

Density-aware Walks for Coordinated Campaign Detection

Atul Anand Gopalakrishnan¹, Jakir Hossain¹, Tuğrulcan Elmas², and Ahmet Erdem Sarıyüce¹(✉)

¹ University at Buffalo {atulanan,mh267,erdem}@buffalo.edu

² University of Edinburgh telmas@ed.ac.uk

Abstract. Coordinated campaigns frequently exploit social media platforms by artificially amplifying topics, making inauthentic trends appear organic, and misleading users into engagement. Distinguishing these coordinated efforts from genuine public discourse remains a significant challenge due to the sophisticated nature of such attacks. Our work focuses on detecting coordinated campaigns by modeling the problem as a graph classification task. We leverage the recently introduced Large Engagement Networks (LEN) dataset, which contains over 300 networks capturing engagement patterns from both fake and authentic trends on Twitter prior to the 2023 Turkish elections. The graphs in LEN were constructed by collecting interactions related to campaigns that stemmed from ephemeral astroturfing. Established graph neural networks (GNNs) struggle to accurately classify campaign graphs, highlighting the challenges posed by LEN due to the large size of its networks. To address this, we introduce a new graph classification method that leverages the density of local network structures. We propose a random weighted walk (RWW) approach in which node transitions are biased by local density measures such as degree, core number, or truss number. These RWWs are encoded using the Skip-gram model, producing density-aware structural embeddings for the nodes. Training message-passing neural networks (MPNNs) on these density-aware embeddings yields superior results compared to the simpler node features available in the dataset, with nearly a 12% and 5% improvement in accuracy for binary and multiclass classification, respectively. Our findings demonstrate that incorporating density-aware structural encoding with MPNNs provides a robust framework for identifying coordinated inauthentic behavior on social media networks such as Twitter.

Keywords: Random weighted walks · Coordinated campaigns · Graph density

1 Introduction

Social media platforms like Twitter (now X) provide a space for people to express their opinions and stay informed about trending topics. However, like other social media platforms, Twitter is vulnerable to manipulation by malicious actors. These actors often engage in coordinated attacks that artificially amplify

trends using fake accounts and bots. They can operate in a synchronized manner while concealing their identities, misleading users, journalists, and policymakers about what is genuinely trending. Such tactics also coerce users into engaging with fabricated trends, making it increasingly difficult to distinguish between organic trends and those driven by manipulation. Prior research has shown that coordinated campaigns are prevalent in several countries, including Turkey, Pakistan, and India [10, 20, 21].

Gopalakrishnan et al. [16] recently introduced a new graph classification dataset, LEN, consisting of engagement networks including some coordinated campaigns within Turkey’s Twitter sphere during the 2023 Turkish elections. To identify ground-truth campaign graphs, they focus on ephemeral astroturfing, a tactic where a coordinated network of bots rapidly generates a large volume of tweets to manipulate Twitter’s trending list, only to delete them shortly afterward. In each engagement graph, nodes represent users, while edges represent user interactions in the form of retweets, quotes, or replies.

The problem of coordinated campaign detection can be considered as a graph classification task, making it well-suited for message-passing neural networks (MPNNs) [18, 22, 36, 40]. However, Gopalakrishnan et al.’s analysis using established MPNNs highlights the challenges posed by LEN due to its large network sizes. MPNNs are often designed for domains with significantly smaller graphs, such as molecular structures. In contrast, LEN contains approximately ten times more edges, on average, than typical datasets of graphs, such as ogbn-ppa, one of the largest biological graph datasets.

Present work. In this paper, we exploit the fact that campaign-related engagement graphs tend to be denser. We aim to accurately identify coordinated campaigns using our method, called **D**ensity-aware walks for **C**oordinated campaign **D**etection (**DECODE**). We incorporate network density into node embeddings by leveraging node-level density properties, such as degree, core number, and truss number, using random weighted walks (RWWs). For the RWW, we sample a new node using the current node’s density, ensuring that each node maintains a similar local density throughout the walk. These RWWs are converted to density-aware embeddings using Skipgram [26]. We then train a message-passing neural network (MPNN) using these embeddings as input features, enabling the model to leverage density awareness for improved classification. Figure 1 provides a descriptive diagram of our framework. The key contributions of our work can be summarized as follows:

- We leverage multiple density measures, namely degree, core numbers and truss numbers, to distinguish campaign and non-campaign networks based on local density.
- We introduce DECODE, which uses RWWs to encode each node such that its embedding closely resembles those of neighboring nodes with similar densities.
- We train MPNNs on the LEN dataset using the density-aware embeddings to identify campaigns and their subtypes. To evaluate their effectiveness, we compare our models with the baselines from [16].

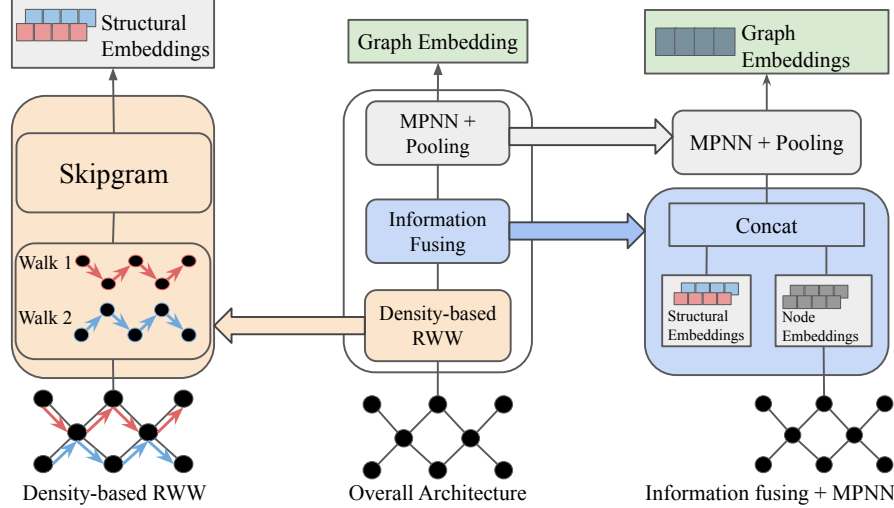


Fig. 1: An overview of DECODE. Density-based random weighted walk captures the local densities around a node. This representation is then concatenated with the input node feature available in the dataset. The concatenated embedding is encoded using an MPNN and subsequently aggregated to form the graph embedding, which is used for downstream classification.

