Linking Terrorist Network Structure to Lethality: Algorithms and Analysis of Al Qaeda and ISIS

Youdinghuan Chen, Chongyang Gao, Daveed Gartenstein-Ross, Kevin T. Greene, Karin Kalif, Sarit Kraus, Francesco Parisi, Chiara Pulice, Anja Subasic, and V. S. Subrahmanian

***Abstract*— Without measures of the lethality of terrorist networks, it is very difficult to assess if capturing or killing a terrorist is effective. We present the predictive lethality analysis of terrorist organization (PLATO) algorithm, which merges machine learning with techniques from graph theory and social network analysis to predict the number of attacks that a terrorist network will carry out based on a network structure alone. We show that PLATO is highly accurate on two novel datasets, which cover Al Qaeda (AQ) and the Islamic State (ISIS). Using both machine learning and statistical methods, we show that the most significant macrofeatures for predicting AQ's lethality are related to their public communications (PCs) and logistical subnetworks, while the leadership and operational subnetworks are most impactful for predicting ISISs lethality. Across both groups, the average degree and the diameters of the strongly connected components (SCCs) within these networks are strongly linked with lethality.**

***Index Terms*— Counterterrorism, machine learning, terrorism.**

I. Introduction

THOUGH terrorism has been a major concern since the 1970s, the events of 11 September 2001 saw the emergence of the "global war on terror." Since then, billions of dollars and thousands of lives of military personnel, and even more lives of civilians, have been expended in this war.

Counterterrorism efforts over the last 20 years have included a number of instruments aimed at targeting the lethality of terror networks. These include offering rewards for information leading to the capture and/or conviction of certain individuals, as well as operations to remove terrorists from their networks (e.g., the operation to capture Osama bin Laden).

When a terror network is targeted in this way, the network reshapes itself [1], [2], and so when considering the removal of a terrorist, such as Osama bin Laden from the Al Qaeda (AQ) network, it is important to be able to predict the lethality of the resulting network (after the removal). In this article, we propose methods to correlate the structure of a network with its lethality. While several definitions of lethality are possible, we define lethality as the estimated number of attacks carried out by a future terror network after reshaping. We develop a novel algorithm called predictive lethality analysis of terrorist organization (PLATO) for this purpose. When a counterterrorism agency considers removing a terrorist, it can use a system, such as shaping terrorist organizational network efficiency (STONE) [1], [2], to identify the new possible networks that result (and their associated probabilities) and then use a lethality model, such as the one proposed in this article, to identify the expected lethality of the resulting network.

The first effort at building such a lethality model was in [3] who proposed removing critical nodes in a network using a node centrality measure. But, they did not link networks to lethality, and because their data were cross sectional, they were unable to assess the impact of their method on lethality, as there were no "before removal" and "after removal" states of the network. Horowitz and Potter [4] find a correlation between the connections between groups and lethality, but also rely on cross-sectional data. Others have analyzed terror networks from the point of view of cell structure [5], their ability to communicate while remaining covert [6], [7], and how they can be destabilized [8]. STONE [1], [2] developed four simple lethality measures, which were tested on small datasets, along with sophisticated methods to identify who to remove from the network. To effectively forecast the lethality of terror groups and to evaluate how various policy interventions impact a terrorist network's lethality, we need features that involve the group's network structure, which are longitudinal (vary over time) and a model that can accurately map these features to group violence.

Our contributions fall into three categories. First, we leverage two novel longitudinal network datasets detailing the relationships between members of two prominent terror groups, AQ and the Islamic State (ISIS). Our AQ time-series network dataset consists of 16 years of data comprising 139 networks.

Manuscript received 11 January 2022; revised 29 June 2022; accepted 30 July 2022. This work was supported in part by the Office of Naval Research (ONR) under Grant N00014-16-1-2918. *(Corresponding author: V. S. Subrahmanian.)*

Youdinghuan Chen is with the Department of Biomedical Data Science, Dartmouth College, Hanover, NH 03755 USA (e-mail: youdinghuan.chen.gr@dartmouth.edu).

Chongyang Gao, Chiara Pulice, and V. S. Subrahmanian are with the Computer Science Department, Buffett Institute for Global Affairs, Northwestern University, Evanston, IL 60208 USA (e-mail: chongyanggao2026@u.northwestern.edu; chiara.pulice@northwestern.edu; vss@northwestern.edu).

Daveed Gartenstein-Ross is with Valens Global, Washington, DC 20003 USA (e-mail: daveed@valensglobal.com).

Kevin T. Greene is with the School of Public and International Affairs, Princeton University, Princeton, NJ 08544 USA (e-mail: kg2082@princeton.edu).

Karin Kalif and Sarit Kraus are with the Department of Computer Science, Bar-Ilan University, Ramat Gan 5290002, Israel (e-mail: karinsheri28@gmail.com; sarit@cs.biu.ac.il).

Francesco Parisi is with the Department of Computer Engineering, Modeling, Electronics, and Systems, University of Calabria, 87036 Rende, Italy (e-mail: fparisi@dimes.unical.it).

Anja Subasic is with the Computer Science Department, Dartmouth College, Hanover, NH 03755 USA (e-mail: anja.subasic.gr@dartmouth.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCSS.2022.3198820, provided by the authors.

Digital Object Identifier 10.1109/TCSS.2022.3198820

Our ISIS dataset consists of 49 networks spanning four years. To the best of our knowledge, these are the most extensive datasets showing the evolution of these two networks over time. We define a set of network-related features, some of which are used for the first time in terrorist network analysis. These features are based on the functional roles played by individual terrorists and the subnetworks induced by different functional roles using concepts from graph theory and social network analysis.

Second, we devise new algorithms to predictively link a terrorist network structure to future attacks. Our PLATO algorithm is an ensemble that uses a mix of regression methods, feature selection methods, and time lags to solve the following problem. Suppose a new network $\mathcal{N}_{t+1}$ comes into existence at the beginning of a time period I_{t+1}. We would like to predict the lethality of $\mathcal{N}_{t+1}$ as soon as it comes into existence. PLATO is initially invoked with a set of features, some of which are eliminated by the algorithm. Using past ground-truth data up to and before time period I_t, PLATO identifies the best parameters for an ensemble of regressors and uses late fusion to learn the optimal weights. Thus, PLATO uses a careful mix of feature engineering, subnetwork selection, feature selection, and regression, together with an ensemble model for generating predictions. We show that all three versions of PLATO significantly beat out a strong regression baseline in terms of predictive accuracy measured by Pearson correlation coefficient (PCC).

Third, we derive a new understanding of the link between the lethality of AQ and ISIS and their network structure. Because many features vary slightly, we introduce the concept of a macrofeature, which is an aggregation of similar features into one. We study which macrofeatures are most closely linked to AQ and ISIS's lethality. Surprisingly, the subnetwork of AQ involved in public communications (PCs) contributes to five of the most significant macrofeatures, while the logistical subnetwork of AQ is involved in four. In contrast, in the case of ISIS the top-ten most significant features are dominated by the leadership subnetwork (six of ten), followed by the operational subnetwork (two of ten). Moreover, in the case of AQ, the average degrees of nodes in various subnetworks are strongly linked with lethality, as are properties associated with the diameters of the strongly connected components (SCCs), as well as other centrality measures. This is also mirrored in the case of ISIS where degrees, centrality measures, and diameters of strongly connected subnetworks play important roles.

II. Network and Features

A terrorist network $\mathcal{N} = (V, E, \pi, \mathrm{cat})$ consists of four parts: a set V of nodes (terrorists), a set E of edges linking those nodes (relationships between terrorists), and two functions π and cat. The function cat assigns a set of functional categories to each terrorist. The set of possible functional categories is $\mathcal{C} = \{\texttt{operational}, \texttt{financial}, \texttt{logistical}, \texttt{R\&T}, \texttt{PC}, \texttt{leadership}\}$, where R&T and PCs stand for recruitment and training and public communications, respectively. For instance, during the March–August 2015 time period, Bana Fanaye, one of the leaders of Boko Haram, had two associated functional categories: logistical and leadership. The function π assigns a value for each property that a terrorist (node) might have. For instance, rank is a property on a 0 (lowest rank) to 10 (top leader) scale. Thus, we might have $\mathrm{rank}(\mathrm{Osama\ bin\ Laden}, 10)$, signifying that bin Laden had a value of 10 for the rank property in a given network. Nodes may have other properties, such as dead, jail, and alive (and free). Another important node property we consider is role (e.g., with values, such as trainer, bomb maker, and so on). Every node is required to have values for the rank and role properties, but other properties may or may not have associated values.

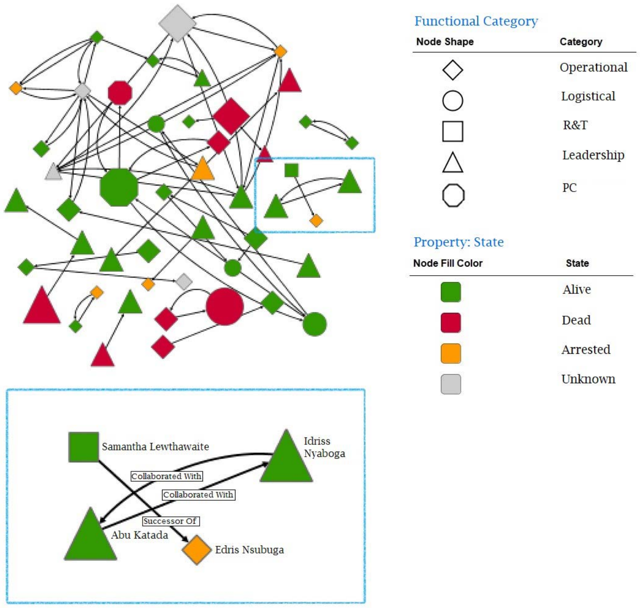

Fig. 1. AQ network during September–October 2014. This network has 48 nodes and 58 edges in total. The different node shapes denote the functional categories, and the colors denote the individual's state during that time frame. Node size is proportional to the individual's rank.

