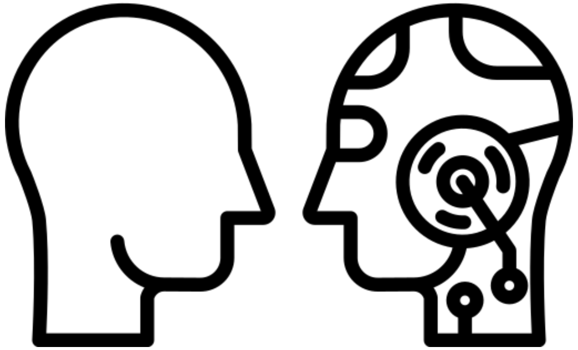
Twitter Bot Identification: An Anomaly Detection Approach

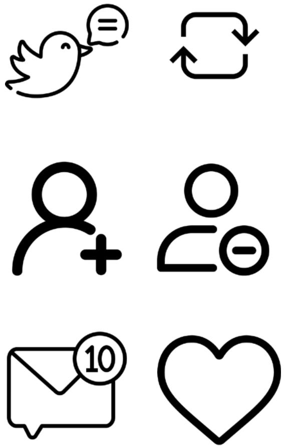
Lulwah AlKulaib, Yanshen Sun, Lei Zhang, and Chang-Tien Lu

Introduction: Twitter Bot Detection

- Bot or Not?