The rest of the paper is structured as follows. In Section 2, we review related work, while Section 3 introduces the dataset and essential terminologies. In Section 4, we present our methodology, including the random weighted walk algorithm and its density-awareness encoding using degrees, core numbers, and truss numbers. We verify and discuss the performance improvements using the density-aware embeddings to demonstrate their importance in Section 5. Finally, we conclude by summarizing our findings and addressing potential limitations and future directions in Section 6. The code for DECODE is available at <https://github.com/erdemUB/ECMLPKDD25>.

2 Related Work

In this section we discuss related work in the domain of structural and positional encoding and its relevance in MPNNs and graph transformers (Section 2.1), and coordinated campaigns on social media platforms (Section 2.2).

2.1 Structural and Positional Encoding

Structural encoding is the process of ensuring that nodes with similar structural roles in a graph have similar embeddings. Positional encoding captures the proximity between two nodes in a graph. Both of these encodings can be

obtained using random weighted walks. In Deepwalk, random walks are generated for each node [30]. These walks are converted to node embeddings using Skipgram [26]. Node2Vec biases the random walks to preserve both local (BFS) and global structure (DFS) [17]. Struct2Vec constructs similarity graphs using time-warping as the similarity function [33]. Random-walks are performed on the similarity graphs to generate node embeddings. Modularized non-negative matrix factorization (M-NMF) preserves community structure by using a community embedding matrix and community modularity score in addition to random walk embeddings [38]. Random walk features have also been used to improve structural awareness of MPNNs [43,44] and graph transformers [6,7,32]. Other commonly used structural and positional encoding include heat kernel [13,23], subgraphs [3,4], shortest distance [24], and node degree centralities [41]. In this work, we devise a new random-walk that captures the local density around nodes.

2.2 Coordinated Campaigns on Social Media

The process by which users on a social media platform coordinate in large groups to engage in malicious behavior is known as a coordinated campaign (also known as influence operations) [29]. These coordinated campaigns are often designed to mislead users by disseminating misinformation or by propagating falsified ideologies. Examples of coordinated campaigns include, using advertisements and influencers to dominate trends [27], deploying bots to boost user popularity [8,9], and state-sponsored influence operations, such as Russia’s interference in the 2016 US elections [42] and alleged coordinated attacks by the Chinese Communist Party to sway public opinion [19]. Our work leverages coordinated campaigns driven by ephemeral astroturfing, where bots flood Twitter with random tweets to bypass filters and then delete them immediately [10,16]. Since Twitter updates trends in windows, deleted tweets remain unaccounted for until the next update, allowing adversaries to exploit the illusion of organic engagement.

Over the years, methods have been developed to counter coordinated efforts on Twitter. These include techniques such as tweet and hashtag similarity [25,28], temporal methods focusing on tweet frequency [28,37], shared URLs and articles [15], and detecting other coordination signals [12,31,39]. Recent approaches have explored centrality-based node pruning on similarity networks [25], and graph neural networks [11,15] for detecting these attacks. Our work attempts to identify coordinated campaigns by modeling it as a graph classification problem. Additionally, we encode density-based properties using RWWs. Incorporating density-aware embeddings into MPNN training leads to improved performance in classification compared to using only the raw node features from the dataset.

3 Preliminaries

In this section, we describe our dataset and the key terminologies required for this work. We first provide details on the LEN dataset in Section 3.1. Then we

Table 1: Statistics of the engagement networks for LEN, containing 314 networks.

	Sub-types	# G	# nodes			# edges			Explanation
			Max	Min	Avg	Max	Min	Avg	
Campaign	Politics	62	50,286	100	6,570	71,704	203	10,210	Political content including slogans, and misinformation camps.
	Reform	58	19,578	131	1,229	1,105,918	540	25,268	Organized movements advocating political changes.
	News	24	54,996	581	10,368	80,784	942	15,582	News amplified through bot and troll activity.
	Finance	14	9,976	273	1,802	10,725	243	2,334	Financial promotions, primarily cryptocurrency-related.
	Noise	9	55,933	454	12,180	48,937	473	10,882	Content that does not fit into any specific category.
	Cult	6	7,880	313	2,303	11,615	637	3,431	Slogans from a cult leveraging bots.
	Entertainment	3	4,220	678	2,237	132,013	3,806	48,767	Celebrities using bots for self-promotion.
	Common	3	9,974	3,487	5,919	9,470	2,818	7,066	Frequently used phrases forming trends organically
	Overall	179	55,933	100	5,157	1,105,918	203	16,006	
Non-Campaign	News	52	95,575	818	24,834	213,444	709	43,201	Coverage of news not sourced from Twitter.
	Sports	30	75,653	469	9,530	101,656	403	12,948	Discussions around major sports events.
	Festival	17	119,952	885	35,466	199,305	803	55,947	Trends related to holidays, festivals, and special occasions.
	Internal	11	87,720	4,188	33,061	196,103	4,374	54,442	Events primarily generated within Twitter's ecosystem.
	Common	10	64,320	1,214	17,079	99,306	1,270	24,869	Frequently used phrases forming trends organically.
	Entertainment	8	20,060	1,477	7,289	45,211	1,712	12,578	Engagement with popular TV shows and online videos.
	Announ. cam.	4	26,358	6,650	13,382	50,864	14,362	24,817	Officially launched political campaigns.
	Sports cam.	3	4,661	2,880	3,654	7,367	4,451	5,534	Hashtags initiated by professional sports teams.
	Overall	135	119,952	469	20,632	213,444	403	33,765	