Formally, a terrorist's category of a terrorist can be included within π. We treat cat separately as our analysis focuses on these categories. The properties in π may be positively or negatively correlated with cat. A bomb maker (role) is rarely the leader (negative correlation), but is often in the operational category (positive correlation). This does not affect our analysis.

As terrorist networks evolve over time, our data for each of the AQ and ISIS networks have an associated set of T time intervals $I_0, I_1, \ldots, I_T$. We use $\mathcal{N}_t = \langle V_t, E_t, \pi_t, \mathrm{cat}_t \rangle$ to denote the snapshot of a network at the time interval I_t (with $t \in T$). An example of network from our dataset for the time interval September–October 2014 is shown in Fig. 1. For each network $\mathcal{N}_t$, which existed during the time interval I_t, we also collected A_t, the number of attacks carried out by the terror network during time I_t.

We built such tiime series of networks for AQ from November 2001 to January 2017 and for ISIS from December 2012 to January 2017. The ISIS dataset consists of 49 time-indexed networks, with an average time interval (i.e., the period of validity of time-indexed networks) of approximately

TABLE I
WbFs. Definitions 1–3 Can Be Found in Appendix A-A

Feature	Description	Example
Functional Category Fraction $\phi_1(t, f)$ (cf. Definition 1)	Fraction of people belonging to category f at time t	ϕ_1(*2015-03*,PC): percentage of people belonging to the functional category *Public Communications* in March 2015
Average Category Weight $\phi_2^W(t, f)$ (cf. Definition 2)	Average weight W of people in category f at time t	ϕ_2^r(*2015-03*,PC), where $r(v) = \pi(v, rank)$: average rank of people in category *Public Communications* in March 2015
Normalized Category Weight $\phi_3^W(t, f)$ (cf. Definition 3)	Fraction of total weight W of people with category f at time t over the total weight of people at time t	ϕ_3^{PR}(*2015-03*,PC), where $PR(v)$ is the PageRank of v: sum of the PageRanks of people belonging to category PC in March 2015 over the sum of PageRanks of all the people at the same time

one month. The average number of members per network is 74.17, with 10 as the min and 110 as the max. The number of relationships in the ISIS dataset ranged from 12 to 642 across the networks, with an average of 437.11. The AQ dataset consists of 139 time-indexed networks with an average time interval of between three and four months. The average number of members per network is 39.16, with a min of 2 and a max of 159. The average number of relationships per network was 113.90, with a minimum of 26 and a maximum of 808. We designed a codebook detailing the data collection procedures involving more than 20 social scientists to code the data. The data were collected from a variety of open-source materials in multiple languages, including English, Arabic, French, and Spanish. The coders mainly relied on primary sources, including jihadist propaganda and social media posts, publicly available intercepted documents produced by these organizations, and declassified intelligence reports. Examples of primary sources we relied upon include statements issued by the media arm of AQ in the Islamic Maghreb (Al-Andalus Media Foundation), first-hand accounts of troop movements in Idlib Governorate posted to Twitter in December 2015, and U.S. Department of State announcements of additions to its list of designated terrorists. The coders also consulted secondary sources, including books, journal articles, newspapers and news sites, and other publications. Coders factored in the credibility and possible biases of each source consulted and sought to corroborate all factual claims across multiple independent sources. The coding process included meetings to harmonize the data via discussion when coders coded data differently. Values for some missing properties were filled in using a set of assumptions. Missing start/end dates of a terrorist's membership were manually set to be consistent with the known relationships of the terrorists. If a relationship start date was unknown, we set the start date of the relationship to the start date of one of the two members who next joined the network. A similar adjustment was made for relationship end dates. We remove isolated vertices in the network. In total, the number of vertices and relationships filled in by such assumptions is 14.6% and 32.05%, respectively, for the ISIS and AQ datasets.

Features: We developed a feature vector for each network $\mathcal{N}_t$. We use fv(N_t, SF) to denote the feature vector extracted for N_t for a given set SF of features. We developed a total of 3738 features consisting of 534 basic features and 3204 time-lagged variants of the basic features. Because our data detail the category of each individual, and their rank and role within the network, we can construct a variety of novel features. We are able to treat terror groups, not as unitary actors, but rather as collections of individuals who exhibit considerable heterogeneity and dynamics over time. Our ability to construct a range of novel network features can help guide future work aiming to predict and understand the behavior of terror groups.

Our basic features include weight-based features (WbFs), restricted network-based features (RbFs), group-based features (GbFs), and cluster-based features (CbFs). WbFs assign a weight to each node (e.g., role, a centrality score, and a combination of rank and centrality score) and then aggregate them together for the network in different ways (e.g., mean/median of these node weights with respect to all nodes; mean/median/expected value of these node weights with respect to nodes belonging to a given functional category). RbFs look at the characteristics of the induced subnetwork that results when only certain functional categories are considered, e.g., operational. It includes features, such as the number of k-SCCs of size k or more with varying k, the average diameter of k-SCCs, the average standard deviation of k-SCCs, the density of various subnetworks induced by restricting to specific functional categories, and more. GbFs look at sets of nodes belonging to a given functional category (e.g., all operational nodes) and compute the group PageRank (GPR; which we define), group betweenness centrality (GBC) [9], and various combinations of such metrics. CbFs include global clustering coefficients for both the directed and undirected versions of the networks, different kinds of clustering coefficients obtained by focusing on specific subnetworks. Tables I–IV summarize the basic features that we introduced. Every row of one of these tables contains the name of a class of features, together with a brief description and an example of a concrete feature from that class. The formal definitions of the features can be found in Appendix A. For the sake of brevity, we use $\phi_i(t, \mathcal{F})$ with $i \in [1, 21]$ and $\mathcal{F} \subseteq \mathcal{C}$ to refer to the different kinds of features whose name and description are provided in the table. In most cases, $\mathcal{F}$ is a singleton—so we simply write $\phi_i(t, f)$ with $f \in \mathcal{C}$. Some features take an additional parameter, such as a natural number k, as input—so we may write $\phi_i(t, f, k)$. Time-lagged variants of these four types of basic features are also defined, because, for instance, the lethality of a network may depend upon the values of these basic features from past networks. Definitions can be found in Appendix A.

TABLE II

FEATURES BASED ON RESTRICTIONS OF THE NETWORK. DEFINITIONS 4–13 CAN BE FOUND IN APPENDIX A-B

Feature	Description	Example
k-Strongly Connected Components $\phi_4(t,f,k)$ (cf. Definition 4)	Number of Strongly Connected Components (SCCs) having size $\geq k$ for the restricted network $\mathcal{N}_t[f]$	ϕ_4(*2015-03*,PC,10): number of SCCs of size greater than or equal to 10 for the network restricted to category PC at time March 2015
Average k-SCC Diameter $\phi_5(t,f,k)$ (cf. Definition 5)	Average diameter of k-sized SCCs for the restricted network $\mathcal{N}_t[f]$	ϕ_5(*2015-03*,PC,10): average diameter of SCCs of size k w.r.t. the network restricted to category PC at time March 2015
Standard Deviation SCC Diameters $\phi_6(t,f)$ (cf. Definition 6)	Standard deviation of the diameters of SCCs for the network $\mathcal{N}_t[f]$	ϕ_6(*2015-03*,PC): standard deviation of the diameters of SCCs for the network restricted to category PC at time March 2015
Functional Sub-network Density $\phi_7(t,f)$ (cf. Definition 7)	Density of the restricted network $\mathcal{N}_t[f]$	ϕ_7(*2015-03*,PC): density (i.e., fraction of actual edges over potential ones) of the network restricted to category PC at time March 2015
Internally Biconnected Fraction $\phi_8(t,f)$ (cf. Definition 8)	Probability that a random vertex v in $\mathcal{N}_t[f]$ belongs to a triangle (v,u,w) s.t. u and w are connected to other vertices in $\mathcal{N}_t[f]$	ϕ_8(*2015-03*,PC): probability that a vertex v in the network restricted to PC at time March 2015 belongs to a triangle whose vertices different from v are connected to other vertices in the restricted network
Externally Biconnected Fraction $\phi_9(t,f)$ (cf. Definition 9)	Probability that a random vertex v in $\mathcal{N}_t[f]$ forms a pentagon involving two neighbors *outside* $\mathcal{N}_t[f]$	ϕ_9(*2015-03*,PC): probability that a vertex v in the network restricted to PC at time March 2015 forms to a pentagon having two vertices different from v and connected outside the restricted network
Functional In-Degree $\phi_{10}(t,f)$ (cf. Definition 10)	Average in-degree of vertices in $\mathcal{N}_t[f]$	ϕ_{10}(*2015-03*,PC): average in-degree of vertices belonging to the network restricted to category PC at time March 2015
Functional Out-Degree $\phi_{11}(t,f)$ (cf. Definition 11)	Average out-degree of vertices in $\mathcal{N}_t[f]$	ϕ_{11}(*2015-03*,PC): average out-degree of vertices belonging to the network restricted to category PC at time March 2015
Complementary Functional In-Degree $\phi_{12}(t,f)$ (cf. Definition 12)	Average in-degree of vertices in $\mathcal{N}_t[\mathcal{C}\setminus\{f\}]$ (i.e. the network restriction complementary to f)	ϕ_{12}(*2015-03*,PC): average in-degree of vertices belonging to the network restricted to all categories but PC in March 2015
Complementary Functional Out-Degree $\phi_{13}(t,f)$ (cf. Definition 13)	Average out-degree of vertices in $\mathcal{N}_t[\mathcal{C}\setminus\{f\}]$ (restriction complementary to f)	ϕ_{13}(*2015-03*,PC): average out-degree of vertices belonging network restricted to all categories but PC at time March 2015

