scientific reports

Check for updates

OPEN

Inductive detection of influence operations via graph learning

Nicholas A. Gabriel[1]✉, David A. Broniatowski[2] & Neil F. Johnson[1]

Influence operations are large-scale efforts to manipulate public opinion. The rapid detection and disruption of these operations is critical for healthy public discourse. Emergent AI technologies may enable novel operations that evade detection and influence public discourse on social media with greater scale, reach, and specificity. New methods of detection with inductive learning capacity will be needed to identify novel operations before they indelibly alter public opinion and events. To this end, we develop an inductive learning framework that: (1) determines content- and graph-based indicators that are not specific to any operation; (2) uses graph learning to encode abstract signatures of coordinated manipulation; and (3) evaluates generalization capacity by training and testing models across operations originating from Russia, China, and Iran. We find that this framework enables strong cross-operation generalization while also revealing salient indicators-illustrating a generic approach which directly complements transductive methodologies, thereby enhancing detection coverage.

Manipulation of public opinion by state-backed entities is an ongoing concern. Several influence operations (IO) campaigns intended to shape geopolitical discourse have been identified on various platforms-and particularly on social media[1–12]. For example, IO campaigns designed to promote fake news, advance nationalistic narratives, and exacerbate political tensions have been detected across social media platforms including Twitter[1,5,11,12], Facebook[6–10,13], Reddit[2,14], and Gab[3,15], among others. Identifying and disrupting such campaigns is an ongoing challenge, in large part because positive attribution of foreign influence is time consuming and does not easily scale within or across platforms. Additionally, the rapid development and adoption of generative AI may enable IO to automate behaviours previously achievable only by human actors, disguising activity and enabling novel strategies which have greater efficacy, scale, reach, and specificity. Most methods of detecting IO to this point have relied on identifying and indexing specific indicators of previous campaigns, making these methods inherently transductive. While such methods will continue to play an important role in detecting and constraining IO activity, identifying increasingly novel and sophisticated IO campaigns will require inductive methods which can generalize from previous observations. We present an inductive learning framework, depicted in Fig. 5, that addresses this challenge by combining data censorship, graph learning, and feature attribution to identify models and indicators that can generalize across operations and across time.

Previous work in detecting influence operations using machine learning has successfully identified a variety of IO campaigns and activity. Broadly speaking, there have been two main approaches: *content-based* and *graph-based*. Some examples of content-based approaches include: Smith et al.[16], who used narratives derived from topic models to classify Twitter IO accounts in French and English speaking networks; and Alizadeh et al.[17] who used post text and URL information to classify Twitter posts as belonging to IO or not. Examples of graph-based approaches include: Monti et al.[18], who used graph networks (GNs) to classify URLs as fake news or not; Vargas et al.[19], who used graph data to classify IO accounts on Twitter which display coordinated behaviour; and Smith et al.[16], who used a network discovery algorithm followed by causal impact estimation to understand the role of individual accounts in propagating IO narratives. Related research not specific to IO includes influential node detection[20,21], which determines the most important nodes with respect to information propagation.

A previously distinct line of research in cybersecurity, kill chain analysis[22–24], focuses on identifying and disrupting threat actors at each phase of their operation. This approach formalizes various sequences of tactics, techniques, and procedures (TTPs) which IO and other cybercrime operations use to achieve their objectives. In particular, online operations kill chains[22] enable the development of technical indicators which are signatures of cybercrime operations at various phases. These indicators can be used to detect future operations, identify abstract themes across campaigns, analyze trends, and compare TTPs across different operations and time periods.

[1]Department of Physics, The George Washington University, Washington, DC 20052, USA. [2]Department of Engineering Management and Systems Engineering, The George Washington University, Washington, DC 20052, USA. ✉email: ngabriel@gwu.edu

These lines of research, as well as reports directly from social media companies, have elucidated a wide range of IO targets, objectives, strategies, and tactics. Many tactics involve the spread of malicious URLs[7], state-backed media, mis/disinformation[25], and particular narratives (e.g., pro-Russian narratives surrounding the Ukrainian war[6,8,9]); other tactics include near-simultaneous link sharing[26], troll farming[7], mass promotion of particular narratives[6,7,16], mass reporting of accounts and content[7,9], and mass spamming or "brigading" of specific pages, posts, and users[7]. Identifying these tactics has enabled well-resourced social media companies such as Twitter, Meta, and Google to automate the detection of new campaigns that reuse TTPs on their respective platforms. This automation has in turn enabled rapid detection and response to coordinated IO activity.

Automated detection has greatly constrained the preferred tactics available to IO on relatively well-regulated platforms such as Facebook and Twitter. For example, networks of coordinated and near-simultaneous link sharing (< 1 min. apart) are now quickly and routinely removed from these platforms[7–11,27]. However, this conspicuous behaviour persists as an IO tactic due to the fact that social media ranking algorithms up-rank content with higher engagement, with immediate engagement having an outsized effect on relative ranking and ultimate reach of content[28]. Hence, to artificially amplify specific narratives during critical periods, IO preferentially coordinate on very short timescales, even at risk of being detected. So while near-simultaneous coordination may be largely curtailed by platforms or even abandoned by IO in the future, coordination on short timescales is expected to continue. For particularly sophisticated IO networks, one would expect that future coordination patterns would mimic that of authentic users.

Fake account detection[7,9–11] has also greatly improved. In response, IO have tried to obviate detection by crafting realistic profiles that mimic authentic users in a process called *persona building*. The process of persona building has presumably been a manual effort to this point, as smaller numbers of these meticulously crafted fake accounts are observed as part of any IO compared to the much larger numbers of less sophisticated "spam" accounts (though part of this discrepancy may be a survivorship bias). A common approach to persona building is to mimic existing accounts that promote narratives favorable to the IO objective, such as inflammatory political content promoted by Russian and Iranian IO campaigns leading up to the 2016 and 2020 U.S. presidential elections[11,27]. This process of mimicry requires significant investment of human effort, as this approach requires the generation of novel content such as text and images. However, it is not difficult to imagine that in the near future a single IO operative could automate the persona building process using novel AI tools to farm a large number of fake accounts. Indeed the use of GAN produced profile pictures[10,22] and deep fakes[29] has been reported. While the automated detection of near-simultaneous coordination and fake accounts will push campaigns towards less efficacious and more costly approaches (Fig. 1), they may be able to compensate with greater scalability, novelty, and specificity enabled by AI.

While mainstream platforms have the resources and desire to improve regulation, alternative platforms are less equipped, and possibly unwilling, to follow suit. Exploiting this situation, the Russian origin Secondary Infektion campaign from 2014–2020 made use of over 300 platforms including WordPress, BlogSpot, Quora, Reddit, and LiveJournal to circulate fake news and seed fabricated primary sources[30]. A subsequent Russian

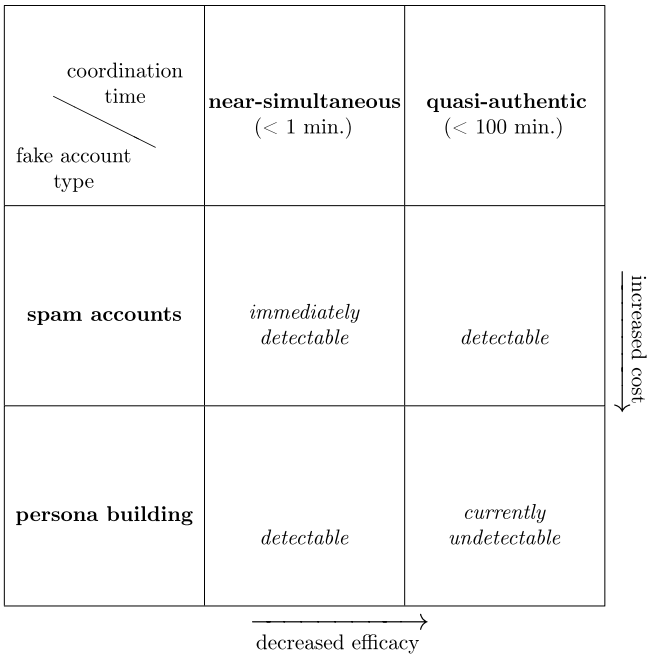


Figure 1. Current landscape of automated detection on mainstream social platforms. Advances in automated detection will push influence operations towards less effective methods of coordination and more costly approaches to fake account creation. In turn, influence operations may be able to compensate by augmenting existing capabilities with emergent AI systems.

campaign from 2020–2022 (likely a continuation of the same operation) targeted 35 alternative platforms that intentionally have little or no regulation such as Gab, Gettr, Parler, and Truth Social[25]. While all of these platforms combined have a smaller audience than most mainstream platforms, they demonstrate continued trends of IO in microtargeting specific audiences and diversifying channels of influence. Countering these trends will require methods of detection that can identify operations across platforms, as well as generalize previous observations on mainstream platforms to newly targeted platforms. Additionally, while mainstream platforms have thus far been proactive in identifying and removing inauthentic actors, it is unclear to what extent this will continue to be true.