give a brief overview of the notation, the density metrics used (degree, k -core, k -truss) and message passing neural networks in Section 3.2.

3.1 Large Engagement Networks (LEN) Dataset

Large Engagement Networks (LEN) is graph dataset that contains coordinated campaigns related to Turkish Twitter. It focuses on the 2023 elections in Turkey when this issue was prevalent. The campaign graphs in the dataset are an outcome of ephemeral astroturfing. The dataset comprises of 314 engagement networks, where each network is associated to a trend. There are 179 campaign graphs and 135 non-campaign graphs. These graphs are further divided into sub-types such as politics, news, finance and more, as shown in Table 1. The nodes represent users and edges represents engagements between the users. A directed edge from node X to Y , signifies that X engaged with (retweeted, replied to, or quoted) Y . The graphs also consist of node and edge features. The features used for the node attributes include user description (bio), follower count, following count, user's total tweet count, and user's verification status. The edge attributes include the type of engagement (retweet, reply, or quote), engagement count (e.g., number of retweets), impression count, text, number of likes, whether the tweet is labeled as sensitive or not, and the timestamp of the tweet. Gopalakrishnan et al. provides three benchmarks for the LEN dataset: (1) binary classification to classify the networks into campaign and non-campaign networks; (2) multi-class classification to categorize campaigns into one of the 7 sub-types as shown in Table 1; and (3) binary classification of news networks into campaign and non-campaign. We use the LEN dataset as it is the only ground-truth graph classification dataset that identifies if a trend's popularity is driven by coordinated campaigns.

3.2 Notation, Density Metrics, and MPNNs

A graph is a collection of vertices and edges. It is depicted as $G = (V, E)$, where V is the number of vertices and E is the number of edges. A graph can also be represented as $G = (A, X, y)$, where A is the adjacency matrix, X is the feature matrix for the nodes, and y is the graph’s label.

Degree, k -core and k -truss: The degree of a node is the number of edges connected to it, providing a simple measure of its local connectivity [14]. A k -core is a subgraph in which every node has at least k connections within the subgraph [35]. The core number of a node represents the largest k -core to which it belongs. Computing core numbers of all the nodes in a graph has a linear cost, $O(|E|)$. Similarly, a k -truss is defined as a subgraph where each edge is part of at least $k - 2$ triangles within the subgraph [5]. The truss number of an edge indicates the highest k -truss to which it belongs. Computing truss numbers is a bit costly, $O(|E|^{1.5})$, but is still polynomial and practical for large networks. Since truss numbers are edge-based, we compute a node’s truss number by averaging the truss numbers associated to the connected edges. Degree, core number, and truss number all measure graph density, with degree indicating direct connections, and core and truss numbers reflecting a node’s role in dense substructures.

We use three local density measures for a node: (1) degree of the node, (2) core number of the node, and (3) average truss numbers of all edges incident to the node (we simply refer them as degree, core, and truss number of a node in the rest of the paper). Note that we ignore the edge directions in the engagement networks, hence we use the original definitions of k -core and k -truss for undirected graphs.

Message Passing Neural Networks: MPNNs consist of two steps, aggregate and update, as shown in Equation 1, where $\mathcal{N}(v)$ is used to represent the neighborhood of node v .

$$h_v^{(l+1)} = \text{UPDATE} \left(h_v^{(l)}, \text{AGGREGATE} \left(\{h_u^{(l)} \mid u \in \mathcal{N}(v)\} \right) \right) \quad (1)$$

In the aggregate step, each node gathers information from its neighbors. This typically involves summing, averaging, or applying more complex functions (e.g., attention mechanisms) to the neighbors’ feature vectors. In the update step, the aggregated information is combined with the node’s own features to update its representation. To do so, a neural network (example, an MLP) or a simple transformation (example, a weighted sum) is applied. Therefore, the nodes refine their representations based on the information received. MPNNs generally differ in the aggregation strategy used. GCN uses dual-degree normalization to account for the varying number of neighbors each node may have [22]. GAT uses attention weight to assign varying weights to each neighbor [36]. GIN uses an MLP to perform aggregation using a trainable parameter (ϵ) to determine the

Table 2: Descriptive statistics to indicate the average degree, core number and truss numbers of the nodes across different sub-types spanning campaign and non-campaign graphs. Overall, campaign graphs exhibit higher local densities than the non-campaign ones.