TABLE III

GbFs. DEFINITIONS 14–17 CAN BE FOUND IN APPENDIX A-C

Feature	Description	Example
Group PageRank $\phi_{14}(t,f)$ (cf. Definition 14)	Group-page-rank (GPR) of the set of vertices in $\mathcal{N}_t$ whose functional category is f	ϕ_{14}(*2015-03*,PC): GPR of vertices with functional category *Public Communications* in the network in March 2015.
Group Betweenness Centrality $\phi_{15}(t,f)$ (cf. Definition 15)	Group Betweenness Centrality (GBC) of the set of of vertices in $\mathcal{N}_t$ whose functional category is f	ϕ_{15}(*2015-03*,PC): GBC, i.e. probability of traversing a vertex having functional category PC in the network concerning the time interval March 2015
Functional Rank GPR $\phi_{16}(t,\mathcal{F},r)$ (cf. Definition 16)	Average GPR of all sets of people with rank $\geq r$ and function category in $\mathcal{F}$ at time t	ϕ_{16}(*2015-03*,PC,10): average GPR of people ranked at least 10 and with category PC in March 2015
Functional Rank GBC $\phi_{17}(t,\mathcal{F},r)$ (cf. Definition 17)	Average GBC of all sets of people with rank $\geq r$ and function category in $\mathcal{F}$ at time t	ϕ_{17}(*2015-03*,PC,10): average GBC of people ranked at least 10 and with category PC in March 2015

TABLE IV

CbFs. DEFINITIONS 18–21 CAN BE FOUND IN APPENDIX A-D

Feature	Description	Example
Global Directed Clustering Coefficient $\phi_{18}(t,f)$ (cf. Definition 18)	Clustering coefficient of $\mathcal{N}_t[f]$	ϕ_{18}(*2015-03*,PC): clustering coefficient of the network restricted to category PC in March 2015
Global Undirected Clustering Coefficient $\phi_{19}(t,f)$ (cf. Definition 19)	Clustering coefficient of the undirected version of $\mathcal{N}_t[f]$	ϕ_{18}(*2015-03*,PC): clustering coefficient of the undirected version of network restricted to category *Public Communications* in March 2015
Average Group Functional Ranked CC $\phi_{20}(t,\mathcal{F},r)$ (cf. Definition 20)	Average of the cluster coefficients (CC) of all sets of r-ranked people whose category is in $\mathcal{F}$ at time t	ϕ_{20}(*2015-03*,PC,10): clustering coefficient of the sets of people whose rank is at least 10 and having category *Public Communications* in March 2015
Average Neighbor Functional Ranked CC $\phi_{21}(t,\mathcal{F},r)$ (cf. Definition 21)	Average of the cluster coefficients of the neighbors of r-ranked people whose category is in $\mathcal{F}$ at time t	ϕ_{21}(*2015-03*,PC,10): average clustering coefficient of the neighbors of people whose rank is at least 10 and having category PC in March 2015

III. PLATO ALGORITHM

Algorithm 1 shows our PLATO algorithm to predict the number of attacks that a terrorist group will carry out at the next time point. As mentioned earlier, our data consist of a time-series of networks $\mathcal{N}_1, \mathcal{N}_2, \ldots$ Each network $\mathcal{N}_j$ is in existence during an associated time interval I_j. Without loss of

Algorithm 1 PLATO With Ensemble and Late Fusion

```
Input: Set of features $SF$; Training dataset $TS_x = \{\langle fv(\mathcal{N}_1, SF), A_{1+x}\rangle, \dots, \langle fv(\mathcal{N}_{t-x}, SF), A_t\rangle\}$; New network $\mathcal{N}_{t+1}$; Ensemble of $n$ Regression Model types $\mathcal{E} = \{RM_1, \dots, RM_n\}$; Feature window size $w_F < t-x$, Training window size $w_T \le w_F$; Feature selection approach $FA$; Number of top features $k$.
Output: Estimated number of attacks $\widehat{A}_{t+1+x}$ for $\mathcal{N}_{t+1+x}$; PCC score.
1:  $groundTruth = \langle A_{1+x}, \dots, A_t\rangle$;
2:  $prediction[j] = \emptyset,\ result[j] = \perp, 1 \le j \le n$;
3:  $i = 0$;
4:  for $SW_i = \{\langle fv(\mathcal{N}_\ell, SF), A_{\ell+x}\rangle \mid \ell \in [i+1,\ i+w_F]\}$ do
5:    if $FA = IFS$ then
6:      $SF_{ij}$ = IterativeFeatureSearch($SW_i, \mathcal{E}, SF, w_T$);
7:    else
8:      $SF_{ij}$ = Select top-k features from $SF$ using $FA$ for $SW_i$;
9:    end if
10:   $TW = \{\langle fv(\mathcal{N}_l, SF), A_{l+x}\rangle \mid l \in [i+1+w_F-w_T,\ i+w_F]\}$
11:   for each regressor type $RM_j \in \mathcal{E}$ in parallel do
12:     $TW_{ij} = \{\langle fv(\mathcal{N}_l, SF_{ij}), A_{l+x}\rangle \mid \langle fv(\mathcal{N}_l, SF), A_{l+x}\rangle \in TW\}$
13:     $RM_j^*$ = Select best parameter setting for $RM_j$ on $TW_{ij}$;
14:     if $i + w_F < t - x$ then
15:       $result[j]$ = Apply $RM_j^*$ to $fv(\mathcal{N}_{i+w_F+1}, SF_{ij})$;
16:       $prediction[j] = prediction[j] \oplus_j result[j]$;
17:       $i = i + 1$;
18:     else
19:       $result[j]$ = Apply $RM_j^*$ to $fv(\mathcal{N}_{t+1}, SF_i)$;
20:       break;
21:     end if
22:   end for
23: end for
24: $W = \underset{W s.t. \Sigma_{i=1}^{n} W[i]=1}{\text{argmax}} PCC(\Sigma_{j=1}^{n} W[j] \cdot prediction[j], groundTruth)$;
25: $score = PCC(\Sigma_{j=1}^{n} W[j] \cdot prediction[j], groundTruth)$;
26: $result_{lf} = \Sigma_{j=1}^{n} W[j] \cdot result[j]$;
27: return $result_{lf}, score$
```

generality, we assume that the interval I_j precedes the interval I_{j+1} for all j.

Suppose the network in existence now is network $\mathcal{N}_t$; i.e., networks $\mathcal{N}_1, \dots, \mathcal{N}_{t-1}$ are from the past, and the current network is $\mathcal{N}_t$. A new network $\mathcal{N}_{t+1}$ comes into effect when a change occurs (e.g., a terrorist is captured or killed, some relationships between terrorists change, we have information about some new terrorists, and so on). The idea is that PLATO will be used to predict the lethality of a new network $\mathcal{N}_{t+1}$ as soon as the new network comes into being. For instance, the AQ network N_i changed on May 2011 when Osama bin Laden was killed. PLATO could be applied immediately on the new network N_{i+1} resulting from the change of Osama bin Laden's status from "alive (and free)" to "dead." Though the nodes and edges in this new network may be the same as in the previous one, it is considered different, because a property of one node (bin Laden) has changed. The ability to produce a new predictions when there are structural changes in the network, new edges or nodes are added, or simply a change in roles within the network, and a lower ranked leader is promoted, opens new possibilities for making more dynamic forecasts of terror group lethality.

Suppose the networks in $\{\mathcal{N}_1, \dots, N_t\}$ are known, $\mathcal{N}_{t+1}$ is the new network, and we are interested in estimating A_{t+1+x}, the number of attacks in the future network N_{t+1+x}. PLATO takes a training set $\text{TS}_x = \{\langle \text{fv}(\mathcal{N}_1, \text{SF}), A_{1+x}\rangle, \dots, \langle \text{fv}(\mathcal{N}_{t-x}, \text{SF}), A_t\rangle\}$ consisting of feature vectors of the first $t - x$ networks we have, along with the corresponding numbers of attacks carried out $1 + x$ networks in the future (but never going beyond network N_t), and tries to predict how many attacks the network $\mathcal{N}_{t+1+x}$ will carry out. One challenge in making this prediction is that we do not know how many and which past networks in $\{\mathcal{N}_1, \dots, N_t\}$ should be considered. The reason is that we do not know which of the networks in $\{\mathcal{N}_1, \dots, N_t\}$ provides an important signal for predicting $\mathcal{N}_{t+1+x}$. For instance, does the number of attacks carried out by $\mathcal{N}_{t+1}$ depend on just $\mathcal{N}_t$? On just $N_{t-2}, \mathcal{N}_{t-1}$ and $\mathcal{N}_t$? Additionally, we do not know which subset of features are relevant.

PLATO handles these challenges using a sliding window (Line 4) of w_F networks from the training set TS_x in each iteration of the main **for** loop (Lines 4–23). For each sliding window SW_i, it iteratively selects relevant features (Lines 5–9). Feature selection can be done using any method—in our experiments, we consider principal component analysis (PCA) and mutual information (MI) as well as a feature selection approach that we defined called iterative feature search (IFS) (Line 6). The best features for each regressor-type RM_j are stored in the set SF_{ij}. In the case of PCA and MI, these sets of features are the same for each regressor. In the case of IFS, they can change as IFS selects the best features by considering the regression model type used. More details on IFS are provided at the end of this section.

PLATO then creates a training set TW consisting of the feature vectors/number of attacks of the last w_T networks from TS_x in SW_i (Line 10). In Lines 11–22, PLATO trains each regression model type [e.g., lasso versus ridge versus support vector regression (SVR)] in parallel and does hyperparameter optimization to create the best regressors RM_j^* for each regressor type (Line 13). Each regression model type in the ensemble is trained using the feature vectors TW_{ij} (i.e., the training set restricted to the selected features). If the sliding window is not the last, i.e., there are still networks that have not been considered in the training dataset, the regression model RM_j^* is used to predict the number of attacks A_{i+w_F+1+x} using the network following the sliding window (Line 15), and the prediction list is updated with this new prediction (Line 16). Otherwise, RM_j^* is applied to the test network $\mathcal{N}_{t+1}$,

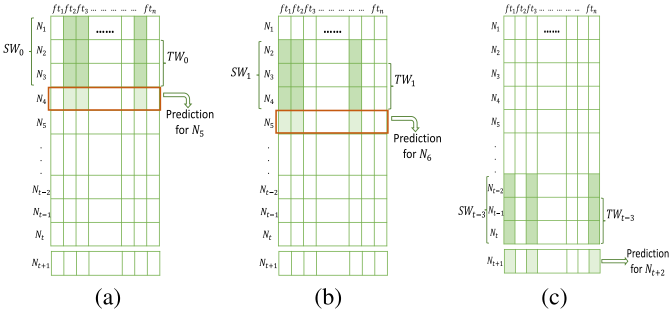

Fig. 2. Execution of PLATO. (a) First iteration. (b) Second iteration. (c) Last iteration.

thus making a prediction of A_{t+1+x} (Line 19). In particular, in Line 16, the best jth regression model type generates a prediction, and the $\oplus_j$ operator concatenates the vector of results (prediction vector) generated by the jth regression model type. The $\oplus_j$ reflects this concatenation operator.