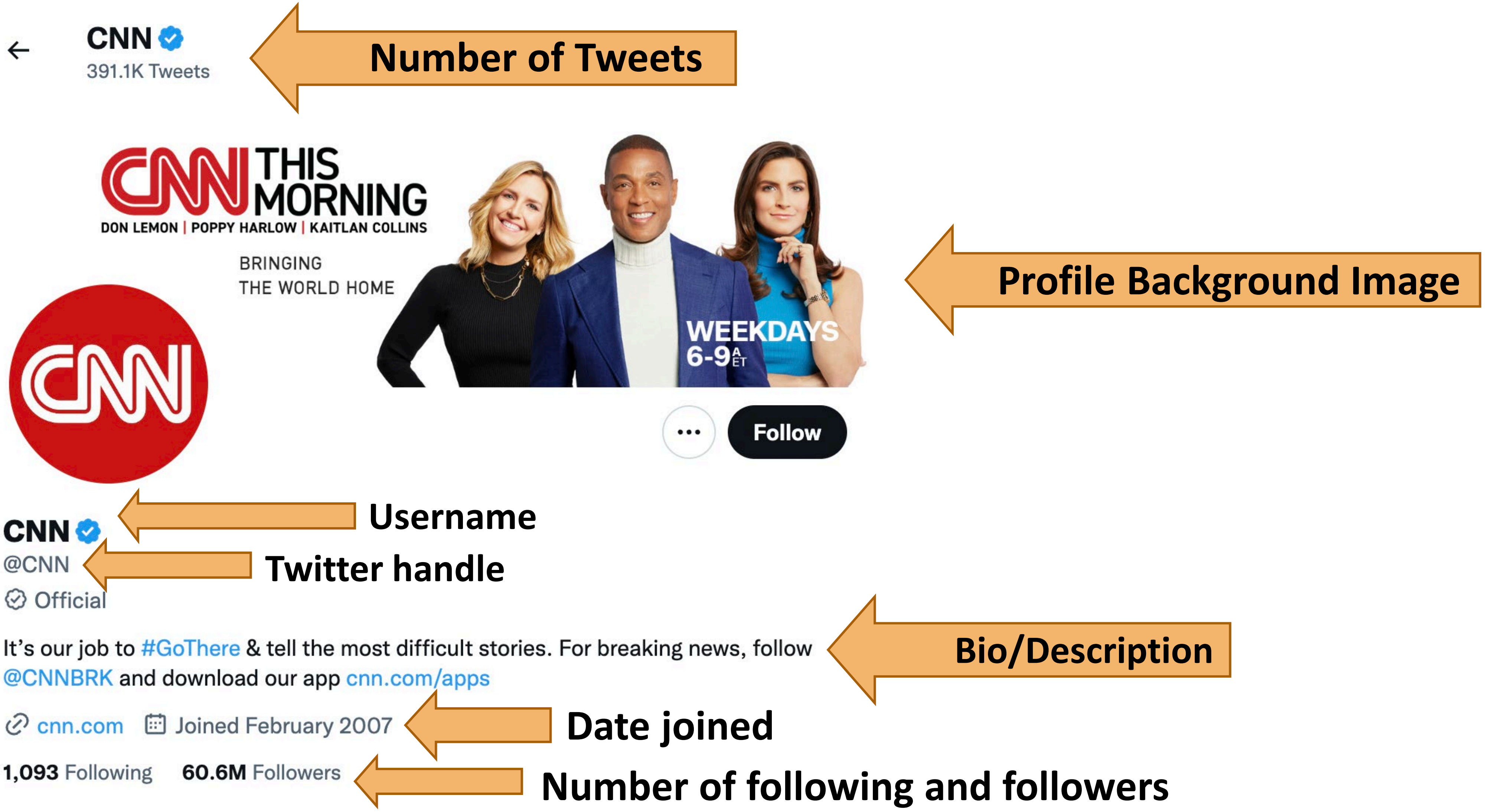


- Bots capabilities



- Why is it important to identify bot accounts?
  - Some helpful bots
  - Many malicious bots
    - Spreading misinformation
    - Scams and exploitation

Introduction: Twitter Bot Profile Information



Motivation

- The cost to train models with large attributed networks is very high.
- Bot accounts’ behavior is ‘anomalous’ when compared to regular human behavior.
- Challenges:
  - How to choose the most informative user nodes in training our model?
  - How to accurately detect bot accounts when their behavior evolves constantly to evade automatic detection?

Existing Works: Bot Detection

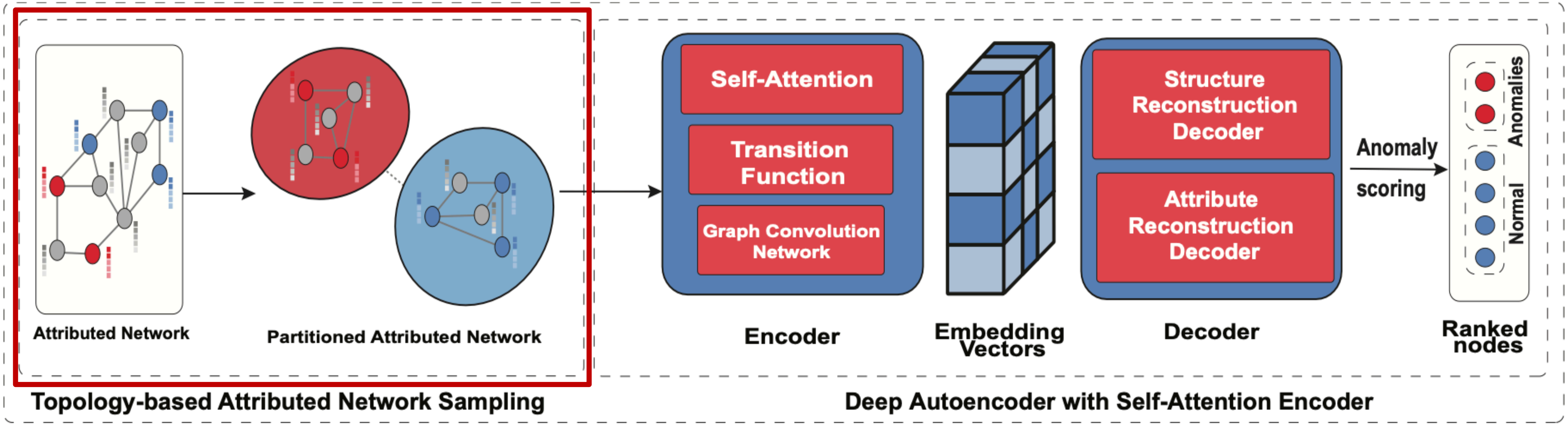
- Existing Studies:
  - [WWW'16] Rely on manually annotated datasets.
  - [CIKM'21] Handcrafted features.
  - [ASONAM'21] Relational GCNs.

- All these methods:
  - Require huge amounts of training samples.
  - Their performance drops drastically when using large graphs as an input.