	Sub-types	Mean Degree	Mean Core number	Mean Truss number
Campaign	Politics	3.108 ± 1.483	1.635 ± 0.872	0.384 ± 1.595
	Reform	16.258 ± 12.118	9.067 ± 7.067	17.104 ± 10.545
	News	3.015 ± 0.990	1.598 ± 0.481	0.172 ± 0.365
	Finance	2.590 ± 1.512	1.414 ± 0.823	0.287 ± 0.977
	Noise	2.080 ± 0.790	1.214 ± 0.342	0.261 ± 0.370
	Cult	2.981 ± 0.690	1.580 ± 0.409	0.569 ± 0.919
	Entertainment	11.477 ± 0.264	6.180 ± 0.171	2.311 ± 1.920
	Common	2.391 ± 1.586	1.380 ± 0.793	0.572 ± 0.751
	Overall	3.941 ± 10.459	2.120 ± 6.083	5.057 ± 7.933
Non-Campaign	News	3.443 ± 0.845	1.778 ± 0.412	0.179 ± 0.106
	Sports	2.718 ± 0.581	1.428 ± 0.242	0.155 ± 0.150
	Festival	2.867 ± 0.501	1.526 ± 0.239	0.245 ± 0.157
	Internal	3.293 ± 0.901	1.705 ± 0.438	0.184 ± 0.110
	Common	2.913 ± 0.650	1.529 ± 0.305	0.109 ± 0.068
	Entertainment	3.453 ± 0.941	1.816 ± 0.471	0.651 ± 0.491
	Announced Camp.	3.711 ± 1.135	1.925 ± 0.588	1.309 ± 0.970
	Sports Camp.	3.029 ± 0.191	1.590 ± 0.100	0.037 ± 0.012
	Overall	3.210 ± 0.81	1.672 ± 0.387	0.219 ± 0.261

amount of importance given to the ego node, as compared to its neighbors [40]. GraphSAGE is an inductive graph representational learning model that has the ability to generalize to unseen nodes, unlike transductive models [18]. This is done by learning a message-passing model on a sampled set of nodes in the given graph.

4 Methodology

We propose DECODE, a random weighted walk (RWW) approach for learning density-aware node embeddings. Here, node densities are used to determine transition probabilities in the RWWs. This emphasis on density is because campaign graphs in LEN are denser than non-campaign graphs. Specifically, we use degree, core number, and truss number as density metrics due to their widespread use and computational efficiency [2, 34]. Table 2 provides detailed statistics showcasing the density metrics across campaign and non-campaign graphs. Notably, the densest campaign graphs belonged to the reform sub-type, which constitutes a large portion of the dataset, as shown in Table 1.

Algorithm 1 provides a formal overview of DECODE. In our algorithm, ϕ represents the density function, where $\phi(v)$ returns the normalized density of a given node v . The function ϕ is defined based on the chosen density metric for

Algorithm 1 Density-aware random weighted walk (DECODE)

Input: Graph $G = (V, E)$, walk length L , density func. $\phi : V \rightarrow [0, 1]$, threshold τ
Output: List of walks W
Initialize $W \leftarrow []$
for each node $v \in V$ **do**
 Initialize walk $w \leftarrow [v]$
 for $t = 1$ to L **do**
 $v_t = \text{Top}(w)$
 Let $N(v_t) \leftarrow \{u \in V \mid (v_t, u) \in E\}$
 if $N(v_t) \neq \emptyset$ **then**
 if $\phi(v_t) > \tau$ **then**
 Set $w_u = \phi(u)$ for all $u \in N(v_t)$
 else
 Set $w_u = 1 - \phi(u)$ for all $u \in N(v_t)$
 end if
 Sample $v_{t+1} \sim P(u) = w_u / \sum_{u' \in N(v_t)} w_{u'}$
 Push v_{t+1} to w
 end if
 end for
 Append w to W
end for
return W

RWWs. It can be set to return the degree, core number, or truss number of a node.

Additionally, we introduce τ , a scalar threshold parameter that differentiates between high and low-density nodes in RWWs. The threshold is set to one of the following values: 0.5, the median node density in the graph, or the midpoint of node densities, as detailed in Section 5.1. The steps for collecting RWWs in our algorithm are as follows:

1. At each step of the RWW, the next node is selected based on the density of the current node.
2. If the current node’s density exceeds the threshold τ , transitions to higher-density neighbors are preferred, with sampling weights defined as $(w_u = \phi(u))$, where w_u represents the weight assigned to node u
3. Conversely, if the current node’s density is below τ , transitions to lower-density neighbors are favored by inverting the sampling weights ($w_u = 1 - \phi(u)$).
4. The transition probabilities for the neighbors are obtained by normalizing the sampling weights and new nodes are sampled using them at each step.
5. Once we obtain the RWWs, we use Skipgram to encode them, following prior methods [17, 30].

These density-aware embeddings and node feature are fed into the MPNNs for downstream classification. The MPNNs used in this paper include GCN,

GAT, GIN, and GraphSAGE. In the following section we discuss the experimental setup used in this paper and discuss our results for binary and multiclass classification by comparing our method to the results provided in [16].

5 Experimental Evaluation

We evaluate the performance of DECODE on the LEN dataset using two tasks: (i) campaign vs. non-campaign classification in engagement networks (binary classification) and (ii) campaign sub-type classification, where the sub-types are provided in Table 1 (multi-class classification). Section 5.1 details the experimental setup. Sections 5.2 and 5.3 present the experimental results for binary and multi-class classification, respectively.

5.1 Experimental Setup

We run our model on two input configurations: (i) density-aware embeddings and (ii) a concatenation of density-aware embeddings with the input node features available in the dataset. We consider each of the three density-based features in our random walks—degrees, core numbers, and truss numbers—and provide comparisons. To contextualize the empirical results of DECODE, we compare our method against four baselines: GCN, GAT, GIN, and GraphSAGE. These models are trained solely on the input node features available in the dataset. This comparison allows us to evaluate the importance of density-aware embeddings over existing node features. To construct the RWW embeddings, we set the walk length to 100. For encoding the nodes using Skipgram, we use a window length of 4, meaning each node is encoded using its four neighboring nodes in the random weighted walks. The walk embedding size is set to 128. We set the threshold parameter (τ) to the following values:

- **0.5**: A fixed value of 0.5.
- **Median**: The median of the list of the density-based features in a graph.
- **Mid-point (abbreviated as mid)**: This value is calculated as the average of the smallest and largest values of the density-based feature under consideration.