When we get to Line 24, the situation is as follows: the best regressor RM_j^* for each regressor-type RM_j has made the same number of predictions, each stored in prediction[j] (whose size is the same as that of groundTruth). PLATO tries to find an assignment of weights W (such that the weights sum to 1) for the regressors in the ensemble, so that PCC of the linear combination of the predictions made by the ensemble and the ground truth is the highest possible. The weights are discovered using a grid search that considers all combinations of weights in the increments of 0.2. PLATO eventually returns the PCC corresponding to the best weight assignment (Line 25) along with the predicted number of attacks $\widehat{A}_{t+1+x}$ for network N_{t+1+x} (Line 26).

Fig. 2 illustrates how PLATO works when $w_F = 3$, $w_T = 2$, and $x = 1$, and the ensemble consists of just one regression model, showing the first two iterations of the algorithm [Fig. 2(a) and (b)] and the last one [Fig. 2(c)] on a toy example. At each iteration, a subset SF_{ij} of features (highlighted in green) is selected and used by the regression model to make a prediction of the number of attacks carried out one time point ahead.

Selecting Features via Iterative Search: Computing the set of best features is an integral part of PLATO. To do this, we defined a bottom-up greedy algorithm called iterated feature search (Function 1) that iteratively selects features as long as predictive accuracy increases. Intuitively, a set of features are good if they allow us to accurately predict the number of attacks occurring in the future. To this end, Function 1 considers the regression model type during its execution and, for each model RM_j, keeps track (via the BestSet[j] vector) of the features that allow RM_j to generate the highest PCC score. The latter is found upon calling the findPath subroutine, which takes as input, a set of features SF, a regression model-type RM_j, a set of features SF_b, and PCC Score obtained by RM_j using the features in SF_b, and then extends SF_b as much as possible (i.e., till Score increases). Initially, the set of features is empty, and the score is set to 0 (Line 3). The set is then updated by exploring all unexplored features in SF and adding to it the feature f_b that, together with those in SF_b, generates the highest score (Line 7). The score is then updated according to the predictionScore (Line 8) procedure. The process is continued until no more features can be added to SF_b (Lines 9–12).

IFS relies on the predictionScore subroutine to detect the most relevant features. This procedure takes the regression model type and a set of features as input and returns the average PCC that is computed as follows: the set TS is split in a training window TW of size w_T (Line 21) and a test set TestNtw of size w_F (Line 24). The subroutine trains the regression model RM on the feature vectors TW_i (restricted to the features in SF_b) and uses the resulting model RM^* to make predictions on the restricted test set (Line 24). The PCC of the predicted values and the ground truth is then computed (Line 26). The average of these scores, obtained by repeating the abovementioned operations for all consecutive training windows of size w_T in TS, is eventually returned (Line 29).

IV. Experimental Evaluation

This section reports on our experimental assessment of PLATO's performance. We also show that PLATO identifies features that are important for the prediction. We conclude this section by discussing results on predicting the density of attacks, instead of number of attacks and extending the ensemble of regression models used by PLATO.

We analyzed the impact of various parameters on the performance of PLATO: the feature window size ($w_F = \{3, 4, 5, 10, 15, 20, 25\}$), the training window size ($w_T = \{3, 4, 5, 6, 10, 15, 20, 25\}$), the number of features to be selected ($k = \{10, 20, 30, 40, 50\}$), the temporal offset ($x = \{0, 1, 2, 3, 4, 5\}$), three feature selection approaches (MI, PCA, and our IFS), and an ensemble of six regression models: ridge [10], lasso [11], random forest [12], linear, polynomial, and radial basis function (RBF) SVR [13][1].

Because classic k-fold cross validation may end up using networks from the future (in training folds) to predict the number of attacks for networks in the past (in the test fold), we used a standard rolling window technique that ensures that networks in the test data always occur after the networks in the training data. The baseline, named BAS, splits the data into a training set containing the first 80% of the time-indexed networks and a test set with the last 20%. Predictions are made for the last 20% of the data, and a PCC is calculated using these predictions. Given a feature selection method X (either MI or PCA), our baseline results BAS used four well-known regression models (lasso, SVR, ridge, and random forest) using X. The result reported by BAS is the best result obtained by running these eight models in conjunction with X—hence, this is a strong baseline. The only past work linking network structure to lethality [1], [2] used a very small number of features (already included in our BAS baseline) and used a very simple linear regressor. BAS already does more than this past work and, furthermore, augments it with eight models.

Table V shows the PCC score of each approach using various feature selection methods. For readability, for each

[1] All experiments were run on a Linux cluster of Intel Xeon nodes with RAM ranging from 16 to 64 GB. All the algorithms were written in Python.

TABLE V
BEST PCC SCORES FOR BASELINE BAS AND PLATO

	AL QAEDA						ISIS					
	$x=0$	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$	$x=0$	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$
BAS[MI]	-0.250	-0.282	-0.312	0.341	0.390	0.375	0.402	-0.382	-0.248	-0.235	0.502	0.508
BAS[PCA]	-0.278	-0.353	-0.335	0.262	0.305	-0.249	0.140	-0.576	-0.308	-0.357	-0.368	0.230
PLATO[MI]	**0.694**	**0.690**	**0.697**	0.701	**0.719**	0.691	**0.686**	**0.623**	**0.574**	**0.523**	**0.561**	**0.667**
PLATO[PCA]	0.689	0.682	0.689	**0.710**	0.687	**0.697**	0.657	0.611	0.570	0.517	0.516	0.529
PLATO[IFS]	0.659	0.659	0.671	0.643	0.646	0.689	0.583	0.54	0.521	0.515	0.486	0.453

Function 1 Iterative Feature Search

```
Input: Training dataset $TS_x = \{\langle fv(\mathcal{N}_1, SF), A_{1+x}\rangle, \ldots, \langle fv(\mathcal{N}_{w_F}, SF), A_{w_F+x}\rangle\}$; Ensemble of $n$ Regression Model types $\mathcal{E} = \{RM_1, \ldots, RM_n\}$; Set of features $SF$; Sliding window size $w_T$.
Output: Top features $BestSet[j] \subseteq SF$ for each $RM_j \in \mathcal{E}$.
$BestSet[j] \leftarrow \emptyset$, $BestScore[j] = 0$, $1 \leq j \leq n$
for each regressor type $RM_j \in \mathcal{E}$ in parallel do
    findPath($SF, RM_j, BestSet[j], BestScore[j]$);
end for
return $BestSet$
procedure findPath($SF, RM_j, SF_b, Score$)
$f_b = \underset{f \in SF \setminus SF_b}{\text{argmax}}$ predictionScore($RM_j, SF_b \cup f$)
$CurrentScore$ = predictionScore($RM, SF_b \cup f_b$);
if $CurrentScore > Score$ then
    $SF_b = SF_b \cup f_b$
    $Score = CurrentScore$
    findPath($RM_j, SF, SF_b, Score$)
else if $Score > BestScore[j]$ then
    $BestSet[j] \leftarrow SF_b$
    $BestScore[j] = Score$
end if
$SF \leftarrow SF \setminus \{f_b\}$
procedure predictionScore($RM, SF_b$)
$avgScore = 0$;
$i = 0$;
for $TW = \{\langle fv(\mathcal{N}_l, SF), A_{l+x}\rangle \mid l \in [i+1, i+w_T]\} \subseteq TS_x$ do
    $TW_i = \{\langle fv(\mathcal{N}_l, SF_b), A_{l+x}\rangle \mid \langle fv(\mathcal{N}_l, SF), A_{l+x}\rangle \in TW\}$
    $RM^*$ = Select best parameter setting for $RM$ on $TW_i$;
    $TestNtw = \{\langle fv(\mathcal{N}_l, SF), A_{l+x}\rangle \mid l \in [i+w_T+1, i+w_F]\}$
    $predictions$ = Apply $RM^*$ to $\{fv(TestNtw, SF_b)\}$;
    $avgScore = avgScore$ + PCC($predictions, groundTruth$);
    $i = i + 1$;
end for
return $avgScore/i$
```

TABLE VI
PAIRED t-TEST COMPARISONS OF PLATO VERSUS BASELINE

Comparison	Mean Difference (PLATO - BAS)	99%CI, lower	99%CI, upper	P-Value
AL QAEDA, PCA	0.759	0.637	0.881	1.08E-16
AL QAEDA, MI	0.681	0.546	0.817	2.56E-14
ISIS, PCA	0.666	0.546	0.786	2.07E-15
ISIS, MI	0.664	0.515	0.813	5.48E-13

dataset and temporal offset, we highlight the best PCC score in red. PLATO[MI] and PLATO[PCA] have the best scores, with the former outperforming the latter by a negligible amount. PLATO[IFS] obtains comparable scores as well. However, we note that both PLATO[PCA] and PLATO[MI] are faster than PLATO[IFS] (one to two days versus two to three weeks). To compare PLATO with the baseline in an unbiased manner, we implemented two-tailed paired Students t-tests and found that PLATO has significantly higher Pearson coefficients than the baseline across groups, temporal offsets, and feature-selection methods (all mean differences >0 and $P < 1.0e-12$, see Table VI).