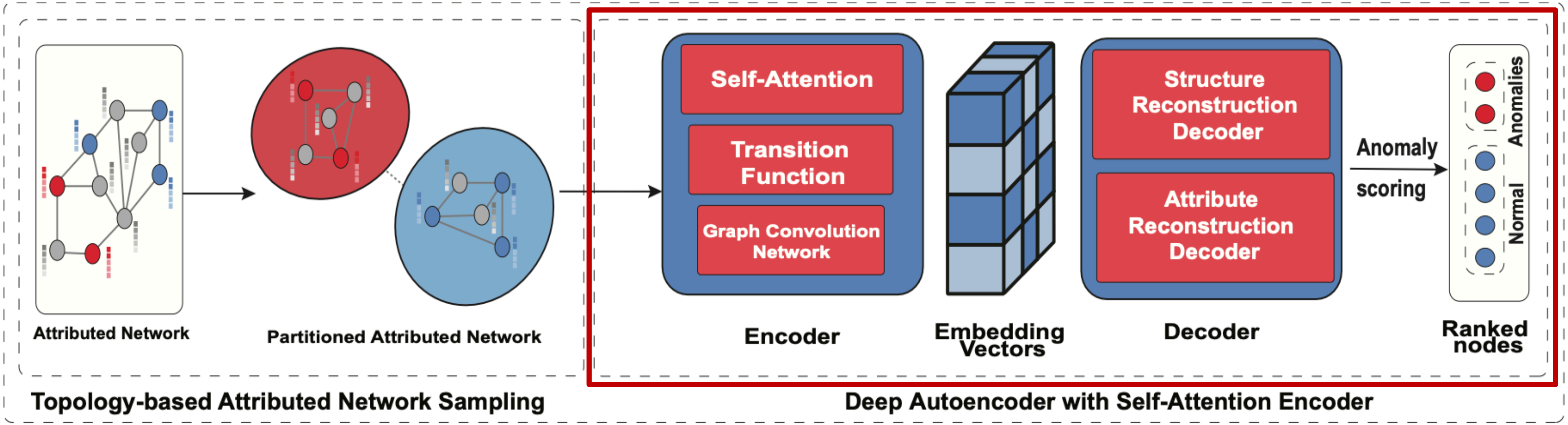
Overview: ANDET

- Contributions:
  - Development of a novel attributed network topology-based active learning framework:
    - ANDET: an active learning anomaly detection framework for attributed networks.
    - Objective: select the most informative nodes to be labeled such that the anomaly detection performance is improved with minimal labeling cost.
  - Design of an active learning algorithm for anomaly detection in attributed networks
  - Extensive experimental evaluation and performance analysis
  - Extension of three existing real-world attributed networks for the anomaly detection task using Twitter data

Proposed Model



Proposed Model



Proposed Model: Algorithm

Algorithm 1 Active Learning Anomaly Detection for Attributed Networks

0: **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 :
0: $S \leftarrow$TopologyBasedANSampling$(L, U, G, K, \eta, \alpha)$
0: $L \leftarrow L \cup S$
0: $U \leftarrow U - S$
0: $lambda$ $\leftarrow$train(L,G) //train model M with labeled samples acquired from topology sampling
=0

1. Partitions the graph into K-partitions
2. For each partition, perform topology-based community detection on labeled nodes
3. Assign unlabeled nodes to communities based on their similarity
4. Select unlabeled nodes that are closest to each centroid as the most informative node to train the model

Topology-based Attributed Network Sampling

- Topology-based attributed network partitioning method:
  - Modularity:

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

Degree of connected node decoupling into a community

  - Purity:

Purity of a partition

$$P = \frac{1}{|C|} \sum_{c \in C} P_c$$

$$P_c = \prod_{a \in A} \frac{\max\left(\sum_{v \in c} a(v)\right)}{|c|}$$

Purity for a given community

- Most Informative Nodes Selection:

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

Experiments: Dataset

TABLE I: Attributed networks datasets details

	verified-2019 & botwiki-2019	cresci-rtbust-2019	gilani-17	CiteSeer	ACM	PubMed
# Nodes	53,321	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
	Twitter Datasets			Citation Datasets		

Experiments: Baselines

- **Bot Detection Methods:**
- Botometer
- Alhosseini
- SATAR
- BotRGCN

- **Graph—based Anomaly Detection Methods:**
- DOMINANT
- ANOMALOUS
- Radar
- Graph Transformer

Experiments: Evaluation Metrics

- **Precision@N:**

  The proportion of anomalies in the top-N nodes in the ranked list.

