Data Driven Inference in Populations of Agents

by

Elham Shaabani

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved February 2019 by the
Graduate Supervisory Committee:

Paulo Shakarian, Chair
Hasan Davulcu
Ross Maciejewski
Scott Decker

ARIZONA STATE UNIVERSITY

May 2019

ABSTRACT

In the artificial intelligence literature, three forms of reasoning are commonly employed to understand agent behavior: inductive, deductive, and abductive. More recently, data-driven approaches leveraging ideas such as machine learning, data mining, and social network analysis have gained popularity. While data-driven variants of the aforementioned forms of reasoning have been applied separately, there is little work on how data-driven approaches across all three forms relate and lend themselves to practical applications. Given an agent behavior and the percept sequence, how one can identify a specific outcome such as the likeliest explanation? To address real-world problems, it is vital to understand the different types of reasonings which can lead to better data-driven inference.

This dissertation has laid the groundwork for studying these relationships and applying them to three real-world problems. In criminal modeling, inductive and deductive reasonings are applied to early prediction of violent criminal gang members. To address this problem the features derived from the co-arrestee social network as well as geographical and temporal features are leveraged. Then, a data-driven variant of geospatial abductive inference is studied in missing person problem to locate the missing person. Finally, induction and abduction reasonings are studied for identifying pathogenic accounts of a cascade in social networks.

DEDICATION

To my dear family

ACKNOWLEDGMENTS

I would like to express my utmost appreciation and gratitude to my advisor, Prof. Paulo Shakarian who has mentored, supported, encouraged and motivated me throughout my Ph.D. I would also like to thank my committee members, Prof. Hasan Davulcu, Prof. Ross Maciejewski, and Prof. Scott Decker. Thank you all for sharing your knowledge and passion with me.

I am thankful to Jana Shakarian for her support during the last 5 years. I also would like to thank my lab-mates at Cyber-Socio Intelligent Systems (CySIS) Lab who have all contributed to my success: Ashkan Aleali, Ruocheng Guo, Hamidreza Alvari, Eric Nunes, Ericsson Santana Marin, Soumajyoti Sarkar, Mohammed Almukaynizi, Priyama Biswas, and Abhinav Bhatnager. I am also thankful to my friends at Arizona State University specially Ghazal Shams and Nooshin Shomal Zadeh for their support.

During my Ph.D., I had three valuable internship experiences. I spent summer 2017 under the supervision of Wai-Kin Lau at GoDaddy Inc. I would like to thank him, and Jacob Huang. I am also grateful to Amit Bansal and Farhad Farahani for being great mentors during my 2018 summer internship at PayPal Inc. I also would like to thank my collaborators Swati Jain, and Chao Cheng. I spent Fall 2018 at Walmart Labs where I am proud to join as a full-time employee. I was truly fortunate to work with Marcus Csaky, Matyas Sustik, Jagdish Ramakrishnan, and Chao Li. I am really grateful to my team at Walmart Labs.

Words cannot express how grateful I am to my husband, Ashkan, my mom, Fatemeh, my dad, Ahmad, and my sister, Shabnam for their unconditional love, support and encouragement throughout my life. To them, I dedicate this thesis.

This work was supported by Find Me Group, DoD Minerva program and AFOSR (grant FA9550-15-1-0159), and ARO (grant W911NF-15-1-0282).

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

In the artificial intelligence literature, three forms of reasoning are commonly employed to understand agent behavior: deductive, inductive, and abudctive. Deductive inference infers a result for a specific case given a premise while induction inference is to infer a premise given the case and result. Abductive inference is to infer the likeliest possible explanation given a set of facts. In abduction, we need to make the hypotheses as well as finding the most plausible one.

More recently, data-driven approaches leveraging ideas such as machine learning, data mining, and social network analysis have gained popularity. While data-driven variants of the aforementioned forms of reasoning have been applied separately, there is less work on how these approaches relate and lend themselves to practical applications. In this dissertation, we aim to answer the following fundamental questions: *How can we identify a specific outcome given the agent behavior and the percept sequence? How can real-world problems be solved using the combination of reasoning methods? When we have to use abductive, inductive or deductive reasonings? How can we infer the likeliest explanation given a set of facts?* Answers to such questions are vital to understand the different types of reasonings and can lead to better data driven inference.