A. Statistical Analysis

In this section, we provide an analysis showing that PLATO identifies features that are important for prediction. For both PLATO[PCA] and PLATO[MI] and for each temporal offset $x \in [0, 5]$, we selected the 20 most relevant features by counting the number of times they were selected across all training windows and ranked them from 1 to 20. That is, every time a feature was selected as being a relevant feature in a training window, we increased the count of that feature. Hence, for each dataset (either AQ or ISIS) and for each feature selection approach (either PCA or MI), we obtained six lists of most relevant features (one for each value of x). A total of 24 lists of features was, thus, obtained, each consisting of 20 features. We then introduced the notion of rank and occurrence to measure the frequency of a given feature with respect to the different values of the temporal offset x. In particular, rank is the average of the six individual ranks obtained for each value of x, while occurrence is the percentage of times that a given feature occurs over the different temporal offsets (e.g., a feature has occurrence equal to 100% if it is in the list of the top-20 features for each value of x).

Table VII reports the rank and occurrence of the top features with respect to the two datasets for PLATO[PCA]. It turned out that, in the case of PLATO[PCA], for all values of x, all the top ranked features rely on one SCC, regardless of the dataset used. Moreover, Table VII shows that most of these features involve the category `operational`, followed by

TABLE VII
OCCURRENCE (OCC.) AND RANK OF THE TOP-FEATURES USED BY PLATO[PCA]

	Al Qaeda		ISIS	
Features	Rank	Occ.	Rank	Occ.
F1: 1-SCCs for networks restricted to category `operational`	4.5	100%	3.5	100%
F2: 1-SCCs for networks restricted to category `leadership`	15.5	100%	6	100%
F3: 1-SCCs for networks restricted to category `logistical`	-	-	-	-
F4: 1-SCCs for *alive* members of networks restricted to category `operational`	9.3	100%	11.8	100%
F5: 1-SCCs for *not jailed* members of networks restricted to category `operational`	-	-	-	-
F6: 1-SCCs for *not jailed* members of networks restricted to category `leadership`	18	33%	19	17%
F7: 1-SCCs for *alive* members of networks restricted to category `leadership`	-	-	-	-
F8: 3-time-points lagged feature value of F1	1.2	100%	3	100%
F9: 2-time-points lagged feature value of F1	1.8	100%	1.2	100%
F10: 1-time-points lagged feature value of F1	3.5	100%	2.3	100%
F11: 3-time-points lagged feature value of F2	15.3	100%	5.3	100%
F12: 2-time-points lagged feature value of F2	14.8	100%	7.2	100%
F13: 1-time-point lagged feature value of F2	17.5	100%	8.5	100%
F14: 3-time-points lagged feature value of F3	16	33%	-	-
F15: 2-time-points lagged feature value of F3	20	33%	-	-
F16: 3-time-points lagged feature value of F4	5	100%	17.2	100%
F17: 2-time-points lagged feature value of F4	6.5	100%	10.3	100%
F18: 1-time-point lagged feature value of F4	9.5	100%	10.8	100%
F19: 3-time-points lagged feature value of F5	8.7	100%	-	-
F20: 2-time-points lagged feature value of F5	6.3	100%	16.6	83%
F21: 1-time-point lagged feature value of F5	11	100%	15.3	67%
F22: 3-time-points lagged feature value of F6	19	33%	16	83%
F23: 2-time-points lagged feature value of F6	-	-	18	50%
F24: 3-time-points lagged feature value of F7	19	17%	13.5	83%
F25: 2-time-points lagged feature value of F7	-	-	18.7	100%
F26: 1-time-point lagged feature value of F7	-	-	19	17%
F27: 3-time-points lagged average value of F1	20	33%	15	33%
F28: 2-time-points lagged average value of F1	16	67%	13.8	83%
F29: 1-time-point lagged average value of F1	12	67%	8.5	100%
F30: 1-time-point lagged average value of F2	-	-	18	83%
F31: 1-time-point lagged average value of F4	19.3	67%	-	-
F32: 1-time-point lagged average value of F5	19	17%	-	-

`leadership`. In fact, both datasets share the same top-four features—to ensure readability, the ranks of first-, second-, third-, and fourth-ranked features are highlighted in green, yellow, orange, and red, respectively—and, since these features involve the category `operational` and occurrence is 100%, it means that `operational` is used for all values of the temporal offset x.

1) Macrofeatures: In the case of PLATO[MI], the "top" features look very heterogeneous. But, if we disregard the time lag, the "top" features become more homogeneous, which suggests that the same features are in play, but at different time lags. Therefore, to analyze the features selected by PLATO[MI], which according to Table V performs better than the other approaches, we grouped the most relevant features into macrofeatures. Two features were considered to be in the same group if they only differ in the time lag or if they only distinguish between features using different properties of nodes (e.g., alive and jail). Specifically, let $\phi^\pi(t, f)$ be a feature evaluated on the network $\mathcal{N}_t$ restricted to the functional category f and to the nodes, such that property π is true (where $\pi \in \{\text{alive, jail, free}\}$). Let $\Psi^\pi(t, f, \tau)$ be the τ-lagged variant of a $\phi^\pi(t, f)$; the lagged variant uses the information provided by the network $\mathcal{N}_{t-\tau}$; i.e., its value is equal to $\phi^\pi(t - \tau, f)$. We, thus, collapsed all time-lagged features $\Psi^\pi(t, f, \tau)$, for $\tau \in \{1, 2, 3\}$ and $\pi \in \{\text{alive, jail, free}\}$, of a given feature into a single macrofeature $\Phi(t, f)$. We obtain 66 macrofeatures for AQ and ISIS using PLATO[MI]. The original features from which a macrofeature is obtained are said to be compatible with the macrofeature [e.g., $\phi^\pi(t, f)$ is compatible with $\Phi(t, f)$].

Given a dataset and the six lists of top-20 features (one for each value of x) for that dataset, we measure the importance of macrofeatures as follows. For each macrofeature mf and rank $r \in [1, 20]$, let $\mathrm{mf}(r)$ be the percentage of lists out of six in which mf is compatible with a top-h feature in the list, with $h \leq r$. This means that if a macrofeature mf is compatible with every top-1 feature in all the lists of most relevant features for a dataset, then $\mathrm{mf}(r) = 100\%$ for each $r \in [1, 20]$. Then, the importance of mf is given by the integral of $\mathrm{mf}(r)$ between $r = 1$ and $r = 20$, that is, the area under the cumulative percentage $\mathrm{mf}(r)$.

Table VIII reports the top-ten highest ranked macrofeatures for AQ and ISIS, respectively. For each dataset, they are ranked from the most important to the least important. Due to space constraints, we report the complete list of the highest-ranked macrofeatures in the Supplemental Material. The subnetworks of AQ involved in logistical support and PCs are consistently among the most predictive macrofeatures. As with the PLATO[PCA] results, the leadership and operational subnetworks are most important for predicting future ISIS violence. Moreover, in the case of AQ, the average degrees of nodes in various subnetworks are strongly linked with lethality, as are properties associated with the diameters of the SCCs, as well as other centrality measures. This is also mirrored in the case of ISIS where degrees and diameters of strongly connected subnetworks play an important role.

Due to space limitations, additional analytic results we performed are reported in the Supplemental Material.

B. Density Prediction and Ensemble Extension

As different networks can last for different time periods, it is of interest to predict the density of attacks, i.e., the number of attacks per month when the network is valid. Predicting density is complementary to predicting the number of attacks and is of interest to analysts who cannot estimate the duration of a network. We used PLATO to predict the number of attacks

TABLE VIII
MACROFEATURES FOR PLATO[MI] FOR AQ (LEFT) AND ISIS (RIGHT) RANKED BY IMPORTANCE

	AL QAEDA	**ISIS**
1	Average 1-SCC Diameter for networks restricted to category logistical	Group PageRank for networks restricted to functional category leadership
2	Functional Sub-network Density for networks restricted to logistical	Functional Sub-network Density for networks restricted to leadership
3	Functional In-Degree for networks restricted to functional category PC	1-Strongly Connected Components for networks restricted to leadership
4	Functional In-Degree for networks restricted to category financial	Complementary Functional In-Degree for networks restricted to leadership
5	Functional In-Degree for networks restricted to category logistical	Functional Out-Degree for networks restricted to category leadership
6	Functional Out-Degree for networks restricted to functional category PC	Complementary Functional Out-Degree for networks restricted to leadership
7	Group Betweenness Centrality for networks restricted to category logistical	1-Strongly Connected Components for networks restricted to operational
8	Average 1-SCC Diameter for networks restricted to functional category PC	Functional In-Degree for networks restricted to category leadership
9	Standard Deviation SCC Diameters for networks restricted to category PC	Functional In-Degree for networks restricted to category operational
10	Group Betweenness Centrality for networks restricted to category PC	Normalized Category Rank for networks restricted to category leadership

TABLE IX
BEST PCC SCORES FOR PLATO PREDICTING DENSITY AND FOR PLATO WITH GCN IN THE ENSEMBLE

		AL QAEDA						**ISIS**					
		$x = 0$	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 0$	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$
Density prediction (standard ensemble)	**PLATO[MI]**	0.738	0.730	0.730	0.745	0.726	0.732	Density prediction coincides with attack prediction.					
	PLATO[PCA]	0.743	0.749	0.716	0.734	0.712	0.704						
Density prediction (ensemble with GCN)	**PLATO[MI]**	0.743	0.731	0.736	0.745	0.726	0.740						
	PLATO[PCA]	0.743	0.749	0.716	0.734	0.720	0.704						
Number of attacks (ensemble with GCN)	**PLATO[MI]**	0.707	0.716	0.701	0.701	0.719	0.702	0.692	0.623	0.608	0.540	0.561	0.667
	PLATO[PCA]	0.689	0.682	0.689	0.710	0.687	0.697	0.661	0.614	0.573	0.546	0.522	0.531

per month, as time intervals of our networks are multiple of one month. This makes no difference for the prediction for ISIS, as the duration of each network in ISIS is about one month. For AQ, the PCC scores of the PLATO[MI] and PLATO[PCA] variant predicting the density are shown in the third and fourth rows of Table IX, respectively. Density predictions by PLATO are more accurate, improving the PCC scores of 3.4% and 4.1% on average as opposed to predicting the raw number of attacks (see Table VI).