Even in light of these trends, continuing to identify and index TTPs for transductive detection will still be paramount to constrain future IO. In other words, the foundation of IO detection will continue to be transductive-or based on specific, previously observed indicators. Transductive approaches will continue to be effective in constraining IO for two main reasons: (1) operations can only develop new TTPs so quickly; (2) previously indexed TTPs often represent the preferred tactics of IO, which they may be slow to abandon. In order to continue shaping public discourse in the near term, one can then expect continued reuse of TTPs, even if these are largely ineffective on mainstream platforms. In the long term, however, one can expect IO to develop novel tactics that avoid detection and reach larger segments of online users. On one hand, this means that future IO will likely have less impact per action (post, like, share, etc.) since they cannot maximally exploit the platforms in which they are embedded. On the other hand, AI systems such as StyleGAN2[31], DeepFaceLab[32], GPT[33], and DALL-E[34] may allow IO to more easily craft realistic profiles and content, thereby enabling novel campaigns that employ previously costly tactics at greater scale. In such cases, it is unclear how effective transductive methods will be, if at all. Hence, developing inductive methods of detection will be necessary to proactively identify and disrupt novel campaigns which can consequentially alter public opinion in a matter of days (e.g., in the days leading up to an election[11,27]). To this end, we observe two fundamental techniques that IO use when manipulating public opinion:

(I) Linking to off-platform websites that are considered credible by the target audience, possessing decreased regulation, and/or containing multimodal content such as text, images, audio, and video;
(II) Coordinated promotion of content supporting specific narratives.

Arguably, IO can have little impact on public opinion without employing these techniques in some form. We use this observation to design (I) content-based and (II) graph-based indicators which are general enough to identify novel campaigns from previous campaigns, and use graph representation learning to encode abstract signatures of coordination from these indicators. In particular, we determine indicators that are not specific to any particular IO campaign by explicitly censoring previously identified content- and graph-based technical indicators. We call indicators resulting from this type of censorship *generalized indicators*, since they will be common across both IO campaigns and authentic users, and also across platforms.

We investigate how specific choices of generalized indicators and graph learning techniques can identify inauthentic actors across IO campaigns, thereby developing a framework that directly complements the transductive methodologies established in previous work. In doing so, we note the correspondence between the generalized indicators used here and previously used technical indicators, summarized in Table 1:

Furthermore, we investigate three specific advances of previous approaches:

(1) Identification of content-based and graph-based indicators which enable cross-operation generalization;
(2) Utilization of graph learning to encode abstract signatures of coordination, thereby automating graph-based feature engineering and inference;
(3) Investigation of a broad coordination window, moving from near-simultaneous (< 1 min.) to quasi-authentic (< 100 min.) interarrival times.

Results

Following the framework presented in Fig. 5, we assess the extent to which specific machine learning models and generalized indicators can identify IO accounts across campaigns, both intra-operation and inter-operation (results shown in Table 2). For this purpose, we select six IO campaigns (Fig. 2) belonging to three coordinated operations: Russia, China, and Iran; and a comprehensive baseline described in the next section. In this case each operation corresponds to a single nation state and each operation has two underlying campaigns. We analyze intra- and inter-campaign co-URL statistics in Fig. 4, demonstrating operation specific trends and the independence of the three operations chosen. In Table 3, we determine which indicators enable cross-campaign generalization using an axiomatic attribution method, integrated gradients[35].

Feature type	Technical indicator[16,17,19,22,26]	Generalized indicator
Content-based	Political and news domains; URLs containing malware, propaganda, and fake news	Censored domains
Graph-based	Graph size, betweenness, clustering	Censored graph learning
Coordination	Near-simultaneous (< 1 min.)	Quasi-authentic (< 100 min.)

Table 1. Correspondence between previously effective technical indicators and generalized alternatives.

Model	F1(val.)‡	F1(test)	AUC(test)	G.E.	Model	F1(val.)	F1(test)	AUC(test)	G.E.
(A): Combined, intra-operation					**(B): Combined, inter-operation**				
LR	92.92	84.94	85.23	∗ † † †	LR	82.97	76.92	77.64	∗ † † †
RF	*97.50*	86.58	86.90	∗ ∗ ∗∗	RF	82.36	81.06	81.52	∗ ∗ ∗∗
MLP	**95.96**	**91.13**	*94.68*	† † † †	MLP	**92.04**	**89.92**	*96.59*	† † † †
GCN	95.33	*91.71*	91.79	† † † †	GCN	86.39	*91.25*	93.92	† † † †
MP-GCN(s)	95.55	91.02	**94.44**	† † † †	MP-GCN(s)	91.65	88.05	**96.28**	† † † †
MP-GCN	95.49	90.64	93.01	† † † †	MP-GCN	*92.40*	88.06	96.05	† † † †
(A1): Rus(18) → Rus(18) / Rus(20)					**(B1): Chn(19) + Iran(19) → Rus(18) / Rus(20)**				
$N_{\text{train}}^{(y=1)} = 2702 \rightarrow N_{\text{val./test}}^{(y=1)} = 676/1059$					$N_{\text{train}}^{(y=1)} = 1920 \rightarrow N_{\text{val./test}}^{(y=1)} = 676/1059$				
LR	96.25	90.27	90.45	† † † †	LR	*91.93*	89.91	90.11	∗ † † †
RF	*99.09*	91.82	92.95	† † † †	RF	83.50	86.46	86.71	∗ ∗ ∗∗
MLP	**97.27**	91.90	*96.72*	† † † †	MLP	**90.11**	90.48	*96.77*	† † † †
GCN	94.80	90.10	93.83	† † † †	GCN	84.64	92.22	93.63	† † † †
MP-GCN(s)	96.29	**92.80**	**95.46**	† † † †	MP-GCN(s)	86.12	**93.21**	94.43	† † † †
MP-GCN	96.36	*92.84*	95.41	† † † †	MP-GCN	86.15	*93.44*	**94.51**	† † † †
(A2): Chn(19) → Chn(19) / Chn(20)					**(B2): Rus(18) + Iran(19) → Chn(19) / Chn(20)**				
$N_{\text{train}}^{(y=1)} = 377 \rightarrow N_{\text{val./test}}^{(y=1)} = 95/4201$					$N_{\text{train}}^{(y=1)} = 4826 \rightarrow N_{\text{val./test}}^{(y=1)} = 95/4201$				
LR	94.54	88.08	88.30	∗ † † †	LR	89.23	82.66	83.08	† † † †
RF	*97.06*	91.75	91.87	∗ ∗ ∗∗	RF	82.31	93.47	93.54	∗ ∗ ∗∗
MLP	**95.44**	92.63	*93.70*	† † † †	MLP	91.52	**94.63**	*97.82*	† ∗∗ †
GCN	95.40	92.83	89.26	† † † †	GCN	90.40	94.41	**97.51**	† † † †
MP-GCN(s)	95.04	*93.57*	**92.56**	† † † †	MP-GCN(s)	*92.10*	94.16	97.38	† † † †
MP-GCN	95.26	**93.33**	90.15	† † † †	MP-GCN	**91.79**	*94.78*	97.29	† † † †
(A3): Iran(19) → Iran(19) / Iran(21)					**(B3): Chn(19) + Rus(18) → Iran(19) / Iran(21)**				
$N_{\text{train}}^{(y=1)} = 1158 \rightarrow N_{\text{val./test}}^{(y=1)} = 290/179$					$N_{\text{train}}^{(y=1)} = 3850 \rightarrow N_{\text{val./test}}^{(y=1)} = 290/179$				
LR	88.36	77.60	78.01	∗ † † †	LR	73.36	71.36	72.56	∗ ∗ ∗∗
RF	*96.38*	77.06	77.64	∗ ∗ ∗∗	RF	71.94	75.01	75.61	∗ ∗ ∗∗
MLP	95.21	**88.95**	**93.69**	† † † †	MLP	**89.57**	**86.58**	*98.41*	† † † †
GCN	95.80	*92.26*	92.40	† † † †	GCN	75.62	*87.39*	92.05	† † † †
MP-GCN(s)	**95.33**	86.97	*95.35*	† † † †	MP-GCN(s)	87.85	83.67	96.15	† † † †
MP-GCN	94.87	86.13	93.61	† † † †	MP-GCN	*90.65*	84.01	**97.37**	† † † †

Table 2. Top: Aggregated intra-operation (**A**) and inter-operation (**B**) results; F1 and ROC-AUC scores are the harmonic mean of the individual subtasks shown in the bottom six tables; G.E. is the median value of subtasks. Bottom: Individual subtask results for intra-operation (**A1**, **A2**, and **A3**) and cross-operation (**B1**, **B2**, and **B3**) classification. We note that the validation and test sets in the intra-operation and cross-operation subtasks are sampled identically, and hence can be compared. Significant values are in bold and italics. Each graph encoding (G.E.) denotes the absence (∗) or presence (†) of **node2vec**, **Laplacian Eigenmaps**, **Random Walk Positional Encoding**, and **Network Features** determined from a censored graph. Each model is trained with $\gamma_{\max} = 0.54$ and $k_{\text{top}} = 2500$ per Figure 3. ‡In sample.