For MPNNs, we perform hyperparameter tuning over hidden layer sizes, $h \in \{128, 256, 512, 1024\}$, and learning rates, $l \in \{0.001, 0.0001, 0.00001\}$ as done in [16]. We also use mean pooling to produce graph embeddings.

5.2 Results for Campaign vs Non-campaign Classification

LEN consists of 179 campaign graphs and 135 non-campaign graphs. The results for accuracy and F1-score are presented in Tables 3 and 4, respectively. We observe the following key insights:

Table 3: Accuracy for binary classification. **NF** denotes the MPNN trained with node features and **RWW** denotes the one that used random-weighted walks. Best in each group is in bold. Underlined value denotes the best overall accuracy.

	Input	Degree		Core Numbers		Truss Numbers	
		τ	Accuracy	τ	Accuracy	τ	Accuracy
GCN	NF	-	0.702 ± 0.018	-	0.702 ± 0.018	-	0.702 ± 0.018
	RWW	mid	0.810 ± 0.013	0.5	0.787 ± 0.010	median	0.800 ± 0.019
	NF + RWW	mid	0.784 ± 0.006	median	0.805 ± 0.018	0.5	0.803 ± 0.019
GAT	NF	-	0.735 ± 0.015	-	0.735 ± 0.015	-	0.735 ± 0.015
	RWW	median	0.795 ± 0.015	mid	0.808 ± 0.022	median	0.836 ± 0.016
	NF + RWW	median	0.792 ± 0.022	0.5	0.785 ± 0.010	median	0.792 ± 0.030
GIN	NF	-	0.633 ± 0.065	-	0.633 ± 0.065	-	0.633 ± 0.065
	RWW	0.5	0.756 ± 0.005	median	0.766 ± 0.014	0.5	0.771 ± 0.006
	NF + RWW	0.5	0.751 ± 0.005	mid	0.792 ± 0.014	mid	0.782 ± 0.019
SAGE	NF	-	0.729 ± 0.006	-	0.729 ± 0.006	-	0.729 ± 0.006
	RWW	0.5	0.852 ± 0.010	mid	0.774 ± 0.025	0.5	0.834 ± 0.052
	NF + RWW	median	0.758 ± 0.010	0.5	0.790 ± 0.017	0.5	0.813 ± 0.018

Table 4: F1-score for binary classification. **NF** denotes the MPNN trained with node features and **RWW** denotes the one that used random-weighted walks. Best in each group is in bold. Underlined value denotes the best overall F1-score.

	Inp.	τ	Degree	τ	Core Numbers	τ	Truss Numbers
			F1		F1		F1
GCN	NF	-	0.687 ± 0.021	-	0.687 ± 0.021	-	0.687 ± 0.021
	RWW	mid	0.839 ± 0.004	median	0.814 ± 0.008	median	0.806 ± 0.021
	NF + RWW	median	0.805 ± 0.011	median	0.838 ± 0.017	0.5	0.824 ± 0.017
GAT	NF	-	0.765 ± 0.018	-	0.765 ± 0.018	-	0.765 ± 0.018
	RWW	median	0.825 ± 0.012	mid	0.840 ± 0.015	median	0.853 ± 0.011
	NF + RWW	median	0.820 ± 0.020	0.5	0.824 ± 0.006	median	0.824 ± 0.030
GIN	NF	-	0.710 ± 0.037	-	0.710 ± 0.037	-	0.710 ± 0.037
	RWW	0.5	0.807 ± 0.005	median	0.800 ± 0.013	0.5	0.790 ± 0.003
	NF + RWW	mid	0.782 ± 0.006	mid	0.816 ± 0.012	mid	0.795 ± 0.019
SAGE	NF	-	0.713 ± 0.008	-	0.713 ± 0.008	-	0.713 ± 0.008
	RWW	0.5	0.877 ± 0.010	mid	0.789 ± 0.020	0.5	0.857 ± 0.031
	NF + RWW	median	0.803 ± 0.007	0.5	0.820 ± 0.016	0.5	0.834 ± 0.011

- Pairing GraphSAGE with degree-based RWW achieves the best performance, yielding an accuracy of 0.852 ± 0.010 and an F1-score of 0.877 ± 0.010 , surpassing the best baseline in [16] by 0.117 and 0.112 for accuracy and F1-score, respectively. The value of τ is set to 0.5 in this case.
- RWW features consistently outperform LEN node features, achieving higher accuracy and F1-score in most cases.
- Embeddings learnt from degree-based RWW generally outperforms other density-aware variants, achieving the highest AUROC scores across all models. The only exception is when GCN and GraphSAGE are trained on embed-

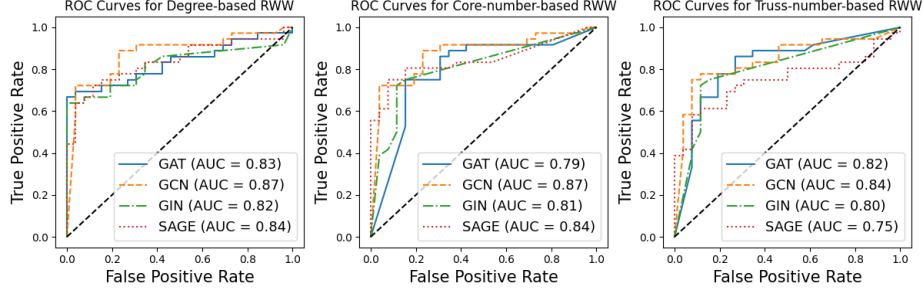


Fig. 2: Receiver Operating Characteristic (ROC) curves for campaign vs. non-campaign classification for degree, core-number and truss-number based random weighted walks. The best performing input format and threshold are taken into consideration for each model.

dings obtained from k -core-based RWW, where the AUC scores are identical. The results are illustrated in Figure 2.

