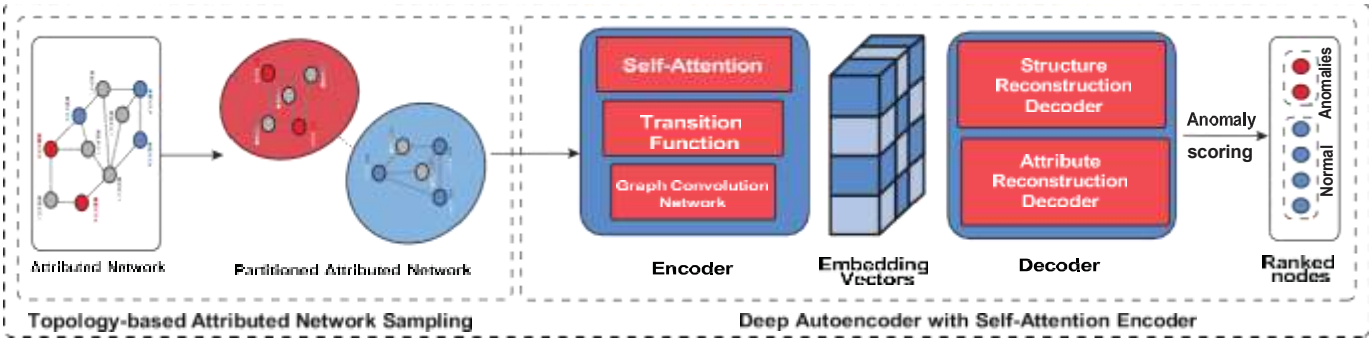


Figure 1: An overview of our proposed framework, ADNET, where a graph transformer-based autoencoder is trained to detect anomalies from a subset of the network chosen based on the most informative nodes in each partition

**Modularity** is a quality function that measures the degree to which connected nodes within a network can be decoupled into communities or partitions. The equation for modularity can be denoted as:

$$Q = \frac{1}{2m} \sum_{vw} \left[ A_{v,w} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w) \quad (8)$$

*2) Most Informative Nodes Selection:* Once the partitions are detected, unlabeled nodes are assigned to the communities based on their similarity to a partition (algorithm 2 line 20). Instead of measuring the similarities between all nodes $v_j$ in a partition $C_k$, we pre-define c as a character vector of a given node and evaluate the cost of each node in the partition $C_k$. As c can be realized in many ways, we use the average of all

$$\arg\min_{k=1,\ldots,K} f(g(v_i), c) \quad (12)$$

where m is the number of graph edges, $A_{v,w}$ is the adjacency matrix for $v$, $w \in V$, $k_v$, $k_w$ is the degree of v, w, and $\delta(c_v, c_w)$ is the function determining if a node v, w belongs to the same partition with a value of 1 or 0 if the node doesn't belong to the partition.

**Purity** is a measure of the extent to which a partition contains a single class. The purity of a partition C is the average purity of all communities in the partition, computed as:

$$Y = \sum_{c \in C} \max_{v \in c} a(v)$$

The centroids for each community are then computed using K-Means [45] by calculating the mean value for all nodes in $P_c$ =the community and selecting $\eta_k$ unlabeled nodes that are clos- est to the community

where $f(\cdot, \cdot)$ and $g(\cdot)$ are distance measure function, and aggregation function respectively. Jensen-Shannon divergence [44] for the f function. The g function is chosen according to our GCN model in the topology-based attributed network sampling ($g(v_i) = (A^2F)$ where A is the normalized adjacency matrix and F is the feature matrix). The c is defined as

$$c = \frac{1}{|C_k|} \sum_{c \in C} g(v)$$

where P is maximized when nodes in the same community share the same label and $P_c$ is the frequency of the most common class in the network present in one partition. Purity follows:

$$P = \frac{1}{\text{len}(C_k)} \quad (13)$$

for a given community c is denoted as:

ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

centroids. These nodes are considered

where A is the label set, a ∈ A is a label, a(v) is an indicator function that takes value 1 if a ∈ A(v). The modularity and purity were combined linearly as in Eq. 11, by applying a trade-off parameter α, to tune the importance of each component and adapt the score according to different attributed networks.

$$Z = \alpha P + (1 - \alpha)Q \quad (11)$$

The basic idea is that nodes in the network try to traverse the community labels of all neighbors and select the community label that maximizes the modularity and purity. After maxi- mizing the modularity and purity, each community is treated as a new node, and the process is repeated until the modularity no longer increases. This method is suitable for large-scale networks.

the most informative nodes due to their close proximity to the centroid.