In our research, we propose combination of inference methods to identify outcomes of the agents. These methods are designed to support the objectives of studying the relationships among reasoning approaches and applying them to real-world problems. This thesis is organized as follows:

Chapter 2. We wish to reason if the agent takes a specific type of action as the next action, given a model of past agent behaviors. In this regard, we consider the following

problem, early prediction of violent criminal gang members using deductive reasoning. Our approach relies on modified centrality measures that take into account additional data of the individuals in the social network of co-arrestees which together with other arrest metadata provide a rich set of features for a classification algorithm. We evaluate our method using real-world offender data from Chicago Police Department. We show our approach obtains high precision and recall (0.89 and 0.78 respectively) in the case where the entire network is known and out-performs current approaches used by law-enforcement to the problem in the case where the network is discovered overtime by virtue of new arrests - mimicking real-world law-enforcement operations [64].

Chapter 3. In this chapter, we studied generating the most plausible explanations given a set of observations. We introduce the Missing Person Intelligence Synthesis Toolkit (MIST) which leverages a data-driven variant of geospatial abductive inference for finding missing persons. The system takes search locations provided by a group of experts and rank-orders them based on the probability assigned to areas based on the prior performance of the experts taken as a group. Evaluation of our approach compared to the current practices employed by the Find Me Group (a non-profit organization led by former law enforcement professionals dedicated to missing persons cases) demonstrates that we significantly reduce the search area– leading to a reduction of 53 square miles over 29 cases we examined in our experiments [65].

Chapter 4. In this chapter, we aim to find the set of best explanations given extremist cascades. "Pathogenic Social Media" accounts have the capability of spreading harmful mis-information to viral proportions. We introduced an unsupervised causality-based framework that leverages label propagation. This approach identifies these users without using network structure, cascade path information, content and user's information. We evaluate our approach using Twitter dataset. We show our approach obtains higher

precision (0.75) in identifying pathogenic social media accounts in comparison with random (precision of 0.11) and existing bot detection (precision of 0.16) methods [66].

Chapter 5. And finally, in this chapter, we adopt the causal inference framework along with graph-based metrics in order to distinguish pathogenic social media accounts from normal users within a short time of their activities. We propose both supervised and semi-supervised approaches without taking the network information and content into account. Results on a real-world dataset from Twitter accentuates the advantage of our proposed frameworks. We show our approach achieves 0.28 improvement in F1 score over existing approaches with the precision of 0.90 and F1 score 0.63 [67].

The description of the aforementioned problems is summarized in Table 1.1.

Table 1.1: Description of Studied Agents

	Agent	Percept sequence	Agent behavior	Outcome	Reasoning
Violence Prediction	Criminal gang member	Social network	Past crimes	Violent behavior	Induction, Deduction
Missing person	Missing person case	Reporters	Reported locations	Most potential location	Induction, Abduction
Pathogenic account detection	Extremist cascade	Network (missing), Past cascades	Users of the cascade	Pathogenic accounts	Induction, Abduction
Pathogenic account detection	Extremist cascade	Network (missing), Past cascades	Users of the cascade	Minimum set of Pathogenic accounts	Induction, Abduction

Chapter 2

EARLY IDENTIFICATION OF VIOLENT CRIMINAL GANG MEMBERS

2.1 Introduction

Gang violence is a major problem in the United States [8, 9] - accounting for 20 to 50 percent of homicides in many major cities [33]. Yet, law enforcement actually has existing data on many of these groups. For example the underlying social network structure is often recorded by law-enforcement and has previously been shown useful in enabling "smart policing" tactics [48] and improving law-enforcement's understanding of a gang's organizational structure [51]. In this chapter, we look to leverage this gang social network information to create features that allows us to classify individuals as potentially violent. While the results of such a classifier are insufficient to lead to arrests, it is able to provide the police leads to individuals who are likely to be involved in violence, allowing for a more focused policing with respect to patrols and intelligence gathering. *Our key aim is to significantly reduce the population of potential violent gang members which will lead to more efficient policing.*