- The best-performing threshold varies depending on the model used. Median serves as the best threshold for GCN and GAT, while 0.5 is optimal for GIN and GraphSAGE.

The above insights suggest that RWW based methods yield improvements in performance for both accuracy and F1-score. Additionally, degree-based RWW generally outperforms core or truss-based RWWs. However the choice of threshold is model-dependent.

5.3 Results of Campaign-type Classification

The goal here is to classify campaign graphs into one of the seven sub-types described in Table 1. Among these, the most common categories are Politics (62 graphs) and Reform (58 graphs). The results for accuracy and macro F1-scores are provided in Tables 5 and 6, respectively.

From these results, the following inferences can be made:

- Pairing GIN with degree-based RWW achieves the best performance, with an accuracy of 0.679 ± 0.001 , surpassing the baseline in [16] by 0.045.
- The model accuracies benefit the most when input node features from the dataset are combined with density-aware embeddings, outperforming all the other setups in a majority of the scenarios.
- The best-performing thresholds are mid for GCN, 0.5 for GAT, and median for GIN and GraphSAGE, yielding the highest accuracy for each model.
- The highest macro-F1 score obtained by our work is 0.338 ± 0.051 (for GIN with truss-based RWW and τ set to 0.5) which is 0.013 less than the best performing baseline provided in [16]. We believe this happens due to label imbalance. Several campaign-type labels (example, finance, entertainment, cult) have very few samples, making them harder to classify.

Table 5: Accuracy results for multiclass classification. **NF** denotes the MPNN trained with node features and **RWW** denotes the one that used random-weighted walks. Best in each group is in bold. Underlined value denotes the best overall accuracy.

	Inp.	Degree		Core Numbers		Truss Numbers	
		τ	Acc.	τ	Acc.	τ	Acc.
GCN	NF	-	0.533 ± 0.041	-	0.533 ± 0.041	-	0.533 ± 0.041
	RWW	median	0.619 ± 0.019	0.5	0.614 ± 0.011	0.5	0.628 ± 0.000
	NF + RWW	0.5	0.647 ± 0.009	mid	0.665 ± 0.011	mid	0.651 ± 0.015
GAT	NF	-	0.567 ± 0.033	-	0.567 ± 0.033	-	0.567 ± 0.033
	RWW	0.5	0.647 ± 0.009	mid	0.670 ± 0.037	mid	0.623 ± 0.017
	NF + RWW	0.5	0.628 ± 0.025	0.5	0.674 ± 0.001	mid	0.633 ± 0.027
GIN	NF	-	0.633 ± 0.067	-	0.633 ± 0.067	-	0.633 ± 0.067
	RWW	mid	0.679 ± 0.001	median	0.647 ± 0.009	mean	0.637 ± 0.024
	NF + RWW	median	0.670 ± 0.009	median	0.656 ± 0.017	median	0.679 ± 0.023
SAGE	NF	-	0.583 ± 0.053	-	0.583 ± 0.053	-	0.583 ± 0.053
	RWW	0.5	0.637 ± 0.01	0.5	0.656 ± 0.027	median	0.660 ± 0.011
	NF + RWW	0.5	0.665 ± 0.011	median	0.665 ± 0.065	median	0.651 ± 0.001

Table 6: Macro F1-score results for multiclass classification. **NF** denotes the MPNN trained with node features and **RWW** denotes the one that used random-weighted walks. Best in each group is in bold. Underlined values denote the best overall Macro-F1 score.

	Inp.	Degree		Core Numbers		Truss Numbers	
		τ	Macro-F1	τ	Macro-F1	τ	Macro-F1
GCN	NF	-	0.251 ± 0.022	-	0.251 ± 0.022	-	0.251 ± 0.022
	RWW	median	0.249 ± 0.009	0.5	0.247 ± 0.018	0.5	0.250 ± 0.001
	NF + RWW	mid	0.249 ± 0.004	mid	0.260 ± 0.007	mid	0.259 ± 0.006
GAT	NF	-	0.264 ± 0.014	-	0.264 ± 0.014	-	0.264 ± 0.014
	RWW	0.5	0.255 ± 0.003	mid	0.298 ± 0.024	median	0.233 ± 0.006
	NF + RWW	mean	0.227 ± 0.01	mean	0.255 ± 0.028	median	0.247 ± 0.015
GIN	NF	-	0.351 ± 0.09	-	0.351 ± 0.09	-	0.351 ± 0.09
	RWW	0.5	0.317 ± 0.053	mean	0.260 ± 0.003	0.5	0.272 ± 0.030
	NF + RWW	median	0.305 ± 0.03	0.5	0.280 ± 0.028	mid	0.338 ± 0.051
SAGE	NF	-	0.320 ± 0.061	-	0.320 ± 0.061	-	0.320 ± 0.061
	RWW	0.5	0.266 ± 0.03	median	0.282 ± 0.025	mid	0.274 ± 0.019
	NF + RWW	0.5	0.295 ± 0.011	median	0.267 ± 0.026	median	0.254 ± 0.003

- We also provide confusion matrices for the models across various RWW methods in Figure 3, where we display the confusion matrix for the best-performing configuration of each model-RWW pair. From this, we again observe that models struggle to accurately classify labels with fewer graphs.