## III. Experiments

We present our empirical evaluations using three real-world Twitter-attributed networks and three benchmark attributed network datasets to verify the effectiveness of our proposed framework ADNET. We evaluate each baseline with a labeling budget and report the AUC-ROC score for anomaly detection over the attributed network.

### *A. Datasets*

We evaluate the proposed framework on three real-world datasets collected from Twitter [46], [47]. In addition, to compare our results with anomaly detection results, we evalu- ate our model on three widely used benchmark datasets for

anomaly detection on attributed networks [24], [27]. In an attempt to present three new real-world attributed network datasets for anomaly detection, we use the Twitter API [2] to collect the tweets, user information, following, and followers networks for each user in the datasets found in [46], [47]. It is worth noting that since some of these datasets are old and Twitter removed some accounts, we experienced an average loss of 38% from the original Twitter dehydrated datasets. The statistical summary of all datasets is demonstrated in Table I. Moreover, since there is no ground truth for anomalies in the benchmark datasets (CiteSeer, Pubmed, and ACM), the anomalies are injected [27], [48]. Whereas in the Twit- ter datasets, we treat bots, the automated user accounts, as anomalies.

- **verified-2019 & botwiki-2019** We combine two datasets found in [47], one is a verified dataset of human accounts, and the other is a self-identified list of bots. We collect attributes related to their account, following and follower relations, and recent tweets for each user.
- **cresci-rtbust-2019** This dataset is composed of users who participated in retweeting Italian tweets over a two- week period in 2018 [46]. We collect attributes related to their account, following and follower relations, and recent tweets for each user.
- **gilani-17** This dataset is manually labeled 'bot' or 'hu- man' based on hand crafted rules [49]. We collect at- tributes related to their account, following and follower relations, and recent tweets for each user.
- **CiteSeer** [48] is a public citation network. Each node is a published paper, while each edge denotes a citation relation between two papers. The textual contents of each paper are treated as its node features.
- **Pubmed** [48] is a public citation network. Each node is a published paper, while each edge denotes a citation relation between two papers. The textual contents of each paper are treated as its node features.
- **ACM** [50] is a citation network of papers published in nine areas before 2016. The dataset was turned into an undirected graph due to the sparsity of the original network.

### *B. Baseline Methods*

We compare our proposed framework ADNET

ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

with the following baselines:

- **Botometer** [17]: A bot detection API service that uses a thousand user features in its analysis.
- **Alhosseini** [19]: A GCN approach to learn user repre- sentations for bot detection.
- **SATAR** [18]: A self-supervised representation learning framework that leverages user-related features for bot detection.
- **BotRGCN** [20]: A relational GCN approach for repre- sentation learning and bot detection.

[2]https://developer.twitter.com/en/docs/twitter-api

- **DOMINANT** [27]: State-of-the-art deep model that ex- plicitly models the topological structure and nodal at- tributes for node embedding learning using GCNs.
- **ANOMALOUS** [34]: A joint framework to conduct at- tribute selection and anomaly detection as a whole based on CUR decomposition and residual analysis.
- **Radar** [24]: An unsupervised learning framework used to characterize the residuals of attribute information and its coherence with the network information for anomaly detection in attributed networks.
- **Graph Transformer**: A variant of DOMINANT [27] that we create. It is a deep autoencoder model that captures the topological structure and nodal attributes for node embed- ding learning using a graph-based self-attention encoder to be used in attributed networks anomaly detection tasks.

*C. Evaluation Metrics*

We choose **AUC-ROC, Precision@N, and Recall@N** as our evaluation metrics since they are widely used in anomaly detection research methods [24], [27], [34].