Finally, we analyzed the impact of augmenting the PLATO ensemble (consisting of six regression models) with a graph convolution network (GCN) approach [14] appropriate for our network-based data. Although using advanced GCNs can avoid using application-specific features, the results shown in Table IX (rows from fifth to eighth) show that adding GCN to the ensemble leads to a very small improvement that is not statistically significant (all paired Students t-tests give p-values > 0.13 with 95% CI) for both ISIS and AQ for attacks or density prediction.

V. LIMITATIONS AND FUTURE WORK

While our AQ and ISIS data are among the first longitudinal datasets on these terror groups, we (like most researchers) are limited, because much data on these groups are classified. Though strong efforts were made to harmonize differences in the data collected by different coders, we do make assumptions on some missing data as detailed in Section II. Measuring lethality as the number of attacks as we have done is valid, but also a limitation. Measuring it via other metrics (e.g., number of casualties and economic damage) offers possible avenues for future work.

Finally, embedding lethality computations into algorithms for reshaping terror networks need to be studied further. Past work suffered from being unable to measure the dependent variable, i.e., efficacy of reshaping efforts [1], [2]. These efforts may also be aided by parallel studies on how to shape corporate board networks where data (e.g., when a person joined or left a board) are more readily available and where related dependent variables (e.g., share price) are also publicly available.

VI. CONCLUSION

Our work contributes to the growing body of research on forecasting political violence [15], [16], [17], [18]. Our model, based on network features within terror groups, adds to the existing research, which has found that including network information about violent groups improves predictive performance [19], [20]. We extend these efforts by leveraging novel data about not only the nodes and edges within AQ and ISISnetworks, but also node attributes (e.g., the rank and role of individuals) and category types (e.g., logistical and operational). Our ability to accurately predict the lethality of AQ and ISIS suggests that there may be future gains to be made by collecting and leveraging data on such networks.

Additional results (included in the Supplemental Material) also suggest potential policy implications. For the top-ranked features for PLATO[MI] we assessed the relationship between the number of attacks via bivariate Poisson regression accounting for the robust variance and multiple hypothesis testing (through the Bonferroni correction). For AQ, we find that the coefficient for the features related to the average degree of the PC subnetworks is generally statistically significant and positive. Put another way, AQ's PCs network becoming more connected is associated with AQ carrying out more violence. This might explain why previous work does not find an association between propaganda output and the number of attacks carried out [21]. Previous works have not considered the impact of the members of the PCs network, so it is possible scholars have overlooked important factors. For instance, it may be that changes in the connections between the members responsible for propaganda influences group effectiveness, possibly having a lagged impact. However, more work is needed to more fully assess the causality of this relationship.

Appendix A
Definitions of Features

We first provide the definitions of the basic features listed in Tables I–IV and then discuss, in more detail, the time-lagged features. Recall that $\mathcal{N}_t = \langle V_t, E_t, \pi_t, \mathrm{cat}_t \rangle$ denotes a network existing during the time interval I_t.

A. Weighted-Based Features

The first class of features provides insights about the fraction of people at time t belonging to the subnetwork associated with a specific functional category f (e.g., `operational`).

Definition 1 (Functional Category Fraction):

$$\phi_1(t, f) = \frac{|\{v \mid v \in V_t,\ \ f \in \mathrm{cat}_t(v)\ \}|}{|V_t|}.$$

Next, we define two classes of features, which are parametrized by a weight function $W : \mathcal{V} \to \mathbb{N}$. For instance, we may choose $W(v)$ to be the vertex rank by defining $W(v)$ equal to $\pi(v, \mathrm{rank})$ for each $v \in V$.

The first family of features consists of the average of the weights $W(v)$ of people v in category f in the network $\mathcal{N}_t$.

Definition 2 (Average Category Weight):

$$\phi_2^W(t, f) = \frac{\sum_{v \mid v \in V_t,\ \ f \in \mathrm{cat}_t(v)} \ W(v)}{|\{v \mid v \in V_t,\ \ f \in \mathrm{cat}_t(v)\}|}.$$

The second class of features is the sum of the weights $W(v)$ of people in category f divided by the sum of the weights of all people in the network at time t.

Definition 3 (Normalized Category Weight):

$$\phi_3^W(t, f) = \frac{\sum_{v \mid v \in V_t,\ \ f \in \mathrm{cat}_t(v)} \ W(v)}{\sum_{v \mid v \in V_t} W(v)}.$$

Thus, if function W returns the rank, i.e., W is the function $r(v) = \pi(v, \mathrm{rank})$, we obtain $\phi_2^r(t, f)$, which is the average rank of people in category f in the network $\mathcal{N}_t$. Likewise, $\phi_3^r(t, f)$ is the sum of the rank of people in category f divided by the sum of the ranks of all people in $\mathcal{N}_t$.

The next two features are obtained by doing the same as mentioned earlier with PageRank [22] instead of rank. Let $\mathrm{PR}(v)$ be the PageRank of vertex v. Then, $\phi_2^{\mathrm{PR}}(t, f)$ is the average PageRank of people in category f in the network at time t, while $\phi_3^{\mathrm{PR}}(t, f)$ is the sum of the PageRanks of people in category f divided by the sum of the PageRanks of all people in the network at time t.

Similarly, we define $\phi_2^g(t, f)$ and $\phi_3^g(t, f)$ where the weight function $g(v) = \mathrm{PR}(v) \cdot r(v)$ is used.

In summary, using $\phi_2^W(t, f)$ and $\phi_3^W(t, f)$, we defined six features, three for each group, where the weight function $W(\cdot)$ is one of the following: rank $r(\cdot)$, PageRank $\mathrm{PR}(\cdot)$, and the product $g(\cdot)$ of rank and PageRank.

B. Features Based on Restrictions of the Network (RbF)

We use the concept of restriction of a network with respect to a functional category to define features. The restriction of $\mathcal{N}$ with respect to $f \in \mathcal{C}$, denoted as $\mathcal{N}_t[f]$, is the subnetwork induced by the nodes whose functional categories include f; functions π and cat are then restricted to nodes in $\mathcal{N}_t[f]$.

We start by defining features measuring the diameter of our networks at different time points, possibly restricting the network to some features. However, the diameter of a graph is infinite if it is not strongly connected, and this may happen for the kinds of networks we are dealing with. Thus, to define features returning finite values, we will consider SCCs of different sizes.

Given a network $\mathcal{N}$, we use $\mathrm{SCC}(\mathcal{N})$ to denote the set of SCCs in $\mathcal{N}$. Moreover, we use $\mathrm{SCC}(\mathcal{N}, k)$ to denote the set of SCCs of size k (i.e., containing k vertices). Given $\mathcal{N} = \langle V, E, \pi, \mathrm{cat} \rangle$, we say that $\mathcal{N}$ is strongly connected if $\mathrm{SCC}(\mathcal{N}, |V|)$ is $\mathcal{N}$ itself.

Definition 4 (k-SCCs):

$$\phi_4(t, f, k) = \sum_{k' \in [k, |V_t[f]|]} \left|\mathrm{SCC}(\mathcal{N}_t[f], k')\right|.$$

Feature $\phi_4(t, f, k)$ is the number of SCCs of $\mathcal{N}_t[f]$ of size greater than or equal to k. In the following, we use $k_{\max}$ to denote the size of the largest SCCs of $\mathcal{N}_t[f]$. Thus, $\phi_4(t, f, k) > 0$ for $k \leq k_{\max}$; 0 otherwise. Moreover, if $k_{\max} = |V_t[f]|$, then $\phi_4(t, f, k) = 1$—the unique largest connected component is $\mathcal{N}_t[f]$ itself.

Let $d_{\mathcal{N}}(u, v)$ be the shortest distance (i.e., number of edge hops) between vertices u and v in a network $\mathcal{N}$. The diameter of $\mathcal{N} = \langle V, E, \pi, \mathrm{cat} \rangle$ is defined as $D(\mathcal{N}) = \max_{u,v \in V} d_{\mathcal{N}}(u, v)$; it is infinite if $\mathcal{N}$ is not strongly connected.

Definition 5 (Average k-SCC Diameter):

$$\phi_5(t, f, k) = \mathrm{avg}\{|D(N)| \quad \text{s.t. } N \in \mathrm{SCC}(\mathcal{N}_t[f], k)\}.$$

Feature $\phi_5(t, f, k)$ is the average diameter of the SCCs of $\mathcal{N}_t[f]$ having size k. It is worth noting that if $\mathcal{N}$ is strongly connected, then $\phi_5(t, f, |V|)$ coincides with the diameter of $|V|$. Moreover, $\phi_5(t, f, k)$ is finite for each k lower than or equal to the size $k_{\max}$ of the largest SCCs of $\mathcal{N}_t[f]$.

The last family of features based on the concept of diameter considers the standard deviation σ of the diameters of SCCs for the network $\mathcal{N}_t[f]$:

Definition 6 (Standard Deviation SCC Diameters):

$$\phi_6(t, f) = \sigma\{\phi_5(t, f, k) \mid k \leq k_{\max}\}.$$

The next family of features considers the density, instead of the diameter. Feature $\phi_7(t, f)$ is the density of $\mathcal{N}_t[f]$.

Definition 7 (Functional Subnetwork Density):

$$\phi_7(t, f) = \frac{|\{(u, v) \mid (u, l, v) \in E_t[f]\}|}{|V_t[f]|^2}.$$

The next family of features, $\phi_8(t, f)$, represents the probability that a random vertex v in $\mathcal{N}_t[f]$ is internally biconnected; i.e., it is involved in a triangle with two neighbors u' and u'' in $\mathcal{N}_t[f]$, which are connected to other vertices in $\mathcal{N}_t[f]$. We use IB to denote the set of vertices in $\mathcal{N}_t[f]$ that are internally biconnected.