Model and indicator evaluation

We evaluate the effectiveness of several machine learning models-Logistic Regression (LR), Random Forest (RF), Multilayer Perceptron (MLP), Graph Convolutional Network (GCN), deep Message Passing Neural Network (MP-GCN), and shallow Message Passing Neural Network (MP-GCN(s))-on node classification tasks comprising: 10737 influence operation accounts reported by Twitter between 2018 and 2021; and 9793 baseline Twitter accounts not known to be part of any influence operation. The IO accounts were reported in several releases between October 2018 and February 2021, which we split as in Fig. 2 to simulate a prediction task on unseen data. The baseline includes accounts which directly impact public discourse (journalists, media outlets, writers, and academics), random accounts, and accounts highly retweeted by the IO training set. Our goal is to differentiate IO accounts versus this baseline using a set of generalized indicators, as well as determine the optimal graph encoding (G.E.) for each model (i.e. which of **node2vec**, **Laplacian Eigenmaps**, **Random Walk Positional Encoding**, and **Network Features** to include).

In order to assess the generalization capacity of particular model and indicator choices, we formulate two tasks. The first is intra-operation classification (**A** in Table 2), where we train on a campaign of a particular operation (Russia, China, or Iran) and test on a later identified campaign of same operation. The second task is

Train/Validate:

origin	date reported	accounts	target
Russia	Oct. 2018	3378	U.S./Rus.
Iran	June 2019	1448	Varied[‡]
China	Aug. 2019	472	Hong Kong
total		5298	

Test:

origin	date reported	accounts	target
Russia	May 2020	1059	U.S./Rus.
Iran	Feb. 2021	179	U.S.
China	May 2020	4201	Varied[¶]
total		5439	

Baseline:

origin	date reported	accounts	category
U.S./U.K.	-	2681	Political
-	-	5983	Random
Varied[§]	-	1129	Top RT
total		9793	

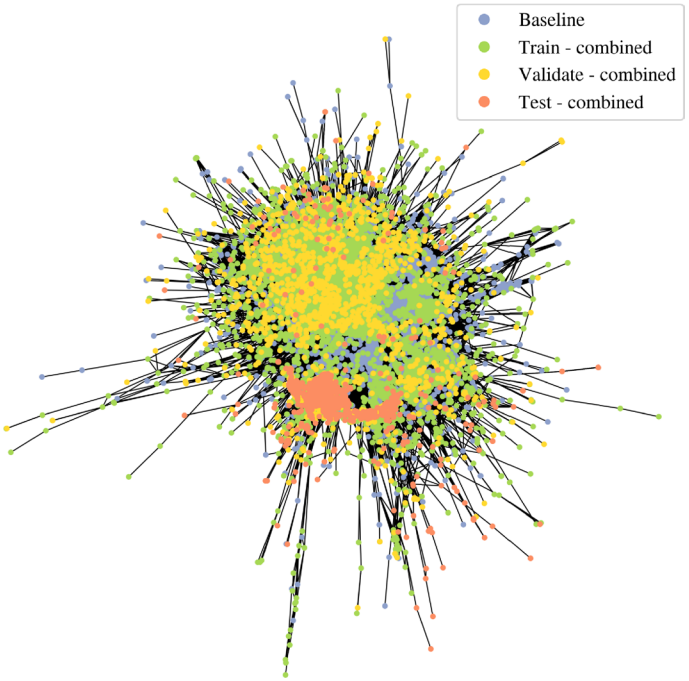


Figure 2. Composite dataset used to assess cross-campaign generalization. Date reported indicates when Twitter released tweet data for each campaign, with the corresponding accounts being removed at some earlier time. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has vertices $\mathcal{V}$ comprising all accounts in the composite dataset, and edges $\mathcal{E}$ indicating one or more co-URLs between accounts. [‡] target audiences includes the U.S., Latin America, Saudia Arabia, Israel, Indonesia. [¶] target audiences includes the U.S., China, and Russia. [§] account origins include U.S., Russia, and China.

inter-operation classification (**B** in Table 2), where we train on all operations *except* the test operation. For the three campaigns in the training/validation set and the test set, this implies three subtasks for **A** and **B**. To enable comparison between the two sets of subtasks, we sample the validation and test sets for each respective subtask identically (e.g., the results of task **A1** and **B1** can be compared directly). This allows us to assess how well each model can generalize from independent operations based on any changes in performance from tasks **A1**, **A2**, and **A3** to tasks **B1**, **B2**, and **B3**, respectively.

We evaluate the effect of varying the content-based feature set both in terms of stringency ($\gamma_{\max}$) and minimum prevalence (k_{top}) on model performance in Fig. 3. We choose MLP with all graph encodings (G.E. = † † † †) as a representative model since it consistently performed well across all subtasks. The effect of increasing k_{top} improves model performance on all metrics in a nearly monotonic manner, ostensibly reaching saturation around $k_{\text{top}} = 2000$. The effect of varying the maximum frequency ratio ($\gamma_{\max}$) has a more nuanced effect on performance, but a fairly stringent value of $\gamma_{\max} \in \{0.43, 0.67\}$ appears to produce greater generalization than smaller or larger values, with increases beyond $\gamma_{\max} = 0.67$ producing a nearly monotonic decrease in performance for F1(val/test), but not for AUC.

Coordination analysis of composite dataset

In order to understand intra- and inter-operation patterns of coordination, we report co-URL counts between all campaigns in Fig. 4. The 2×2 block pattern of co-URL counts along the diagonal of 4a and 4b suggests that each campaign of a particular origin is actually a continuation of the same underlying operation. Observing how these co-URLs are distributed as a function of interarrival time, we see that the earlier identified training campaigns in 4c have similar distributions of co-URLs as the later identified test campaigns in 4c in some cases. In particular, the campaigns of Chinese and Iranian origin appear to adopt near-simultaneous link sharing later than the Russian campaigns. This can be quantified by calculating the distance between CDFs for each campaign (4e). From 4f, we see that the accounts in the Chn.(19) and Iran(19) campaigns appear to use little to no coordination at short timescales compared to the baseline, but the Chn.(20) and Iran(21) campaigns begin to display levels of coordination comparable to the Rus.(18) campaign. The Rus.(20) campaign displays greater levels of coordination than any campaign in our dataset, particularly at short timescales. This delay in near-simultaneous coordination by the Chinese/Iran operations could be due to adoption of this strategy from the Rus.(18) campaign, which made extensive use of co-URLs. Specifically, this strategy could have been directly observed and later adopted as follows: in 4a , we see that the Chn.(19)/Iran(19) campaigns (which do not use near-simultaneous coordination) interact with both Rus.(18)/Rus.(20) campaigns; and in 4d we see adoption of near-simultaneous co-URLs by the later Chn.(20)/Iran(21) campaigns well above the baseline.

Feature	IG (val.)	IG (test)	IG (base.)	Feature	IG (val.)	IG (test)	IG (base.)
(A): Combined, intra-operation				(B): Combined, inter-operation			
Domains	9.32×10^{-2}	1.06×10^{-1}	4.52×10^{-1}	Domains	7.84×10^{-2}	9.31×10^{-2}	4.50×10^{-1}
node2vec	2.11×10^{-1}	3.42×10^{-1}	1.87×10^{-1}	node2vec	1.94×10^{-1}	3.38×10^{-1}	1.93×10^{-1}
LE	9.53×10^{-5}	5.32×10^{-4}	3.92×10^{-5}	LE	9.98×10^{-5}	5.18×10^{-4}	4.65×10^{-5}
RWPE	1.18×10^{-1}	2.83×10^{-1}	3.13×10^{-1}	RWPE	1.15×10^{-1}	2.87×10^{-1}	3.27×10^{-1}
NF	7.00×10^{-1}	8.68×10^{-1}	7.90×10^{-1}	NF	6.38×10^{-1}	8.36×10^{-1}	7.84×10^{-1}
Degree	3.62×10^{-1}	3.02×10^{-1}	2.60×10^{-1}	Degree	3.31×10^{-1}	2.97×10^{-1}	2.65×10^{-1}
Cluster. coef.	9.51×10^{-3}	1.56×10^{-3}	7.34×10^{-3}	Cluster. coef.	4.92×10^{-3}	3.17×10^{-3}	7.22×10^{-3}
Betweenness	1.39×10^{-3}	4.04×10^{-2}	7.83×10^{-2}	Betweenness	3.65×10^{-3}	3.97×10^{-2}	8.08×10^{-2}
Pagerank	1.75×10^{-1}	2.87×10^{-1}	3.60×10^{-1}	Pagerank	1.58×10^{-1}	2.74×10^{-1}	3.60×10^{-1}
HITS	1.45×10^{-1}	2.20×10^{-1}	7.74×10^{-2}	HITS	1.35×10^{-1}	2.21×10^{-1}	7.10×10^{-2}

Table 3. Mean absolute integrated gradients (IG) of trained MLPs over features for validation, test, and baseline subsets. For all features, we report the sum of absolute values to avoid cancellation due to conflicting signs. Additionally, we report the IG of each of the five quantities comprising (**NF**). For IG values of individual subtasks and domains, see Tables 4 and 5 in the SI.