- **Precision@N:** Precision at n is the proportion of anoma- lies in the top-n nodes in the ranked list.
- **Recall@N:**Recall at n is the proportion of true anomalies found in the total number of ground truth anomalies.
- **AUC-ROC:** The AUC-ROC curve is a classification per- formance measure at multiple thresholds. The probability curve, ROC, and AUC represent the capability of ranking an abnormal node higher than a normal node. This means that as the AUC value gets closer to 1, the model is better at ranking anomalies.

*D. Parameter Setting*

In the experiments on our different datasets, we used Adam [51] as an optimizer to minimize the loss function. We trained the proposed model with 300 epochs with a learning rate of 0.005. For the graph transformer encoder, we set the dropout to 0.1, the number of self-attention layers to 2, the number of GCN layers to 2, and the number of heads to 1. For our topology-based active learning sampling method, we choose a budget of 40 nodes from the unlabeled data for the citation datasets and a budget of 210 nodes for the Twitter datasets.

*E. Experimental Results*

In the experiments, we evaluate the performance of our proposed model in detecting anomalies by comparing it with the baseline methods. The precision and recall results for the benchmark datasets are presented in Table II for a budget of 210, and 40 nodes for the Twitter datasets and the citation datasets, respectively. Fig. 2 compares the AUC-ROC results of ADNET with the baselines. We present a sample of the results on two datasets due to page limitations; it is worth noting that all of our results exhibit similar trends. According to the results in tables II, Fig.2, and Fig. 3, we have the following observations:

- ADNET is able to detect bots (Twitter anomalies) better than the baseline methods. It significantly outperforms the

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

Table I: Attributed networks datasets details

| | verified-2019 & botwiki-2019 | cresci-rtbust-2019 | gilani-17 | CiteSeer | ACM | PubMed |
|---|---|---|---|---|---|---|
| # Nodes | 33,321 | 824,902 | 4,239 | 3,327 | 16,484 | 19,717 |
| # Edges | 671,907 | 824,272 | 16,956 | 4,732 | 71,980 | 44,338 |
| # Attributes | 17,509 | 42,051 | 400 | 3,703 | 8,337 | 500 |
| # Anomalies | 704 | 891 | 1,090 | 150 | 600 | 600 |

bot detection baselines. In addition, DOMINANT, Radar, ANOMALOUS, and the Graph Transformer models did not achieve satisfactory results on Twitter data.

- When using 210 and 40 nodes as a budget to label the datasets, we find that on all attributed network datasets, our proposed framework, ADNET, achieves the best anomaly detection performance in terms of Precision@N, Recall@N, and AUC-ROC. In particular, compared to the best results from the baselines, ADNET obtains a signifi- cant improvement on AUC-ROC. The main reason is that ADNET successfully learns from the most informative nodes queried and captures the nodal attributes and the graph structure, which enables the framework to achieve better performance when detecting anomalies.
- ADNET exhibits superior performance over the baselines
confirming that training node selection from graph parti- tions enhances the active learning performance.
- Botometer, Radar, and Anomalous perform poorly com- pared to deep models. These shallow models are not able to capture the nodal and structural complexities in large attributed networks.
- The performance of the baseline methods deteriorates as the size of the attributed network grows. It is evident in the Twitter datasets where ADNET outperforms the baselines with a significant increase in performance.
- When graph-based self-attention is used as an encoder in the auto-encoder model (similar to Graph Transformer and ADNET), it outperforms the baselines. The reason is that it preserves both node embeddings and graph struc- tures better than GCN-based encoders like DOMINANT.
- When comparing the performance of different methods with respect to increasing labeling budgets, all methods' performance increases as the labeling budget increases. Though ADNET offers the most significant improvement over other baselines.

## IV. Conclusion

Here, we present ADNET, a new framework for anomaly detection in networks ascribed to Twitter that makes advantage of active learning. Our proposed topology-based active learning framework overcomes the shortcomings of other approaches, trains the model using a deep autoencoder, and outperforms them in handling big graphs. The suggested solution decreases the annotation cost in Twitter-attributed networks and beats state-of-the-art algorithms in identifying anomalous bot accounts, according to our experimental findings.