Definition 8 (Internally Biconnected Fraction):

$$\phi_8(t, f) = \frac{|\{v \mid v \in V_t[f] \text{ and } v \in \text{IB}|}{|V_t[f]|}.$$

Similarly, feature $\phi_9(t, f)$ will represent the probability that a random vertex v in $\mathcal{N}_t[f]$ forms a pentagon involving two neighbors u and u' outside $\mathcal{N}_t[f]$ (i.e., u and u' belong to $\mathcal{N}_t[\{\mathcal{C} \setminus f\}]$, the restriction of the network to the functional categories different from f). More formally, given $\mathcal{N} = \langle V, E, \pi, \text{cat}\rangle$ and $\mathcal{N}[f] = \langle V[f], E[f], \pi[f], \text{cat}[f]\rangle$, we say that $v \in \mathcal{N}[f]$ is externally biconnected with respect to $\mathcal{N}[f]$ if the set of edges of the whole network $\mathcal{N}$ contains the edges (v, u'), (v, u''), (u', w'), (u'', w''), and (w', w''), where all the vertices are distinct and both u' and u'' belongs to $V \setminus V[f]$. We use EB to denote the set of vertices in $\mathcal{N}_t[f]$ that are externally biconnected.

Definition 9 (Externally Biconnected Fraction):

$$\phi_9(t, f) = \frac{|\{v \mid v \in V_t[f] \text{ and } v \in \text{EB}|}{|V_t[f]|}.$$

The features defined earlier rely on restricting the network to functional categories. More specific kinds of restriction are considered in the following.

We define $\phi_i^{a\neg j}(t, f)$ with $i \in \{4, \ldots, 9\}$ as the versions of $\phi_i(t, f)$, where we use the restriction of network $\mathcal{N}$ to the vertices v, such that $\pi(v,\ \text{alive}) = \text{true}$ and $\pi(v,\ \text{jail}) = \text{false}$; that is, we only focus on people who are alive and not in jail. That is, given $\mathcal{N} = \langle V, E, \pi, \text{cat}\rangle$, features $\phi_i^{a\neg j}(t, f)$ are defined using the network $\mathcal{N}[f, aj] = \langle V[f, aj], E[f, aj], \pi[f, aj], \text{cat}[f, aj]\rangle$ instead of $\mathcal{N}[f]$, where $V[f, aj] = \{v \mid v \in V,\ f \in \text{cat}_t(v)\, \pi(v, \text{alive}) = \text{true}, \pi(v, \text{jail}) = \text{false}\}$, $E[f] = (V[f, aj] \times \mathcal{L} \times V[f, aj]) \cap E$, and $\pi[f, aj]$ and $\text{cat}[f, aj]$ are the restrictions of functions π and cat to the domain $V[f, aj]$.

Likewise, we define features focusing only on people who are alive and use $\phi_i^a(t, f)$ with $i \in \{4, \ldots, 9\}$ to denote them, which are versions of $\phi_i(t, f)$, where we use the restriction of network $\mathcal{N}$ to the vertices v, such that $\pi(v,\ \text{alive}) = \text{true}$.

Let $\deg_{\mathcal{N}}^{\text{in}}(v)$ and $\deg_{\mathcal{N}}^{\text{out}}(v)$ be the in- and out-degrees of vertex v with respect to network $\mathcal{N}$, respectively. The following two features represent the average in- and out-degrees (calculated considering the edges of $\mathcal{N}$ at time t) of vertices belonging to the restriction of network $\mathcal{N}$ with respect to f at time t, respectively.

Definition 10 (Functional In-Degree):

$$\phi_{10}(t, f) = \text{avg}\big\{\deg_{\mathcal{N}_t}^{\text{in}}(v) \;\big|\; v \in V_t[f]\big\}.$$

Definition 11 (Functional Out-Degree):

$$\phi_{11}(t, f) = \text{avg}\big\{\deg_{\mathcal{N}_t}^{\text{out}}(v) \;\big|\; v \in V_t[f]\big\}.$$

In contrast to the two features defined earlier, the following features consider the restriction of the network to all the functional categories except that given in input.

Definition 12 (Complementary Functional In-Degree):

$$\phi_{12}(t, f) = \text{avg}\big\{\deg_{\mathcal{N}_t}^{\text{in}}(v) \;\big|\; v \in V_t[\mathcal{C} \setminus \{f\}]\big\}.$$

Definition 13 (Complementary Functional Out-Degree):

$$\phi_{13}(t, f) = \text{avg}\big\{\deg_{\mathcal{N}_t}^{\text{out}}(v) \;\big|\; v \in V_t[\mathcal{C} \setminus \{f\}]\big\}.$$

C. Group-Based Features

Given $\mathcal{N}_t = \langle V_t, E_t, \pi_t, \text{cat}_t\rangle$ and a set S of vertices (e.g., the set $V_t[f]$ of vertices whose set of functional categories includes f), we define the GPR of a set of nodes as follows:

$$\text{GPR}(S) = \frac{(1-\delta)\cdot|S|}{|V_t|} + \delta \cdot \left(\sum_{\substack{(u,l,v)\in E_t,\\ u\in V\setminus S,\\ v\in S}} \frac{\text{GPR}(\{u\})}{\deg_{\mathcal{N}_t}^{\text{out}}(u)}\right)$$

where $\deg_{\mathcal{N}_t}^{\text{out}}(u)$ is the out-degree of vertex u in $\mathcal{N}_t$, and δ is a damping factor as in the original definition of PageRank [22]. Note that the GPR of a singleton $\{v\}$ coincides with the PageRank of v itself, that is, $\text{GPR}(\{v\}) = \text{PR}(v)$.

The definition of GPR allows us to define a family of feature for $\mathcal{N}_t$ that depends on the choice of the set S. Specifically, we define feature $\phi_{14}(t, f)$ as the GPR of the set of vertices whose functional category is f.

Definition 14 (GPR): $\phi_{14}(t, f) = \text{GPR}(V_t[f])$.

We define another family of features using the GBC [9] of a given set S as follows. $\text{GBC}(S)$ is the sum of the fractions of all shortest paths, which traverse at least one node in S, and, thus, represents the probability that a randomly selected shortest path between two randomly selected vertices in V_t contains a node in S. For instance, if S is the set of vertices belonging to category f in the network at time t, then we obtain the following feature.

Definition 15 (GBC):

$$\phi_{15}(t, f) = \text{GBC}(V_t[f]).$$

This feature represents the probability that a randomly selected shortest path between two randomly selected vertices in V_t traverse a vertex whose functional category is f.

We can define several variants of ϕ_{14} and ϕ_{15} depending on the choice of S. Also, we can aggregate the values of GPRs and GBCs for different sets to obtain new features as follows.

Given a network $\mathcal{N}_t = \langle V_t, E_t, \pi_t\rangle$ and a functional category $f \in \mathcal{C}$, we use $P_{t,f}^r$ to denote be the set of the vertices of $\mathcal{N}_t$ involved in the functional category f and with rank greater than or equal to r, that is, $P_{t,f}^r = \{v \mid v \in V_t,\ f \in \text{cat}_t(v),\ \pi_t(v, \text{rank}) \geq r\}$. For instance, assuming that the maximum rank is 10, $P_{2010,\text{operational}}^{10}$ is the set the top-ranked operational persons in 2010.

For a subset $\mathcal{F} = \{f_1, \ldots, f_{|\mathcal{F}|}\} \subseteq \mathcal{C}$ of functional categories, let $R_t^r(\mathcal{F}) = P_{t,f_1}^r \times P_{t,f_2}^r \times \cdots \times P_{t,f_{|\mathcal{F}|}}^r$ be the Cartesian product of the sets of r-ranked people from the functional categories in $\mathcal{F}$ at time t. Therefore, after appropriately choosing the value of r, $R_t^r(\mathcal{F})$ consists of all possible $|\mathcal{F}|$-tuples of highly ranked persons, one for each functional category in $\mathcal{F}$. For each tuple $\tau \in R_t^r(\mathcal{F})$, let $S_\tau = \{p_1, \ldots, p_{|\mathcal{F}|} \mid \tau = (p_1, \ldots, p_{|\mathcal{F}|})\}$ be the set of the $|\mathcal{F}|$ people in τ.

The next features are the average of GPRs and GBCs of all combinations of people at rank r or more for categories in $\mathcal{F}$.

Definition 16 (Functional Rank GPR):

$$\phi_{16}(t, \mathcal{F}, r) = \text{avg}\big\{\text{GPR}(S_\tau) \bigm| \tau \in R_t^r(\mathcal{F})\big\}.$$

Definition 17 (Functional Rank GBC):

$$\phi_{17}(t, \mathcal{F}, r) = \text{avg}\big\{\text{GBC}(S_\tau) \bigm| \tau \in R_t^r(\mathcal{F})\big\}.$$

D. Clustering-Based Features

Given a path of length two in a network, we call a triplet the set of the three vertices in the path. A triplet is said to be open if the three vertices are connected by exactly two edges, while it is said to be closed if it consists of three edges—a closed triplet corresponds to a path of length 2 that is closed. Thus, a triangle consists of three closed triplets, corresponding to three closed path of length two, each starting on one of the vertices. The clustering coefficient is then defined as the number of closed triplets (i.e., three times the number of triangles) over the total number of triplets (both open and closed ones). It represents the probability that two vertices that are connected possibly through the third one are also directly connected.

The clustering coefficient can be defined for both directed graphs, such as our network $\mathcal{N} = \langle V, E, \pi, \text{cat}\rangle$, and undirected graphs such as the undirected (and unlabeled) version of $\mathcal{N}$ defined through $\overline{E}$, the set of undirected (and unlabeled) edges obtained from E, as $\overline{\mathcal{N}} = \langle V, \overline{E}, \pi, \text{cat}\rangle$. Specifically, given a (directed or undirected) graph G whose set of edges is E, the cluster coefficient for G is as follows:

$$\text{CC}(G) = \frac{3 \times |\{(u, v, w) \mid (u, v), (v, w), \text{ and } (w, u) \in E\}|}{|\{(u, v, w) \mid (u, v) \text{ and } (v, w) \in E\}|}.$$

Thus, considering $\mathcal{N}_t = \langle V_t, E_t, \pi_t\rangle$ and $\overline{\mathcal{N}_t} = \langle V_t, \overline{E}_t, \pi_t\rangle$, we define the following two features.