Feature importance

For each model and subtask, we report the best performing graph encoding in each case (G.E. in Table 2). However, it is not clear from these results what the relative importance of each graph-based feature is on model predictions, or the relative importance of content-based to graph-based features. To quantify the relative importance of each feature set, we calculate the mean absolute integrated gradients (IG) for each feature over validation, test, and baseline sets for each subtask (Table 4 in the SI). In Table 3 we report the aggregated (arithmetic mean) IG values of each feature over all subtasks. We again use MLP as a representative model since it consistently performs well on all subtasks.

Overall, the net attribution of all graph-based features (**node2vec**, **LE**, **RWPE**, **NF**) appears to be substantially larger than that of all content-based features (**domains**), greater by roughly an order of magnitude. Notably, several quantities widely used in network analysis such as Laplacian Eigenmaps (**LE**), clustering coefficient, and betweenness centrality had a marginal impact on predictions, indicating that they provided little useful information for predictions and, since dropout was employed, that this information was not even redundant with other features. Meanwhile, graph embedding techniques such as **node2vec** and **RWPE** enjoy a relatively high utility, having a substantial impact on predictions. This result is not necessarily surprising since **node2vec** and **RWPE** essentially act as deep encoders, which can be decoded with high fidelity by deep neural networks (i.e. MLP and GNs, but not by LR and RF). What *is* surprising, on the other hand, is that several simple network quantities-degree, pagerank, and HITS-were as important to predictions as any other single feature. This implies that these quantities encode some information that is complementary to graph embedding techniques, and do so with only a single scalar value.

Discussion

Constraining influence operations is an ongoing challenge that will require continued advancement of detection capabilities in order to counter novel operations-particularly as they adopt powerful AI technologies. In particular, detection methods which go beyond established transductive methodologies and can identify novel campaigns in an inductive manner will be critical. Here we have examined the systematic application of generalized indicators and graph learning techniques, demonstrating a framework in Fig. 5 which enhances detection coverage. Furthermore, this framework is broadly applicable to detecting manipulation on social media, and naturally complements detection using technical indicators identified in transductive methodologies.

Overall, the most effective approaches utilized: (1) a fairly large content-based feature set (approximately 2000–2500 domains) with fairly stringent removal threshold applied ($\gamma_{\max} \approx 0.5$); (2) a broad range of graph-encoding features, particularly **node2vec**, **RWPE**, degree, pagerank, and HITS (hyperlink-induced topic search); (3) a deep neural architecture. In other words, MLP and the three GNs outperformed LR and RF on every out-of-sample subtask. On the in-sample prediction tasks, as quantified by F1(val) of tasks **A1**, **A2**, and **A3**, RF actually outperforms all of the deep models. It appears that in this case RF was simply able to memorize patterns specific to the training set, as it fails to generalize to the test set. Notably, on tasks **A2** and **A3**, RF achieved superior validation set performance and optimal test set performance with no graph encoding features at all, while all other models saw improved generalization from these features. These results suggest that RF is ill suited to make use of graph-based features, and moreover, fails to generalize both content-based and graph-based features. For task **A**, this failure corresponds to a ~ 5 point decrement on F1(test) and AUC(test) relative to the deep models. On task **B**, for which predictions had to be made by generalizing across campaigns, an even larger decrement of 10–15 points is observed across all metrics.

Among deep models, MLP consistently demonstrates high performance across all subtasks, achieving superior AUC performance in all but one case. However, on all out-of-sample F1 scores, one or more GNs outperformed

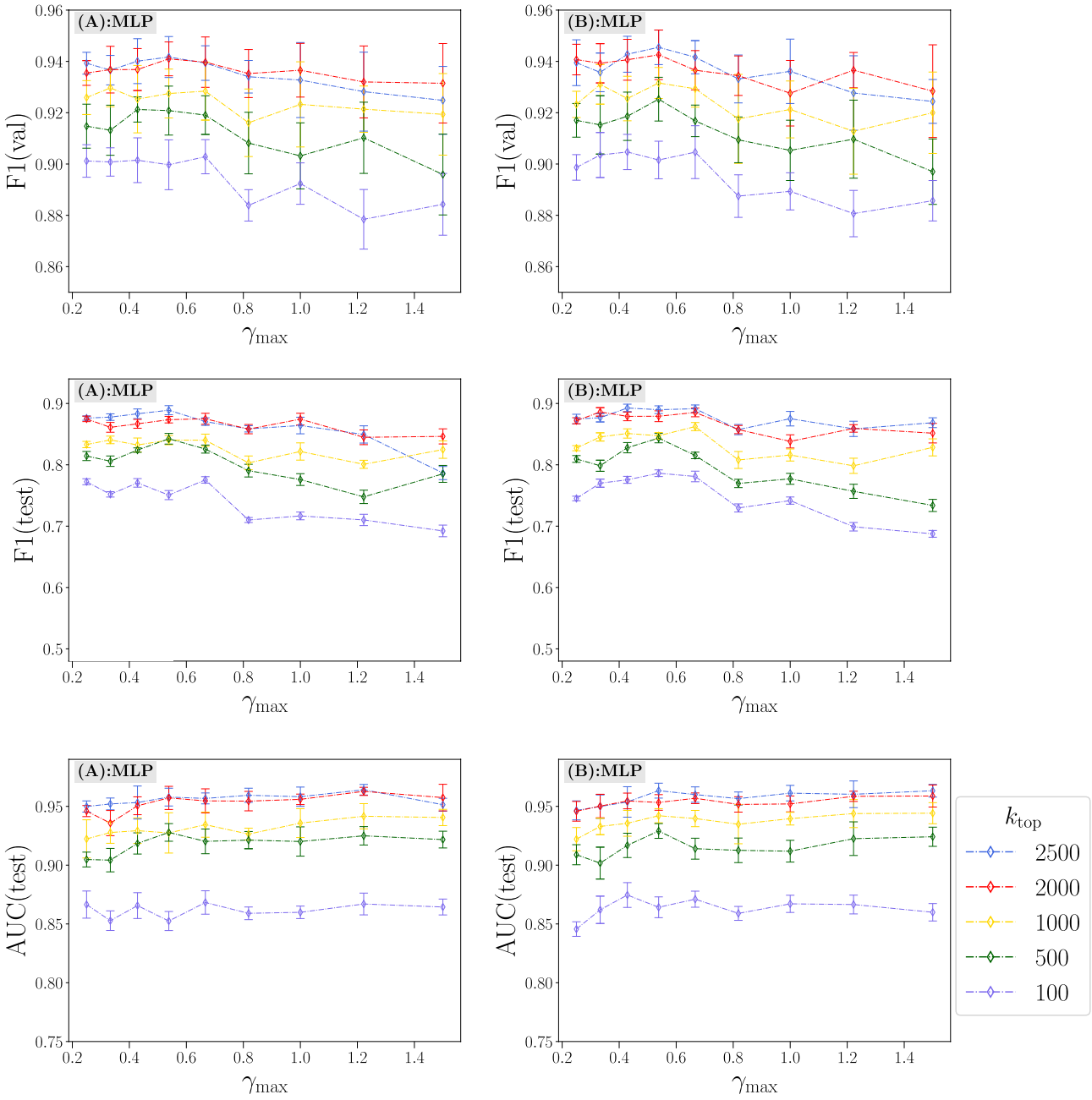
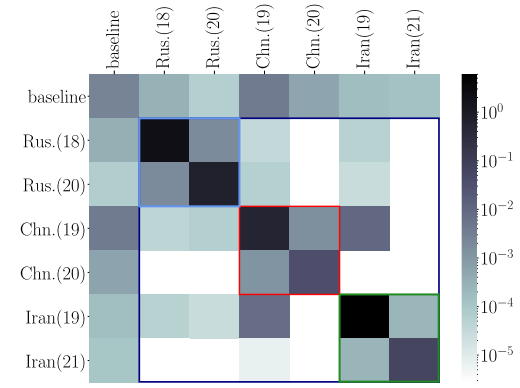


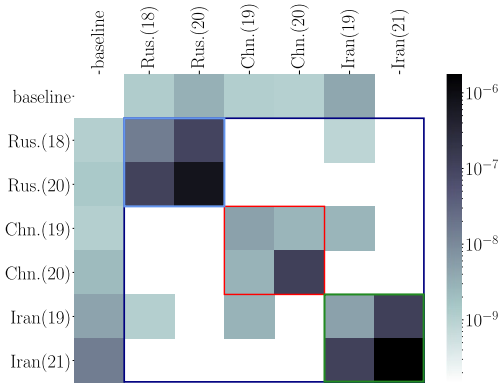
Figure 3. Top to bottom: Aggregated F1(val), F1(test), and AUC(test) for inter-operation (**A**, left column) and inter-operation (**B**, right column) classification with varying $\gamma_{\max}$ and k_{top}. Errors are calculated from individual subtasks (**A1**, **A2**, **A3** and **B1**, **B2**, **B3**) using uncertainty propagation.