## References

[1] One source is "What the fake?" by A. Al-

ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

Rawi, J. Groshek, and L. Zhang. online information review (2018), evaluating the role of bots and networked political spam on Twitter for the spread of #fakenews. In his 2019 article "The Gulf Information War— Propaganda, Fake News, and Fake Trends: The Weaponization of Twitter Bots in the Gulf Crisis," M. O. Jones discusses the use of Twitter bots as weapons during the Gulf War.
In the 2010 IFIP Annual Conference on Data and Applications Security and Privacy, A. H. Wang presented a machine learning method for identifying spam bots on social media websites (pp. 335–342). The paper was published by Springer.

There will be an increase in the misuse of social media bots, according to O. Analytica. The Emerald Group published the briefing in 2021 with the number oxan-db.

"Detecting and character-izing bot-like behavior on Twitter," in 2018's International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation, pp. 228-232, edited by S. Qi, L. AlKulaib, and D. A. Broniatowski (Springer), is cited as [5].

[6] "Online human-bot interactions: Detection, estimation, and characterization," in 2017's Eleventh annual AAAI conference on online and social media, by O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini.

[7] "Feature engineering for social bot detection," in Feature engineering for machine learning and data analytics (pp. 311-334), edited by O. Varol, C. A. Davis, F. Menczer, and A. Flammini, published by CRC Press in 2018, page 311-334.

[8] Ko-Chang Yang, Onno Varol, Charles A. Davis, Elena Ferrara, Alberto Flammini, and

"Providing citizens with AI tools to fight social bots" (F. Menczer), Human Behavior and Emerging Technologies, 1, 1 (January 1, 2019).

2019 (pages 48-61).

This sentence is paraphrased from an article published in 2016 by IEEE Intelligent Systems and titled "Dna-inspired online behavioral modeling and its application to spambot detection." The authors of the article are S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi.

The paper "Twitter spammer detection using data stream clustering" was published in 2014 by Z. Miller, B. Dickinson, W. Deitrick, W. Hu, and A. H. Wang in the journal Information Sciences.

in "Weaponized health communication: Twitter bots and russian trolls amplify the vaccine debate," D. A. Broniatowski, A. M. Jamison, S. Qi, L. AlKulaib, T. Chen, A. Ben-ton, S. C. Quinn, and M. Dredze, 2018 vol. 108, no. 10, pp. 1378-1384, American journal of public health.

[12] "Deep neural networks for bot detection," by S. Kudugunta and E. Ferrara,

Chapter 467, pages 312–322, 2018 in the Information Sciences.

Rosengas: Adaptive social bot identification with reinforced self-supervised gnn architecture search was published in an arXiv preprint in 2022 and was co-authored by Y. Yang, R. Yang, Y. Li, K. Cui, Z. Yang, Y. Wang, J. Xu, and H. Xie.

It was written by Y. Li, Y. Ji, S. Li, S. He, Y. Cao, Y. Liu, H. Liu, X. Li, J. Shi, and others.

With reference to the 2021 International Joint Conference on Neural Networks (IJCNN), Y. Yang presented a paper titled "Relevance-aware anomalous users

ISSN 2347–3657
Volume 11, Issue 4, Dec 2023

detection in social network via graph neural network" (pp. 1-8, IEEE, 2021).

The paper "Botrgcn: Twitter bot detec-tion with relational graph convolutional networks" was presented at the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. The authors are S. Feng, H. Wan, N. Wang, and M. Luo. The work is referenced in the proceedings.

The following is a reference to a paper published in 2021 by Q. Guo, H. Xie, Y. Li, W. Ma, and C. Zhang: "Social bots detection via fusing bert and graph convolutional networks," Symmetry 14, no. 1, p. 30.

Referenced in "Botornot: A system to evaluate social bots" (C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, 2016), in Proceedings of the 25th international conference companion on world wide web, pages 273-274.

The paper "Satar: A self-supervised approach to Twitter account representation learning and its application in bot detection" was presented at the 30th ACM International Conference on Information & Knowledge Management in 2021 and was written by S. Feng, H. Wan, N. Wang, J. Li, and M. Luo.

[19] "Detect me if you can: Spam bot detection using inductive representation learning," by S. Ali Alhosseini, R. Bin Tareaf, P. Najafi, and C. Meinel