- **Recall@N:**

  The proportion of true anomalies found in the total number of ground truth anomalies.

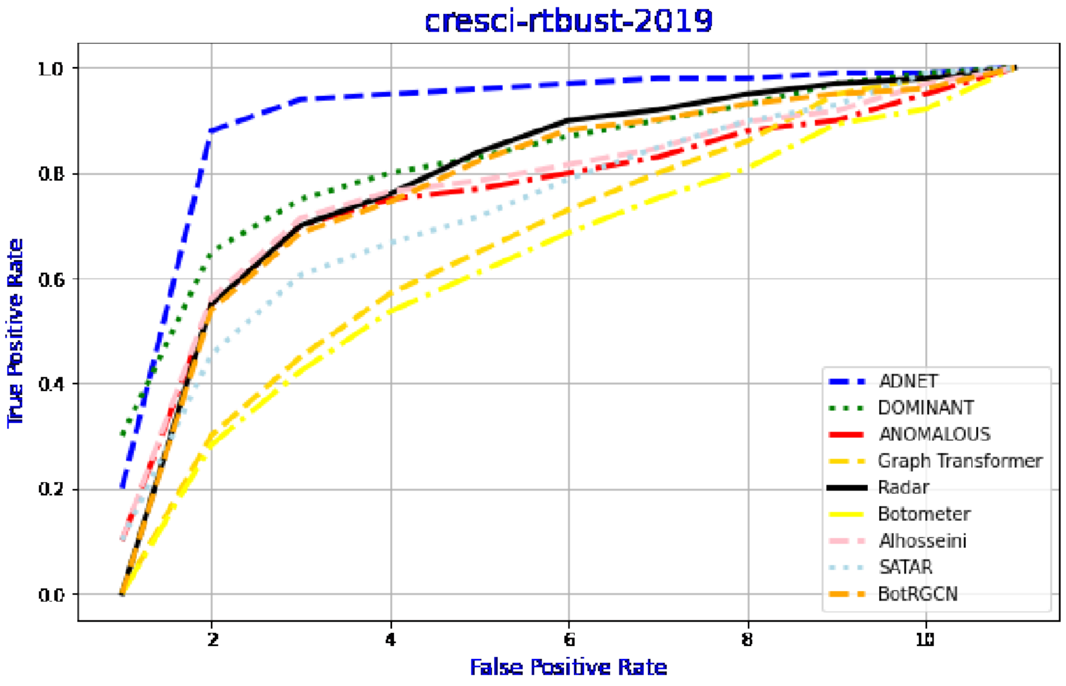
- **AUC-ROC:**

  A classification performance measure at multiple thresholds.

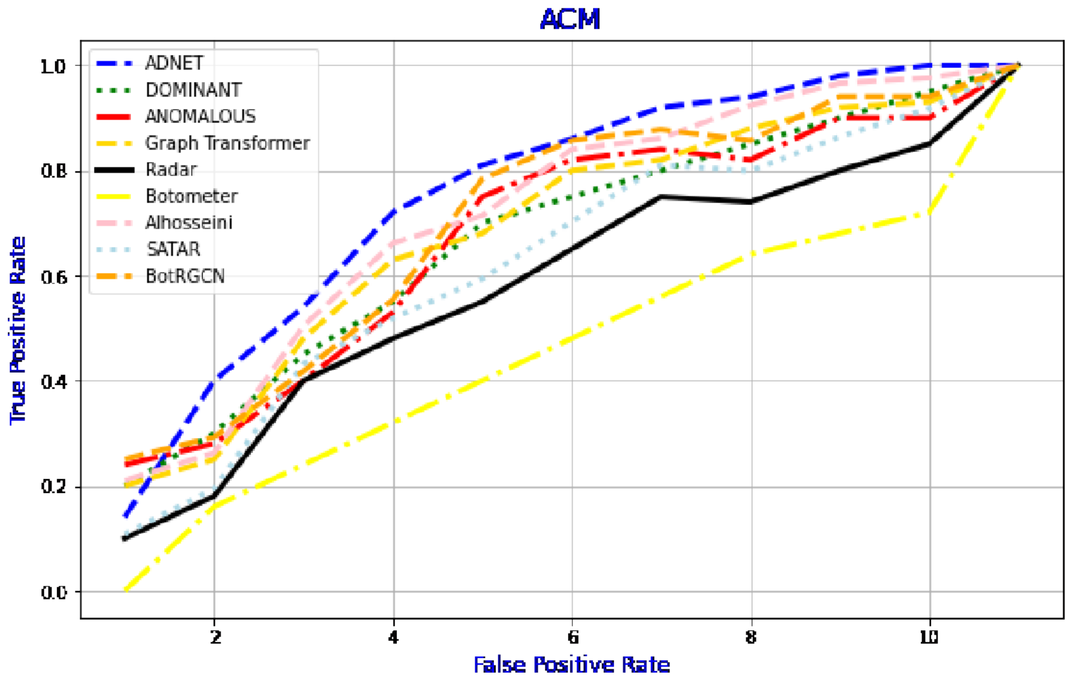
  The probability curve, ROC, and AUC represent the capability of ranking an abnormal node higher than a normal node.