MLP. This indicates that while GNs perform well at the decision boundary for classification (the näive boundary $\alpha = 0.5$ in all cases), misclassifications were by a greater margin than for MLP. This is a possible indication that while the increased expressive power of GNs was beneficial for classification on average, it could lead to even further errors on accounts which were misclassified. Moreover, these results indicate that MLP with graph encoding features may achieve superior performance at higher/lower decision boundaries, which may be relevant for applications with a greater/lesser tolerance for false positives.

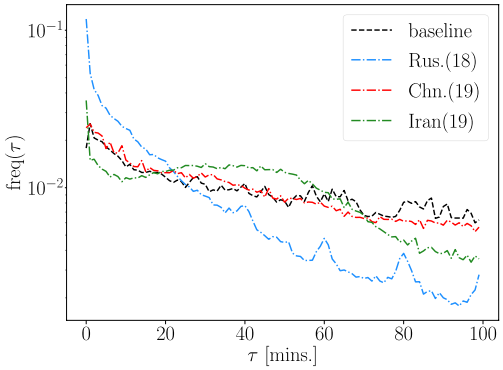
While we have outlined several specific approaches for selecting feature sets and models which can generalize across campaigns, there are a number of improvements which would likely further this work. First, there are presumably other content-based indicators which may aid in out-of-sample identification of IO accounts. Namely, we did not investigate text, images, audio, and video contained directly in posts. On one hand, multimodal embeddings of such content that encode specific narratives or ideologies could presumably provide useful information which is not necessarily specific to a particular IO campaign. On the other hand, off-platform URLs can in principle contain all of these modalities, pose additional moderation challenges to platforms, and have a similar function regardless of platform. For content-based features overall, a unified approach for encoding the content and semantics of text, images, audio, video, and web pages would be ideal, since focusing on a specific



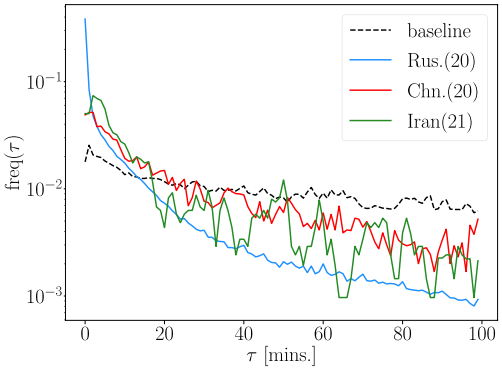
(a) Frequency of co-URLs between each subset. Counts are normalized by the size of the leading and lagging subsets.



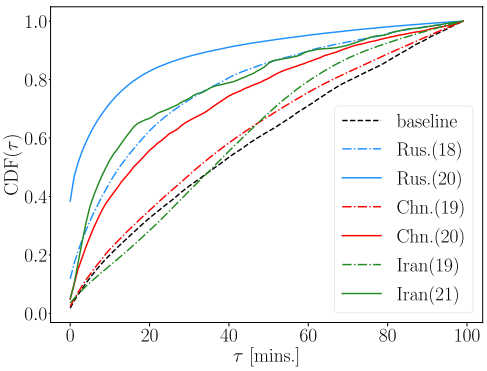
(b) Frequency near-simultaneous co-URLs ($\tau < 1$ min.) between subsets. Counts are normalized by the size of the leading and lagging subsets.



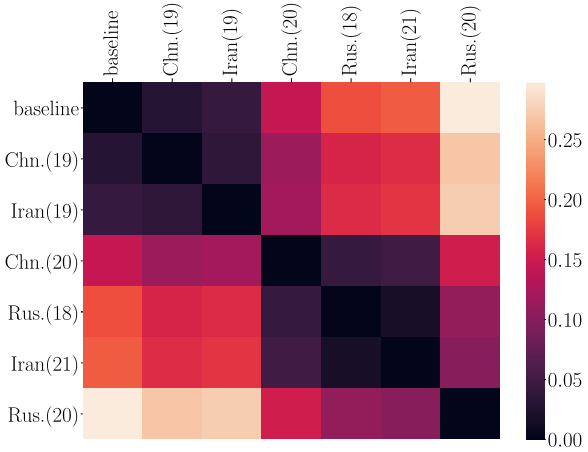
(c) Training set co-URL counts as a function of interarrival time τ. Each series corresponds to a diagonal element in Figure 4a and 4b above.



(d) Test set co-URL counts as a function of interarrival time τ. Each series corresponds to a diagonal element in Figures 4a and 4b above.



(e) Cumulative distribution function of co-URL counts for all campaigns.



(f) 1-Wasserstein distance W_1 between campaigns. An increase in near-simultaneous link sharing was observed across operations even as automated detection improved.

Figure 4. Summary of co-URL statistics for the 6 IO subsets and baseline.

form of content could lead to blind spots. For example, the multimodal encoders used by the multi-task agent Gato[36] or the GPT-4 system[33] could allow for training on text posts and making inferences on video posts, and so on. Additionally, the content of both direct posts and off-platform web pages could be compared on equal footing.

Though the co-URL is a effective and robust tool for quantifying coordination, there are many other edge-wise features e_{ij} which quantify pair-wise relationships in a graph. In particular, several graph-based measures which quantify similarity could add useful information: node-similarity measures based on nearest neighbors such as common neighbors, Jaccard Index, Adamic Adar, and preferential attachment coefficients; as well as path

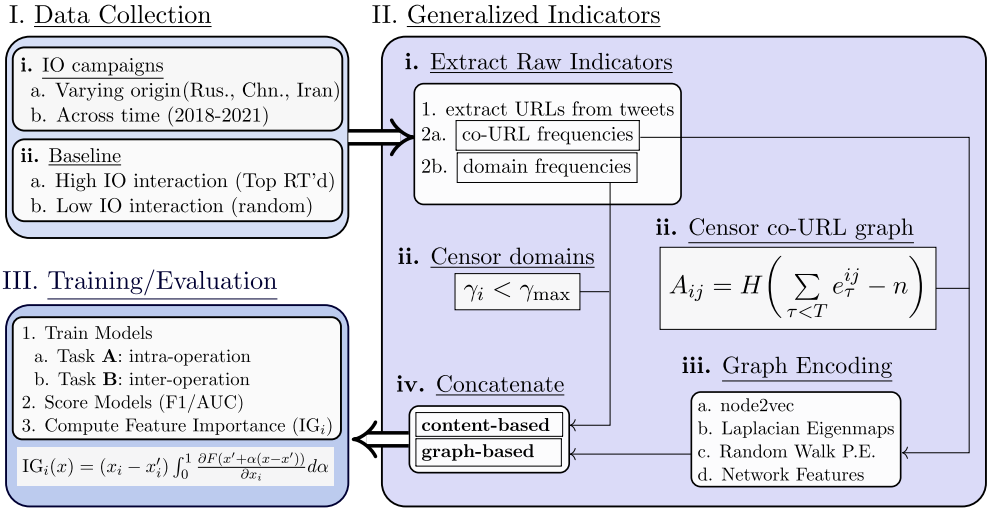


Figure 5. Illustration of the proposed inductive learning framework for the detection of information operations: **(I.)** Collect IO data spanning various operations and time periods, as well as a baseline that interacts with the IO to varying degrees; **(II.)** Extract and censor raw content-based and graph-based indicators and encode signatures of coordination via graph learning; **(III.)** Evaluate model performance on tasks requiring generalization and determine the most important indicators using feature attribution. $H(\cdot)$ is the Heaviside step function,γ_i is the relative frequency of the ith domain in the IO training set ($y = 1$) relative to the baseline training set ($y = 0$), and e_τ^{ij} is a co-URL vector. .

based measures such as shortest path lengths, Katz measure, and hitting time. Other similarity measures can be derived from graph learning measures by applying various distance metrics such as L_p, cosine, and Sørensen-Dice distances to pairs of graph encodings. Fortunately, message passing graph networks provide a natural way to incorporate similarity measures (or any edge-wise features) into predictions, making this type of extension straight forward.