In this research, we introduce our method for identifying potentially violent gang members that leverages features derived from the co-arestee social network of criminal gangs in a classifier to identify potentially violent individuals. We note that this classification problem is particularly difficult due to not only data imbalances, but also due to the fact that many violent crimes are conducted due to heightened emotions - and hence difficult to identify. Though we augment our network-based features with some additional meta-data from the arrest records, our approach does **not** leverage features concerning the race, ethnicity, or gender of individuals in the social network. We evaluate our method using

real-world offender data from the Chicago Police Department. This chapter makes the following contributions:

- We discuss how centrality measurements such as degree, closeness, and betweenness when modified to account for metadata about past offenses such as the type of offense and whether the offense was classified as "violent" can serve as robust features for identifying violent offenders.
- We show how the network features, combined with other feature categories provide surprisingly robust performance when the entire offender is known in terms of both precision (0.89) and recall (0.78) using cross-validation.
- We then test our methods in the case where the network is exposed over time (by virtue of new arrests) which mimics an operational situation. Though precision and recall are reduced in this case, we show that our method significantly outperforms the baseline approach currently in use by law-enforcement - on average increasing precision and recall by more than two and three times respectively.

In addition to these main results, we also present some side results on the structure and nature of the police dataset we examine. The chapter is organized as follows. In Section 2.2 we motivate this difficult problem within the law-enforcement community. This is followed by a description of our dataset along with technical notation in Section 2.3. There, we also describe some interesting aspects of the gang arrest dataset and our co-arrestee network. In Section 2.4 we formally define our problem, describe existing approaches, and then describe the features we use in our approach. Then we present our results in Section 2.5 for both cases where we assume the underlying network is known and when we discover the network over time (mimicking an operational scenario). Finally, related work is discussed in Section 2.6.

2.2 Background

A recent study shows that the network for gunshot victimization is denser than previously believed [50]. According to the authors, within the city of Chicago over 70% of all gunshot victims are contained within only 6% of the total population These findings validate what has been considered common knowledge among police for decades: who you hang out with matters, and if you hang out with those who engage in or are victims of violence you are more likely to become an offender or victim yourself.

Identifying potential offenders of gun violence has also been a staple practice for most law enforcement agencies as an attempt to curtail future victimization. When gang conflicts get "hot," it's common for law enforcement agents to put together a list of known "shooters": those known gang members with an existing criminal history for gun violence and a predilection for engaging in such illegal activity. Law enforcement agents then attempt to make contact with these individuals with the expectation that such direct contact might prevent violence. For most law enforcement agencies, however, this practice is performed in a very ad-hoc manner. Identifying these individuals for intervention has relied primarily on the ability of law enforcement agents to remember and identify at-risk individuals. While feasible for small or discreet networks, the ability to recall multiple individuals in large networks that cross large geographic regions and interact with multiple networks becomes increasing difficult. This difficulty increases significantly as relationships between networks change, known individuals leave the network, and new individuals enter it. In particular, the practice is less than idea because it requires officers to attempt to recall criminal history and network association data that varies between network members. For example, a subject who has been arrested on multiple occasions for carrying a gun or has been arrested for shooting another individual is easy to recall, but recalling and quantifying the risk for a subject with multiple arrests for non-gun violence and a direct association with several offenders and

victims of gun violence can be much more difficult. In short, identifying a known "shooter" is relatively straightforward: they are known. The approach in this chapter synthesizes network connectivity other attributes of the subject to identify those individuals at risk that law enforcement might not yet know.