  This means that as the AUC value gets closer to 1, the model is better at ranking anomalies.

Experiment Results



(a)



(b)

Fig. 2: ROC curves and AUC scores of all methods on different datasets

Experiment Results

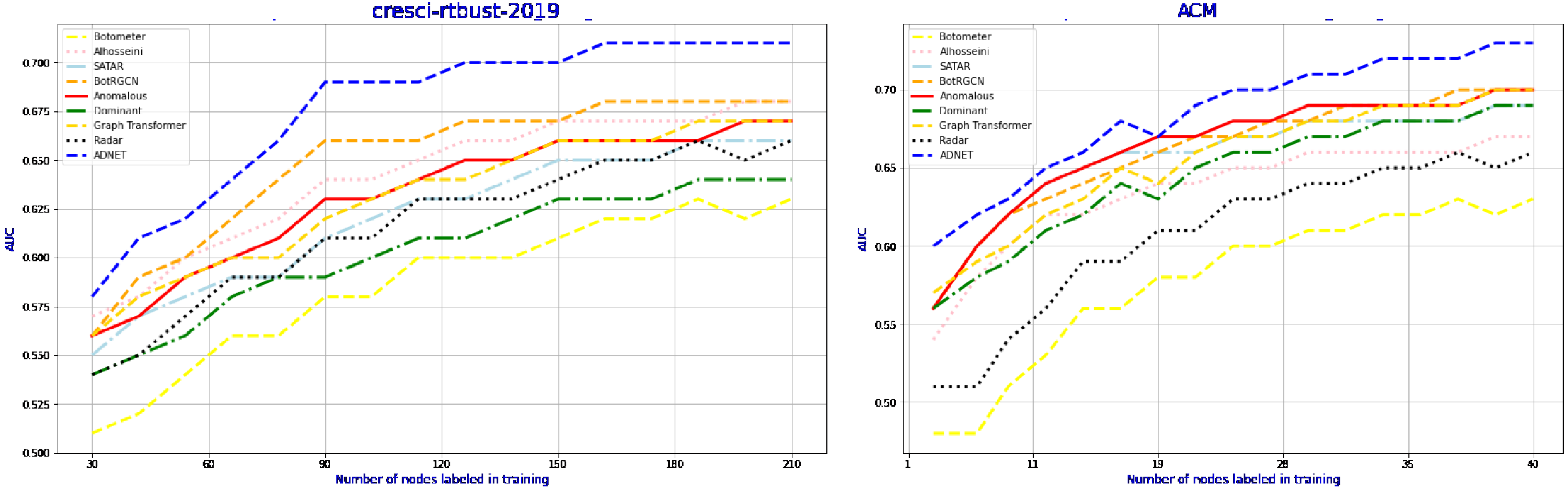


Fig. 3: Performance comparison using different labeling budgets

Experiment Results

(a) Benchmark Datasets (Twitter Data)