Although we attempted to present a range of graph networks-convolutional, shallow message-passing, and deep message-passing graph networks-there are myriad design dimensions of graph networks which we did not explore. Among these are more advanced sampling strategies, attention mechanisms, and various message passing architectures. However, the results in this study are adequate to suggest that both graph learning and graph networks will be an indispensable tool for detecting IO into the future.

Finally, we examined a "hard" measure of coordination, the co-URL, in this paper. There could in the future, however, be softer forms of coordination which evade detection. For example, different URLs could lead to semantically or literally identical content, which would not be measured as coordination by our current approach. To hedge against this possibility, one could encode the content of URLs as embeddings and define coordination as a function of the distance between embeddings. This would generalize the current co-URL approach in which we implicitly assign identical URLs distance 0 and distinct URLs distance ∞. This is yet another indication of the utility of multimodal content encoders in future influence detection efforts.

In summary, we have demonstrated an inductive approach to detecting IO which allow for continued utility into the future and generalization capacity across campaigns, enabling identification beyond technical indicators identified by transductive methodologies. We have illustrated how specific content- and graph-based features realize these objectives, as well as how one can systematically identify these features. Finally, we have identified several refinements of the current approach, enabling continued advancement in the automated detection of IO even as these campaigns continue to evolve.

Methods

Data collection and inclusion criteria

For the purpose of evaluating intra- and inter-operation generalization of machine learning models, we selected IO of several origins (Russia, China, and Iran) for which there were significant campaign sub-networks identified at different times. To this end, six Russian, Chinese, and Iranian origin campaigns identified by Twitter between 2018 and 2021 were suitable. Another important aspect considered was the availability of baseline accounts which both interacted with the IO (to provide adequate coordination measures) and remained independent of IO (to reduce bias in frequency measures). To this end the baselines of[17,19] were used (88.5% of baseline accounts), in addition to 1129 accounts (11.5% of baseline accounts) interacted highly with IO, as measured by co-URLs. Additionally, the 1129 high-interaction accounts were sampled at various maximum follower thresholds ($n = 10^2, 10^3$, and 10^4) since accounts with many followers ($n > 10^4$) were disproportionately interacted with by IO accounts. This resulted in an aggregate baseline which was highly connected to the IO and yet provided broad coverage of various types and sizes of accounts.

For this study, we focus specifically on IO accounts which displayed reasonably organic patterns of sharing, which evaded detection for some amount of time, and which could have reasonably had some impact on public discourse. We therefore selected from the initial data set accounts which: (1) were active for at least 3 months; (2) had at least 300 Tweets; (3) had at least 200 URL shares; (4) shared at least 5 unique domains; (5) had at least 10 co-URLs with at least 2 neighbors.

Extracting raw indicators

To obtain features from the raw tweet data, we expand all shortened URLs contained in tweets using the URL-Expander library[37]. Then, to obtain domain counts for each account, we use the tldextract library to extract the domain of each URL tweeted by an account. We then generate node-wise and edge-wise features from the raw URLs and extracted domains:

Edge features As a measure of coordination between accounts, we compute the interarrival time between shares of the same URL (co-URLs) and bin the results into 1 minute intervals to obtain a vector of co-URL frequencies between all pairs of accounts. Denoting the interarrival time as τ, the co-URL count between the ith and jth account in the interarrival window $\tau - 1 < t \leq \tau$ is then denoted e_{τ}^{ij}.
Node features We use the raw counts of the most frequently shared top-level domains (e.g.,, cnn.com, youtube.com, nytimes.com) from each account in the composite dataset. To avoid having the models simply memorize domains which are specific to a particular IO, we censor domains where more than $\gamma_{\max}$ of occurrences of the domain originate from the IO training set relative to the baseline set. For example, riafan.ru, histantv.com, and tel-avivtimes.com are censored by this method for any $\gamma_{\max} < 10^4$ since each of these domains originate at least 10^4 more frequently from IO accounts than from the baseline. See SI section E for extended examples of censored domains.
Graph Encoding From a censored co-URL graph we compute three graph embeddings-node2vec (dim = 128), Laplacian Eigenmaps (dim = 50), Random Walk Positional Encodings (dim = 50), and several network statistics (degree, clustering coefficient, betweenness centrality, pagerank and HITS). We define the concatenation of all graph-based features the graph encoding.

Content-based generalized indicators

Following transductive methodologies, previous work has enabled the rapid detection of IO which attempt to propagate specific domains containing fake news, propaganda, and malware[7,10,29]. Accounts sharing these domains, particularly in a coordinated manner, are now routinely identified and removed by mainstream platforms. To identify influence efforts beyond these more flagrant indicators, we investigate domains which are commonly shared and yet may be useful indicators of IO activity. In order to quantify the extent to which a particular domain is either common to some baseline users or specific to an IO, we define the relative frequency for the ith domain in the IO training set ($y = 1$) relative to a baseline set ($y = 0$) as

$$\gamma_i = \frac{\mathrm{tf}(i, y = 1)}{\mathrm{tf}(i, y = 0)} \tag{1}$$

where the domain term frequencies for the ith domain are

$$\begin{aligned} \mathrm{tf}(i, y = 1) &= \frac{f_i^{(y=1)}}{\sum_i f_i^{(y=1)}} \\ \mathrm{tf}(i, y = 0) &= \frac{f_i^{(y=0)}}{\sum_i f_i^{(y=0)}} \end{aligned} \tag{2}$$

and $f_i^{(y=1,0)}$ are the raw counts of the ith domain in the IO and baseline training sets. We then censor any domains which exceed a threshold $\gamma_{\max}$ such that we remove domains which are specific to the IO training set with variable stringency. Particular choices of $\gamma_{\max}$ can censor domains which only appear in the baseline set ($\gamma_{\max} = 0$), appear in the IO set no more than parity ($\gamma_{\max} = 1$), or which appear only in the IO set ($\gamma_{\max} = \infty$). Additionally, we retain only a select number of the censored domains, k_{top}, the top-k domains when sorted in descending order by absolute frequency. We can then vary the stringency and minimum prevalence of our content-based feature set with $\gamma_{\max}$ and k_{top}, respectively, in order to investigate the effect of content censorship on generalization. Moreover, at less stringent thresholds ($\gamma_{\max} > 1$), we can observe the effect of directly including technical indicators of previous campaigns used in transductive methodologies.

Graph-based generalized indicators

Coordination by IO on social platforms has taken many forms, including mass spamming, mass reporting, and coordinated content sharing. Among these tactics, coordinated content sharing has perhaps been the most widely observed, and is the chief tactic employed by many campaigns. In particular, many takedown efforts have used near-simultaneous co-URL sharing as the primary means of both identifying and substantiating coordinated inauthentic behaviour. In addition to the ubiquity of co-URLs across a variety of campaigns and platforms, they also have the appealing properties that they are agnostic to the specific content shared, are easily defined across platforms, and automatically imply a graph structure between accounts. Furthermore, each co-URL has an associated time between shares, the *interarrival time*, whose distribution can provide further insight into coordinated activity between accounts (see Fig. 4 for examples).

While near-simultaneous co-URLs are a useful indicator for automated detection of IO activity, this behaviour is not guaranteed to persist, particularly for campaigns with high operational security (i.e., those which closely mimic authentic users). Therefore, generalized indicators of coordination should incorporate a broader time frame within which future campaigns are likely to operate. In particular, we utilize co-URLs with interarrival times from $\tau = 0$ to 100 minutes. This time frame includes near-simultaneous sharing (< 1 min.), the majority of retweets (< 20 min.[38]), and the median half-life of tweet views (∼ 80 mins.[39]). We denote the number of co-URLs with interarrival times $\tau - 1 < t \leq \tau$ as e_τ^{ij}, and the composite co-URL vector as $e_{ij} = \left\{ e_1^{ij}, \ldots, e_{100}^{ij} \right\}$.

The graph structure implied by co-URLs, however, cannot be used directly by machine learning models to make node-wise inferences. Two approaches for utilizing graph structured data in machine learning applications are to: (i) learn unsupervised feature vectors for each node in the graph (*graph embedding*); and/or (ii) define graph operators which systematically aggregate data over the graph at each layer in a neural network. Both of these techniques can be referred to collectively as *graph learning*[40], and neural networks utilizing graph operators as *graph networks*.