The insights above suggest that RWW-based methods improve performance in terms of accuracy. Additionally, we find that degree is an effective parameter for random weighted walks, and the median is a suitable threshold. However,

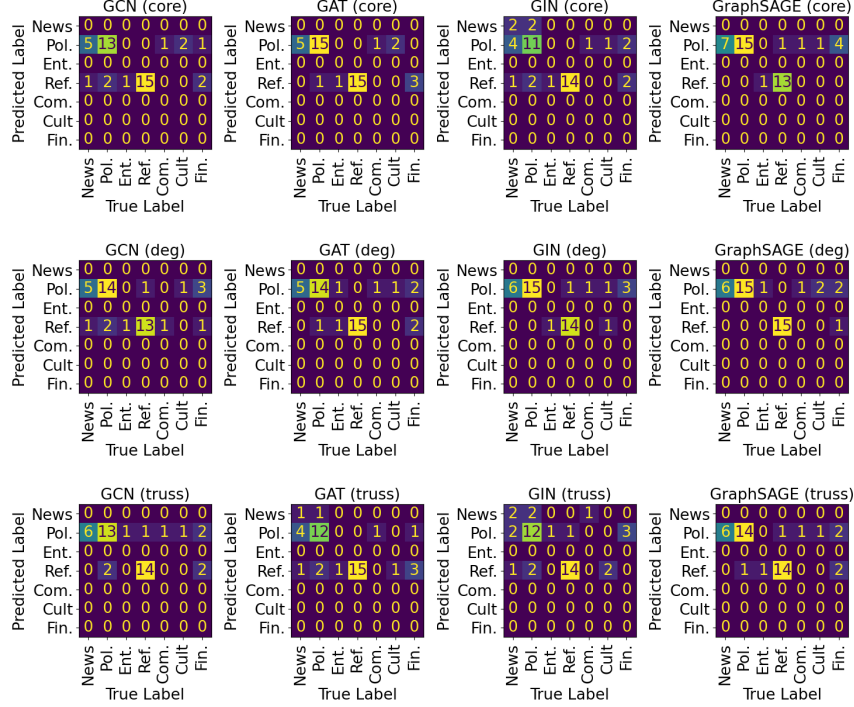


Fig. 3: Confusion matrices for multi-class classification. The best performing configuration is considered for each model and density pair is displayed here.

we observe a drop in F1-scores, likely due to the models' difficulty in classifying graphs associated with labels that have fewer samples.

6 Conclusion

We propose DECODE, a density-based random weighted walk (RWW) approach that leverages local density metrics such as degree, core number, and truss number to detect coordinated campaigns in engagement networks. We prioritize density over other structural properties, as campaign graphs are consistently denser than non-campaign graphs, exhibiting higher mean degree, core number, and truss number. DECODE learns density-aware embeddings using RWWs, where node transitions are guided by local density, ensuring that neighboring nodes have similar density characteristics. These RWWs are then converted into density-aware embeddings using Skipgram. We train an MPNN using these embeddings on the LEN dataset and observe performance improvements, surpassing the accuracy of [16] by 11% and 4.5% in binary and multiclass classification, re-

spectively. Additionally, we outperform their F1-score for binary classification by 0.112. However, our highest macro-F1 score for campaign type classification is 0.013 lower than the best-performing baseline from Gopalakrishnan et al. This is due to the label disparity issues in the campaign classification problem. For future work, we aim to explore alternative RWW methods, such as nearest-neighbor RWW, instead of thresholding approaches. Additionally, we plan to incorporate other structural properties, such as betweenness centrality and clustering coefficient, to further refine the RWW process.

Acknowledgments. A. A. Gopalakrishnan, J. Hossain, and A. E. Sariyuce are supported by NSF awards OAC-2107089 and IIS-2236789, and this research used resources from the Center for Computational Research at the University at Buffalo (CCR 2025) [1].

References

1. Center for computational research, university at buffalo (2025), <http://hdl.handle.net/10477/79221>. Accessed: 2025-04-14
2. Batagelj, V., Zaversnik, M.: An $o(m)$ algorithm for cores decomposition of networks. arXiv preprint cs/0310049 (2003)
3. Bouritsas, G., Frasca, F., Zafeiriou, S., Bronstein, M.M.: Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(1), 657–668 (2022)
4. Chen, D., O’Bray, L., Borgwardt, K.: Structure-aware transformer for graph representation learning. In: International conference on machine learning. pp. 3469–3489. PMLR (2022)
5. Cohen, J.: Trusses: Cohesive subgraphs for social network analysis. National security agency technical report **16**(3.1), 1–29 (2008)
6. Dwivedi, V.P., Bresson, X.: A generalization of transformer networks to graphs. arXiv preprint arXiv:2012.09699 (2020)
7. Dwivedi, V.P., Rampásek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A.T., Beaini, D.: Long range graph benchmark. *Advances in Neural Information Processing Systems* **35**, 22326–22340 (2022)
8. Elmas, T.: Analyzing activity and suspension patterns of twitter bots attacking turkish twitter trends by a longitudinal dataset. In: Companion Proceedings of the ACM Web Conference 2023. pp. 1404–1412 (2023)
9. Elmas, T., Overdorf, R., Aberer, K.: Characterizing retweet bots: The case of black market accounts. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 16, pp. 171–182 (2022)
10. Elmas, T., Overdorf, R., Özkalay, A.F., Aberer, K.: Ephemeral astroturfing attacks: The case of fake twitter trends. In: 2021 IEEE European symposium on security and privacy (EuroS&P). pp. 403–422. IEEE (2021)
11. Elmas, T., Randl, M., Attia, Y.: # teamfollowback: Detection & analysis of follow back accounts on social media. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 18, pp. 381–393 (2024)
12. Erhardt, K., Pentland, A.: Hidden messages: mapping nations’ media campaigns. *Computational and Mathematical Organization Theory* **30**(2), 161–172 (2024)