	verified-2019\ botwiki-2019				cresci-rtbust-2019				gilani-17			
N	50	100	200	300	50	100	200	300	50	100	200	300
Precision@N												
Botometer	0.148	0.129	0.324	0.343	0.176	0.231	0.353	0.390	0.188	0.248	0.377	0.418
Alhosseini	0.410	0.399	0.546	0.589	0.368	0.378	0.410	0.453	0.394	0.405	0.439	0.485
SATAR	0.399	0.495	0.536	0.600	0.347	0.411	0.485	0.496	0.371	0.440	0.519	0.530
BotRGCN	0.454	0.576	0.643	0.677	0.400	0.555	0.600	0.656	0.428	0.594	0.642	0.702
Radar	0.153	0.134	0.335	0.354	0.182	0.239	0.364	0.403	0.194	0.255	0.389	0.431
Anomalous	0.374	0.364	0.498	0.537	0.336	0.345	0.374	0.413	0.359	0.369	0.400	0.441
Dominant	0.363	0.45	0.488	0.546	0.316	0.374	0.441	0.451	0.338	0.400	0.471	0.482
Graph Transformer	0.392	0.497	0.555	0.584	0.345	0.479	0.518	0.566	0.369	0.512	0.554	0.605
ADNET	**0.535**	**0.678**	**0.755**	**0.817**	**0.469**	**0.651**	**0.721**	**0.867**	**0.501**	**0.696**	**0.771**	**0.927**
Recall@N												
Botometer	0.005	0.011	0.016	0.020	0.005	0.012	0.017	0.021	0.005	0.013	0.019	0.022
Alhosseini	0.055	0.115	0.205	0.256	0.060	0.122	0.219	0.273	0.064	0.131	0.234	0.292
SATAR	0.061	0.115	0.205	0.248	0.064	0.122	0.216	0.265	0.069	0.130	0.297	0.283
BotRGCN	0.062	0.119	0.219	0.302	0.067	0.127	0.234	0.323	0.072	0.135	0.251	0.345
Radar	0.005	0.012	0.015	0.018	0.005	0.011	0.016	0.019	0.005	0.011	0.017	0.02
Anomalous	0.047	0.098	0.174	0.217	0.051	0.104	0.186	0.232	0.054	0.111	0.199	0.248
Dominant	0.051	0.096	0.171	0.207	0.054	0.102	0.183	0.221	0.057	0.109	0.195	0.236
Graph Transformer	0.052	0.099	0.181	0.25	0.056	0.105	0.194	0.267	0.059	0.112	0.207	0.285
ADNET	**0.072**	**0.135**	**0.248**	**0.34**	**0.076**	**0.243**	**0.463**	**0.662**	**0.081**	**0.26**	**0.495**	**0.708**

Experiment Results

(b) Benchmark Datasets (Citation Netwroks)

	CiteSeer				ACM				Pubmed			
N	50	100	200	300	50	100	200	300	50	100	200	300
Precision@N												
Radar	0.174	0.171	0.209	0.285	0.226	0.257	0.362	0.400	0.035	0.043	0.057	0.057
Anomalous	0.396	0.524	0.638	0.627	0.480	0.605	0.667	0.705	0.412	0.498	0.555	0.535
Dominant	0.397	0.490	0.618	0.609	0.486	0.619	0.676	0.752	0.392	0.487	0.544	0.572
Graph Transformer	0.447	0.561	0.675	0.722	0.565	0.652	0.695	0.783	0.474	0.515	0.563	0.601
ADNET	**0.616**	**0.774**	**0.832**	**0.920**	**0.777**	**0.897**	**0.957**	**0.962**	**0.649**	**0.705**	**0.771**	**0.810**
Recall@N												
Radar	0.048	0.069	0.114	0.174	0.045	0.080	0.114	0.151	0.005	0.010	0.014	0.017
Anomalous	0.105	0.212	0.349	0.396	0.078	0.149	0.269	0.321	0.045	0.093	0.165	0.206
Dominant	0.102	0.206	0.326	0.397	0.083	0.151	0.275	0.324	0.048	0.091	0.162	0.196
Graph Transformer	0.121	0.225	0.374	0.447	0.080	0.154	0.290	0.378	0.050	0.094	0.172	0.237
ADNET	**0.167**	**0.312**	**0.416**	**0.517**	**0.110**	**0.313**	**0.499**	**0.519**	**0.068**	**0.378**	**0.536**	**0.643**

Conclusion

- We proposed ADNET, which uses active learning for anomaly detection in Twitter-attributed networks.
- Our topology-based active learning framework uses a deep autoencoder to train the model and is able to handle large graphs better than previous methods.
- Our experimental results demonstrate that the proposed approach outperforms state-of-the-art methods in detecting anomalous bot accounts and reduces the annotation cost in Twitter-attributed networks.

Thank you

Contact Information:
Lulwah AlKulaib
lalkulaib@vt.edu