Graph networks can utilize co-URL data in ways which may or may not directly make use of near-simultaneous link sharing behaviour, thereby offering varying degrees of generalization capacity. For example, one can define a graph which censors near-simultaneous link sharing as follows: assign $A_{ij} = 1$ if and only if two accounts share at least n URLs with interarrival times less than T, or in mathematical notation

$$A_{ij} = H\left(\sum_{\tau < T} e_\tau^{ij} - n \right) \tag{3}$$

where $H(\cdot)$ is the Heaviside step function. This definition equally counts the contribution of all co-URLs with interarrival times less than T, thereby censoring any near-simultaneous behavior while still allowing a rigorous threshold for coordination. One can then define graph operators, such as GCN[41], in terms of this censored graph. A standard way of directly using vector-valued edge features such as co-URLs, on the other hand, is within a message passing[42,43] framework. Message passing defines graph operators directly as a function $\phi(e_{ij})$ of the edge-wise feature vectors (i.e., allowing predictions to be made directly using near-simultaneous co-URLs e_1^{ij}). In order to understand the generalization capacity of graph networks with varying degrees of graph censorship, we implement three graph networks as follows: GCN, which utilizes only the censored graph; MP-GCN(s), a message passing variant of the base GCN architecture with a shallow message passing function $\phi = L$, where L is a linear operator; and MP-GCN, which uses the more common deep message passing function $\phi = f$, where f is a neural network. Comparing the performance of these three architectures allows us to examine the effect of graph censorship, as well as compare different graph network architectures in identifying IO.

Graph encoding

In order to understand the utility of different types of graph-based features (from network analysis to graph learning) as well as the utility of specific features, we incorporate several candidate quantities in a node-wise feature vector which we call the graph encoding. Due to the asymmetric nature of our dataset (co-shares of content *by* IO accounts are visible in the dataset, but co-shares *of* IO account content have been removed by Twitter) we treat all graph quantities in an undirected manner by setting $e_{ij} \leftarrow e_{ij} + e_{ji}$. All graph-based features are derived from an undirected graph computed from the co-URL vectors as in Eq. 3 where we select thresholds of $n = 10$ and $T = 15$ to censor the graph. While more stringent n would produce a more robust graph, we find that further reducing the number of edges rapidly disjoints the graph, making graph learning techniques infeasible. The temporal threshold $T = 15$ minutes represents a window in which IO could coordinate effectively and yet avoid detection, while also censoring near-simultaneous link sharing.

Graph representation learning, including graph embedding algorithms such as node2vec[44], originated as an effort to automate the feature engineering process for graph prediction tasks such as node classification and link prediction. From a modern perspective, graph embedding techniques are unsupervised methods which allow one to systematically assign relational, functional, and structural information to each node in a graph. This information can greatly improve the performance of deep learning models, with or without graph operators, on graph prediction tasks. We choose three graph embedding algorithms for our purpose here: (1) node2vec, which encodes neighborhood information of nodes into dense embeddings; (2) Laplacian Eigenmaps, a non-linear spectral embedding technique which provides a local coordinate system on graphs and effectively encodes clustering within the graph; (3) Random Walk Positional Encoding, which is based on the graph diffusion operator and uniquely assigns node embeddings based on the k-hop topological neighborhood of each node. Each of these approaches, in principle, encode different aspects of graph topology and therefore can provide predictive utility independent of one another. In each case, the dimensions of the embeddings are chosen such that further increases yield no benefit to performance across models.

While graph embeddings are a sensible method of encoding topological information for predictive tasks, they do not necessarily preclude the utility of conceptually similar network analysis quantities. To this end we include several quantities which encode relational, functional, and structural information of graphs in our graph encoding: (1) degree, which for undirected graphs is simply the number of directly adjacent neighbors of each node; (2) clustering-coefficient, which quantifies the local clustering of each node as the amount of closure between the neighbors of each node; (3) betweenness centrality, a centrality measure quantifying the extent to which a node facilitates connection within the graph via shortest paths; (4) pagerank, a centrality measure which ranks nodes according to their relative importance within a network; (5) HITS, which also ranks nodes according to relative importance but assigns two scores quantifying the extent to which a node connects the graph (hub score)

and is of relative importance within the graph (authority score). For undirected graphs, the hub and authority scores of HITS are identical.

Graph learning

Given the node-wise and edge-wise data in our feature set, there are several graph learning techniques that are viable choices for the predictive task at hand. Firstly, an MLP with graph encodings serves as a baseline to compare against graph network architectures. We then select several GN architectures of increasing expressive power such that we can compare the utility of different GN design choices and degrees of graph censorship. In particular, we perform ablation on message-passing rules for encoding the co-URL vectors e_τ^{ij}.

In general, a graph network can be written as the series of operations

$$h_i^{(l+1)} = W^{(l)} h_i^{(l)} + b^{(l)} \qquad \text{(affine transformation)} \tag{4}$$

$$h_i^{(l+1)} = \underset{j \in \mathcal{N}(i)}{\text{AGG}} \left(h_j^{(l+1)} \right) \qquad \text{(feature aggregation)} \tag{5}$$

$$h_i^{(l+1)} = \sigma \left(h_i^{(l+1)} \right) \qquad \text{(non-linearity)} \tag{6}$$

where the set $\mathcal{N}(i)$ indicates the neighborhood of the ith node where $A_{ij} = 1$. The simplest graph network that we employ is a spectral GN, the popular Graph Convolutional Network (GCN), with layers defined by the feature aggregation function[41]

$$\underset{j \in \mathcal{N}(i)}{\text{AGG}} = \sum_{j \in \mathcal{N}(i)} \frac{1}{\sqrt{d_i d_j}} h_j^{(l+1)} \qquad \text{(GCN)} \tag{7}$$

where d_i is the degree of the ith node.

There are a number of ways in which message passing rules can be defined, but for graph networks one typically defines the message passing rule as

$$m_{ij}^{(l+1)} = \phi \left(h_i^{(l)}, h_j^{(l)}, e_{ij} \right) \tag{8}$$

where the message passing function $\phi(\cdot)$ can take as input both nodewise features $h_i^{(l)}$ and edgewise features e_{ij}. We then incorporate these messages into the aggregation step as

$$h_i^{(l+1)} = \underset{j \in \mathcal{N}(i)}{\text{AGG}} \left(h_j^{(l+1)}, m_{ij}^{(l+1)} \right) \tag{9}$$

We employ two message passing rules to encode the co-URL vector, the first of which is a shallow message passing rule which defines our MP-GCN(s):

$$m_{ij}^{(l+1)} = \sigma \left(\sum_\tau w_\tau^{(l)} e_\tau^{ij} \right). \tag{10}$$

The second rule utilizes a neural message passing function, implemented as an L layer perceptron which defines our MP-GCN:

$$\begin{aligned} a_{ij}^{(k+1)} &= \sigma \left(W^{(k,l)} a_{ij}^{(k)} + b^{(k,l)} \right); \\ m_{ij}^{(l+1)} &= \sigma \left(W^{(L)} a_{ij}^{(L)} + b^{(L)} \right); \end{aligned} \tag{11}$$

where $W^{(l)}$ and $b^{(l)}$ are the weights and biases of the lth layer and $a_{ij}^{(0)} = e_\tau^{ij}$. To compare with the base GCN implementation, we insert each message passing rule into the base GCN aggregation function as

$$\underset{j \in \mathcal{N}(i)}{\text{AGG}} = \sum_{j \in \mathcal{N}(i)} \frac{m_{ij}}{\sqrt{d_i d_j}} h_j^{(l+1)}. \qquad \text{(MP-GCN(s)/MP-GCN)} \tag{12}$$

Thus we have performed ablation on the message passing rule over the three GCN architectures.

Model training

In the LR and RF implementations we tune all hyperparameters to achieve the best model performance via gridsearch. In the MLP and the three GCN variants we use the same hyperparameters: two hidden layers of 64 units, and a dropout probability $p = 0.5$ applied to all units in the hidden layers. In all message passing layers we apply a dropout probability of $p = 0.2$. For all MLP and GCN training we use a binary cross entropy loss and the Adam optimizer with a learning rate of 10^{-4}.

Integrated gradients

Integrated gradients[35] is an axiomatic attribution method for deep neural networks. Mathematically, the IG of a function $F(x)$ with respect to the ith component of an input x and a baseline x' is

$$\text{IntegratedGradient}_i(x) = \left(x_i - x_i'\right) \int_0^1 \frac{\partial F\left(x' + \alpha\left(x - x'\right)\right)}{\partial x_i} d\alpha \tag{13}$$

where α parameterizes a straight line path from x' to x. This method provides a more robust attribution of predictions to specific features than directly evaluating the product of the gradient and feature value

$$\text{Attr}_i(x) = x_i \frac{\partial F}{\partial x_i} \tag{14}$$

which has historically been a popular attribution method. When using IG, one selects a baseline where the model prediction is neutral. Calculating the IGs of each feature for an MLP, there is not an obvious baseline which yields neutral predictions, i.e., where $F(x') = 0.5$. For example, simply choosing the mean or minimum value of each feature over various subsets of the data produces predictions close to 0 or 1. We therefore construct an empirical baseline comprising the subset of all nodes such that $0.4 \leq F(x_j) \leq 0.6$, or within $\pm$ 0.1 of a neutral prediction. Setting $x' = \langle x_j \rangle$ then yields $F(x') = 0.534 \pm 0.018$ over all six subtasks, which is approximately neutral while ensuring that no particular feature in the baseline takes on an extreme value (which might be the case if we simply chose j to be the single most neutral prediction).