13. Feldman, O., Boyarski, A., Feldman, S., Kogan, D., Mendelson, A., Baskin, C.: Weisfeiler and leman go infinite: Spectral and combinatorial pre-colorings. *Transactions on Machine Learning Research*
14. Freeman, L.C.: Centrality in social networks conceptual clarification. *Social Networks* **1**(3), 215–239 (1978)
15. Gabriel, N.A., Broniatowski, D.A., Johnson, N.F.: Inductive detection of influence operations via graph learning. *Scientific Reports* **13**(1), 22571 (2023)
16. Gopalakrishnan, A.A., Hossain, J., Elmas, T., Sariyuce, A.E.: Large engagement networks for classifying coordinated campaigns and organic twitter trends (2025), <https://api.semanticscholar.org/CorpusID:276742534>
17. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 855–864 (2016)
18. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. *Advances in neural information processing systems* **30** (2017)
19. Jacobs, C.S., Carley, K.M.: # whatisdemocracy: finding key actors in a chinese influence campaign. *Computational and Mathematical Organization Theory* **30**(2), 127–147 (2024)
20. Jakesch, M., Garimella, K., Eckles, D., Naaman, M.: Trend alert: A cross-platform organization manipulated twitter trends in the indian general election. *Proceedings of the ACM on Human-computer Interaction* **5**(CSCW2), 1–19 (2021)
21. Kausar, S., Tahir, B., Mehmood, M.A.: Towards understanding trends manipulation in pakistan twitter. *arXiv preprint arXiv:2109.14872* (2021)
22. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations* (2017)
23. Kreuzer, D., Beaini, D., Hamilton, W., Létourneau, V., Tossou, P.: Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems* **34**, 21618–21629 (2021)
24. Li, P., Wang, Y., Wang, H., Leskovec, J.: Distance encoding: Design provably more powerful neural networks for graph representation learning. *Advances in Neural Information Processing Systems* **33**, 4465–4478 (2020)
25. Luceri, L., Pantè, V., Burghardt, K., Ferrara, E.: Unmasking the web of deceit: Uncovering coordinated activity to expose information operations on twitter. In: *Proceedings of the ACM Web Conference 2024*. pp. 2530–2541 (2024)
26. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
27. Ong, J.C., Cabañes, J.V.: Architects of networked disinformation: Behind the scenes of troll accounts and fake news production in the philippines (2018)
28. Pacheco, D., Flammini, A., Menczer, F.: Unveiling coordinated groups behind white helmets disinformation. In: *Companion proceedings of the web conference 2020*. pp. 611–616 (2020)
29. Pamment, J., Smith, V.: Attributing information influence operations: Identifying those responsible for malicious behaviour online. *NATO Strategic Communication Centre of Excellence* (2022)
30. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 701–710 (2014)
31. Pote, M., Elmas, T., Flammini, A., Menczer, F.: Coordinated reply attacks in influence operations: Characterization and detection. In: *Proceedings of the International AAAI Conference on Web and Social Media*. vol. 19, pp. 1586–1598 (2025)

32. Rampášek, L., Galkin, M., Dwivedi, V.P., Luu, A.T., Wolf, G., Beaini, D.: Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems* **35**, 14501–14515 (2022)
33. Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R.: struc2vec: Learning node representations from structural identity. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. pp. 385–394 (2017)
34. Sariyüce, A.E., Seshadhri, C., Pinar, A., Çatalyürek, Ü.V.: Nucleus decompositions for identifying hierarchy of dense subgraphs. *ACM Transactions on the Web (TWEB)* **11**(3), 1–27 (2017)
35. Seidman, S.B.: Network structure and minimum degree. *Social networks* **5**(3), 269–287 (1983)
36. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: *International Conference on Learning Representations* (2018)
37. Vishnuprasad, P.S., Nogara, G., Cardoso, F., Cresci, S., Giordano, S., Luceri, L.: Tracking fringe and coordinated activity on twitter leading up to the us capitol attack. In: *Proceedings of the international AAAI conference on web and social media*. vol. 18, pp. 1557–1570 (2024)
38. Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., Yang, S.: Community preserving network embedding. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 31 (2017)
39. Weber, D., Neumann, F.: Amplifying influence through coordinated behaviour in social networks. *Social Network Analysis and Mining* **11**(1), 111 (2021)
40. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: *International Conference on Learning Representations*
41. Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., Liu, T.Y.: Do transformers really perform badly for graph representation? *Advances in neural information processing systems* **34**, 28877–28888 (2021)
42. Zannettou, S., Caulfield, T., De Cristofaro, E., Sirivianos, M., Stringhini, G., Blackburn, J.: Disinformation warfare: Understanding state-sponsored trolls on twitter and their influence on the web. In: *Companion proceedings of the 2019 world wide web conference*. pp. 218–226 (2019)
43. Zeng, D., Chen, W., Liu, W., Zhou, L., Qu, H.: Rethinking random walk in graph representation learning. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 1–5. IEEE (2023)
44. Zhou, C., Wang, X., Zhang, M.: Facilitating graph neural networks with random walk on simplicial complexes. *Advances in Neural Information Processing Systems* **36**, 16172–16206 (2023)