Definition 18 (Global Directed Clustering Coefficient):

$$\phi_{18}(t, f) = \text{CC}(\mathcal{N}_t[f]).$$

Definition 19 (Global Undirected Clustering Coefficient):

$$\phi_{19}(t, f) = \text{CC}(\overline{\mathcal{N}}_t[f]]).$$

We now define two classes of features that rely on groups of top-ranked people from functional categories in a set $\mathcal{F}$.

Let $R_t^r(\mathcal{F}) = P_{t,f_1}^r \times P_{t,f_2}^r \times \cdots \times P_{t,f_{|\mathcal{F}|}}^r$ be the Cartesian product of the sets of r-ranked people from the functional categories in $\mathcal{F} = \{f_1, \ldots, f_{|\mathcal{F}|}\} \subseteq \mathcal{C}$ at time t, and $S_\tau = \{p_1, \ldots, p_{|\mathcal{F}|} \mid \tau = (p_1, \ldots, p_{|\mathcal{F}|})\}$ be the set of the $|\mathcal{F}|$ people in $\tau \in R_t^r(\mathcal{F})$, as defined for the GbFs. Given a network $\mathcal{N} = \langle V, E, \pi, \text{cat}\rangle$ and a set S of vertices, we use $\mathcal{N}[S]$ to denote the subnetwork consisting of only the vertices and edges involving S. Thus, for $\tau \in R_t^r(\mathcal{F})$, $\mathcal{N}_t[S_\tau]$ is the subnetwork (at time t) consisting of only the edges between a group of r-ranked people, each having a functional category in $\mathcal{F}$. The following feature is the average of the clustering coefficients of all groups of r-ranked people whose functional category is in $\mathcal{F}$ and belonging to the network at time t.

Definition 20 (Average Group Functional Ranked CC):

$$\phi_{20}(t, \mathcal{F}, r) = \text{avg}\big\{\text{CC}(\mathcal{N}_t[S_\tau])\bigm| \tau \in R_t^r(\mathcal{F})\big\}.$$

Next, we consider immediate neighbors of r-ranked people in $R_t^r(\mathcal{F})$ and define a feature representing the average of the clustering coefficients of the neighbors of r-ranked people. More formally, given a network $\mathcal{N} = \langle V, E, \pi\rangle$ and a set S of vertices, we use $nb_{\mathcal{N}}(S)$ to denote the immediate neighbors of S in $\mathcal{N}$, i.e., $nb_{\mathcal{N}}(S) = \{v \mid (v, l, u) \text{ or } (u, l, v) \in E, u \in S\}$. Given this, we obtain the following feature.

Definition 21 (Average Neighbor Functional Ranked CC):

$$\phi_{21}(t, \mathcal{F}, r) = \text{avg}\big\{\text{CC}(\mathcal{N}_t[S_\tau \cup nb_{\mathcal{N}_t}(S_\tau)])\bigm| \tau \in R_t^r(\mathcal{F})\big\}.$$

E. Time-Lagged Features

We define time-lagged variants for each feature. For each $t \in T$, functional category $f \in \mathcal{C}$, and feature ϕ_i, we define time-lagged features with $\tau \in \{1, 2, 3\}$.

Definition 22 (Lagged Feature Value):

$$\Psi_i(t, f, \tau) = \phi_i(t - \tau, f).$$

That is, $\Psi_i(t, f, \tau)$ is the value taken by feature ϕ_i at the previous time point $t - \tau$.

Definition 23 (Lagged Average Value):

$$\Omega_i(t, f, \tau) = \text{avg}\{\phi_i(t', f) \mid t' \in [t - \tau, t]\}.$$

Thus, $\Omega_i(t, f, \tau)$ is the average of the values of the last $\tau + 1$ single-time point features.

References

[1] F. Spezzano, V. Subrahmanian, and A. Mannes, "Stone: Shaping terrorist organizational network efficiency," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2013, pp. 348–355.

[2] F. Spezzano, V. S. Subrahmanian, and A. Mannes, "Reshaping terrorist networks," *Commun. ACM*, vol. 57, no. 8, pp. 60–69, Aug. 2014.

[3] V. Latora and M. Marchiori, "How the science of complex networks can help develop strategies against terrorism," *Chaos, Solitons Fractals*, vol. 20, no. 1, pp. 69–75, Apr. 2004.

[4] M. C. Horowitz and P. B. Potter, "Allying to kill: Terrorist intergroup cooperation and the consequences for lethality," *J. Conflict Resolution*, vol. 58, no. 2, pp. 199–225, 2014.

[5] V. Krebs, "Mapping networks of terrorist cells," *Connections*, vol. 24, no. 3, pp. 43–52, 2002.

[6] R. Lindelauf, P. Borm, and H. Hamers, "The influence of secrecy on the communication structure of covert networks," *Social Netw.*, vol. 31, no. 2, pp. 126–137, 2009.

[7] R. Lindelauf, H. Hamers, and B. Husslage, "Cooperative game theoretic centrality analysis of terrorist networks: The cases of Jemaah Islamiyah and Al Qaeda," *Eur. J. Oper. Res.*, vol. 229, no. 1, pp. 230–238, 2013.

[8] K. M. Carley, J. Reminga, and N. Kamneva, "Destabilizing terrorist networks," in *Proc. NAACSOS Conf.*, Pittsburgh, PA, USA, 2003, pp. 1–6.

[9] S. Dolev, Y. Elovici, and R. Puzis, "Routing betweenness centrality," *J. ACM*, vol. 57, no. 4, p. 25, 2010.

[10] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[11] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.

[12] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[13] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer-Verlag, 1995.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14.

[15] H. Hegre *et al.*, "ViEWS: A political violence early-warning system," *J. Peace Res.*, vol. 56, no. 2, pp. 155–174, Mar. 2019.

[16] Y. Yang, A. R. Pah, and B. Uzzi, "Quantifying the future lethality of terror organizations," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 43, pp. 21463–21468, Oct. 2019.

[17] V. Subrahmanian, A. Mannes, A. Sliva, J. Shakarian, and J. P. Dickerson, *Computational Analysis of Terrorist Groups: Lashkar-e-Taiba*. Berlin, Germany: Springer, 2012.

[18] C. Leuprecht, O. Walther, D. B. Skillicorn, and H. Ryde-Collins, "Hezbollah's global tentacles: A relational approach to convergence with transnational organized crime," *Terrorism Political Violence*, vol. 29, no. 5, pp. 902–921, Sep. 2017.

[19] N. W. Metternich, C. Dorff, M. Gallop, S. Weschle, and M. D. Ward, "Antigovernment networks in civil conflicts: How network structures affect conflictual behavior," *Amer. J. Political Sci.*, vol. 57, no. 4, pp. 892–911, 2013.

[20] C. Dorff, M. Gallop, and S. Minhas, "Networks of violence: Predicting conflict in Nigeria," *J. Politics*, vol. 82, no. 2, pp. 476–493, Apr. 2020.

[21] N. Rosenblatt, C. Winter, and R. Basra, "Islamic state propaganda and attacks," *Perspect. Terrorism*, vol. 13, no. 5, pp. 39–60, 2019.

[22] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Comput. Netw.*, vol. 30, nos. 1–7, pp. 107–117, Apr. 1998.

Youdinghuan Chen has undertook Statistical and Machine Learning Scientist in Industry. He is currently a Teaching Faculty of data and applied sciences with the Wilmington College of Arts and Sciences, Wilmington University, New Castle, DE, USA.

Chongyang Gao is currently pursuing the Ph.D. degree in computer science with Northwestern University, Evanston, IL, USA.

Daveed Gartenstein-Ross is currently the Founder and the Chief Executive Officer of Valens Global, Washington, DC, USA, and also an Instructor with Carnegie Mellon University, Pittsburgh, PA, USA, and Duke University, Durham, NC, USA. He has authored or is a volume editor of over 30 books and monographs.

Kevin T. Greene is currently an Associate Research Scholar with the Empirical Studies of Conflict Project, Princeton University, Princeton, NJ, USA. His research interests include the role of information communication technologies in both international and domestic politics.

Karin Kalif received the B.Sc. degree in mathematics and the M.Sc. degree in computer science from Bar-Ilan University, Ramat Gan, Israel, in 2012 and 2020, respectively.

She is currently a Computer Vision Algorithm Developer with Percepto, Modi'in-Maccabim-Re'ut, Israel.

Sarit Kraus is currently a Professor of computer science with Bar-Ilan University, Ramat Gan, Israel. Her research interests include intelligent agents and multiagent systems integrating machine-learning techniques with optimization and game theory methods.

Francesco Parisi is currently an Associate Professor of computer engineering with the Department of Computer Engineering, Modeling, Electronics, and System (DIMES), University of Calabria, Rende, Italy. His research interests include inconsistency management, knowledge representation, and reasoning under uncertainty.

Chiara Pulice is currently a Senior Research Scientist in computer science with Northwestern University, Evanston, IL, USA. Her research interests include counter terrorism, machine learning, and social network analysis.

Anja Subasic is currently a Machine Learning Engineer with Bloomberg LP, New York, NY, USA, where she is involved in natural language processing.

V. S. Subrahmanian was a Professor with the University of Maryland, College Park, MD, USA, where he was the Director of the Institute for Advanced Computer Studies. He was a Dartmouth College Distinguished Professor in cybersecurity, technology, and society and the Director of the Institute for Security, Technology and Society, Northwestern University, Evanston, IL, USA, where he is currently a Walter P. Murphy Professor of computer science and a Buffett Faculty Fellow.