Error propagation of aggregated metrics

In Fig. 3, several performance metrics are aggregated over subtasks by computing their harmonic mean. For each subtask and choice of parameters ($\gamma_{\max}$ and k_{top}), there is an associated uncertainty for each metric due to their dependence on a random samples of train/validate/test splits in the data. In order to compare aggregated results for different parameter values, we propagate the uncertainties associated with each metric as follows. In general, the harmonic mean can be written

$$\tilde{x} = \frac{n}{\sum_{i=1}^{n} x_i^{-1}} \tag{15}$$

and the propagated uncertainty (neglecting correlations between x_i)

$$\sigma_{\tilde{x}}^2 \approx \sum_{i=1}^{n} \left(\frac{\partial \tilde{x}}{\partial x_i}\right) \sigma_{x_i}^2. \tag{16}$$

The partial derivatives of $\tilde{x}$ with respect to each x_i are

$$\frac{\partial \tilde{x}}{\partial x_i} = \frac{\tilde{x}^2}{n} \frac{1}{x_i^2} \tag{17}$$

and the propagated uncertainty is then

$$\sigma_{\tilde{x}}^2 = \left(\frac{\tilde{x}^2}{n}\right)^2 \sum_{i=1}^{n} \left(\frac{\sigma_{x_i}}{x_i^2}\right)^2. \tag{18}$$

Using this result we can better understand different choices of $\gamma_{\max}$ and k_{top} shown in Fig. 3.

Data availability

Data for this study is available from the corresponding author by request.

Received: 27 June 2023; Accepted: 11 December 2023
Published online: 19 December 2023

References

1. Broniatowski, D. A. *et al.* Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate. *Am. J. Public Health* **108**, 1378–1384 (2018).
2. Zannettou, S. *et al.* Who let the trolls out? Towards understanding state-sponsored trolls. In *Proceedings of the 10th ACM Web Science* 353–362 (2019).
3. Zhou, Y., Dredze, M., Broniatowski, D. A. & Adler, W. D. Elites and foreign actors among the alt-right: The gab social media platform. *First Monday* (2019).
4. Linvill, D. L. & Warren, P. L. Troll factories: Manufacturing specialized disinformation on twitter. *Political Commun.* **37**, 447–467 (2020).
5. Rossetti, M. & Zaman, T. Bots, disinformation, and the first impeachment of US president Donald Trump. *PloS one* **18**, e0283971 (2023).
6. Nimmo et al. Taking down coordinated inauthentic behavior from Russia and China. *Meta Newsroom* (2022).
7. Nimmo et al. Quarterly adversarial threat report (Q2). *Meta Newsroom* (2022).
8. Meta. Quarterly adversarial threat report (Q3). *Meta Newsroom* (2022).
9. Nimmo et al. Quarterly adversarial threat report (Q4). *Meta Newsroom* (2023).
10. Meta. Quarterly adversarial threat report (Q1). *Meta Newsroom* (2022).
11. Twitter Safety. Disclosing networks of state-linked information operations we've removed. *Twitter Blog* (2020).

12. Twitter Safety. Disclosing networks of state-linked information operations. *Twitter Blog* (2021).
13. Etudo, U., Yoon, V. Y. & Yaraghi, N. From Facebook to the streets: Russian troll ads and black lives matter protests. In *Proceedings of the 52nd Hawaii International Conference on System Sciences* (2019).
14. Hurtado, S., Ray, P. & Marculescu, R. Bot detection in reddit political discussion. In *Proceedings of the Fourth International Workshop on Social Sensing* 30–35 (2019).
15. Zannettou, S. *et al.* Characterizing the use of images in state-sponsored information warfare operations by Russian trolls on Twitter. In *Proceedings of the International AAAI Conference on Web and Social Media* (2020).
16. Smith, S. T. *et al.* Automatic detection of influential actors in disinformation networks. *Proc. Natl. Acad. Sci.* **118**, e2011216118 (2021).
17. Alizadeh, M., Shapiro, J. N., Buntain, C. & Tucker, J. A. Content-based features predict social media influence operations. *Sci. Adv.* **6**, eabb5824 (2020).
18. Monti, F., Frasca, F., Eynard, D., Mannion, D. & Bronstein, M. M. Fake news detection on social media using geometric deep learning. *CoRR*arXiv:1902.06673 *(2019).*
19. Vargas, L., Emami, P. & Traynor, P. On the detection of disinformation campaign activity with network analysis. In *Proceedings of ACM SIGSAC Conference on Cloud Computing Security* (2020).
20. Li, C., Wang, L., Sun, S. & Xia, C. Identification of influential spreaders based on classified neighbors in real-world complex networks. *Appl. Math. Comput.* **320**, 512–523. https://doi.org/10.1016/j.amc.2017.10.001 (2018).
21. Asgharian Rezaei, A., Munoz, J., Jalili, M. & Khayyam, H. A machine learning-based approach for vital node identification in complex networks. *Expert Syst. Appl.* **214**, 119086. https://doi.org/10.1016/j.eswa.2022.119086 (2023).
22. Nimmo, B. & Hutchins, E. Phase-based tactical analysis of online operations. *Carnegie Endowment for International Peace* (2023).
23. Pols, P. The unified kill chain. *Fox-IT* (2017).
24. Sedova, K. *et al.* AI and the future of disinformation campaigns, part 1: The richdata framework. *Georgetown Center for Security and Emerging Technology* (2021).
25. Graphika & The Stanford Internet Observatory. Bad reputation. *Graphika Reports* (2022).
26. Giglietto, F., Righetti, N., Rossi, L. & Marino, G. It takes a village to manipulate the media: Coordinated link sharing behavior during 2018 and 2019 Italian elections. *Inf. Commun. Soc.* **23**, 867–891 (2020).
27. Facebook. Threat report: The state of influence operations 2017–2020. *Meta Newsroom* (2021).
28. Das Sarma, A. *et al.* Ranking mechanisms in twitter-like forums. In *Proceedings of the Third ACM WSDM* (2010).
29. Graphika. Deepfake it till you make it. *Graphika Reports* (2023).
30. Nimmo et al. Secondary infektion. *Graphika Reports* (2020).
31. Karras, T. *et al.* Analyzing and improving the image quality of stylegan. *CoRR*arXiv:1912.04958 *(2019).*
32. Perov, I. *et al.* Deepfacelab: A simple, flexible and extensible face swapping framework. *CoRR*arXiv:2005.05535 *(2020).*
33. OpenAI. Gpt-4 technical report. *CoRR*arXiv:2303.08774 *(2023).*
34. Ramesh, A. *et al.* Zero-shot text-to-image generation. *CoRR*arXiv:2102.12092 *(2021).*
35. Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. *CoRR*arXiv:1703.01365 *(2017).*
36. Reed, S. *et al.* A generalist agent. *Trans. Mach. Learn. Res.* (2022).
37. Yin, L. Smappnyu/urlexpander: Initial release (2018).
38. Yin, H., Yang, S., Song, X., Liu, W. & Li, J. Deep fusion of multimodal features for social media retweet time prediction. *World Wide Web* **24**, 1027–1044 (2021).
39. Pfeffer, J., Matter, D. & Sargsyan, A. The half-life of a tweet. *CoRR*arXiv:2302.09654 *(2023).*
40. Chami, I., Abu-El-Haija, S., Perozzi, B., Ré, C. & Murphy, K. Machine learning on graphs: A model and comprehensive taxonomy. *J. Mach. Learn. Res.* **23**, 1–64 (2022).
41. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. *CoRR*arXiv:1609.02907 *(2016).*
42. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. *Proc. Mach. Learn. Res.* **70**, 1263–1272 (2017).
43. Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y. & Bresson, X. Graph neural networks with learnable structural and positional representations. *CoRR*arXiv:2110.07875 *(2021).*
44. Grover, A. & Leskovec, J. node2vec: Scalable feature learning for networks. *CoRR*arXiv:1607.00653 *(2016).*

Acknowledgements

This material is based upon work supported by the U.S. Air Force Office of Scientific Research under award number FA9550-20-1-0382. N.J. is supported by U.S. Air Force Office of Scientific Research awards FA9550-20-1-0382 and FA9550-20-1-0383, and The John Templeton Foundation.

Author contributions

N.G., D.B., and N.J. conceived the study. N.G. collected data and performed analysis. D.B. and N.J. supervised. N.G. wrote the manuscript text. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-023-49676-z.

Correspondence and requests for materials should be addressed to N.A.G.