Using this information, law enforcement agents may not only more reliably and consistently identify those individuals most likely to engage in acts of violence or become victims of violence due to their personal associations with it, but also to more effectively manage agency resources. Intervention strategies may include service providers outside law enforcement, such as family members, social service providers, current or former educators, and clergy. This diversity in approach not only delivers a more powerful "stop the violence" message but provides a kind of force multiplier for law enforcement, increasing the number of persons involved in the effort to prevent violence. Identifying specific individuals for intervention also allows for a more targeted effort by law enforcement in terms of personnel and geographic areas needing coverage. Blanketing violence reduction strategies that saturate geographic areas with law enforcement agents and rely on direct contact with large numbers of criminal network members are inefficient and resource consuming. Focusing efforts on those individuals most likely to engage in violence allows law enforcement to focus on smaller groups of people and smaller geographic areas (those areas within which those individuals identified are known to frequent). Therefore, our approach can significantly improve such efforts to identify violent individuals. In this chapter, we see how our method not only out-performs the current social network heuristic used by police, but also that it provides a much smaller and more precise list of potentially violent offenders than simply listing those with a violent criminal record.

2.3 Gang Co-offender Network

In this section, we introduce the necessary basic notation to describe our co-offender network and then provide details of our real-world criminal dataset and study some of its properties.

2.3.1 Technical Preliminaries

Throughout this chapter we shall represent an offender network as an undirected graph $G = (V, E)$ where the nodes correspond with previous offenders and an undirected edge exists between offenders if they were arrested together. We will use τ to denote the set of timepoints (dates). We also have three sets of labels for the nodes: $\mathcal{V}$, $\mathcal{S}$, *gang* which are the sets of violent crimes, non violent crimes, and gangs. For each time point t and each node v, the binary variable $arr_v^t \in \{\mathsf{true}, \mathsf{false}\}$ denotes if v was arrested at time t and $distr_v^t, beat_v^t, gang_v^t$ to denote the district, beat, and gang affiliation of v at time t (we will assume that time is fine-grain enough to ensure that at each time unit an individual is arrested no more than once). If we drop the t superscript for these three symbols, it will denote the most recent district, beat, and gang associated with v in the knowledgebase. We shall use the sets $\mathcal{V}_v^t$ and $\mathcal{S}_v^t$ to denote the set of violent and non violent offenses committed by v at time t respectively. Note if $arr_v^t = \mathsf{false}$ then $\mathcal{V}_v^t = \emptyset$. We will drop the superscript t for this symbol to denote the union of labels at any time t in the historical knowledgebase. We also note that the edges in the graph also depend on time, but for sake of readability, we shall state with words the duration of time considered for the edges.

For a given violent crime $c \in \mathcal{V} \cup \mathcal{S}$, we will use the notation $V_c^t = \{v \in V \textit{ s.t. } c \in \mathcal{V}_v^t\}$ (intuitively, the subset of the population who have committed crime c at time t). Again, we will drop the superscript t if v could have committed crime c at any time in the historical knowledgebase. For a set of labels $C \subseteq \mathcal{V} \cup \mathcal{S}$, we will extend this notation: $V_C^t = \{v \in$

V *s.t.* $C \cap \mathcal{V}_v^t \neq \emptyset\}$. We will slightly abuse notation here: $V_\emptyset^t = V$. We will use similar notation for denoting a subset of the population that are members of a certain gang. For instance, V_{gang_v} refers to the set of nodes who are in the same gang as node v. Likewise, we shall use the same notation for subgraphs: G_C^t is the subgraph of G containing only nodes in V_C^t and their adjacent edges. We will use the function $d : V \times V \to \mathbb{N}$ to denote the distance between two nodes - which for this chapter will be the number of links in the shortest path. For a given node v, the set $N_v^i = \{v' \in V \text{ } s.t. \text{ } d(v, v') = i\}$ – the set of nodes that are whose shortest path is exactly i hops from v. For two nodes v, v', we will use the notation $\sigma(v, v')$ to be the number of shortest paths between v and v'. For nodes u, v, v', $\sigma_u(v, v')$ will be the number of shortest paths between v and v' that pass through u.

For a given subgraph G' of G, we shall use $\mathsf{C}(G')$ to denote the largest connected component of G' and for node $v \in G'$, we will use the notation $\mathsf{C}_v(G')$ to denote the connected component of G' to which v belongs. If we apply a community finding algorithm to subgraph G', we will use the notation $\mathsf{P}_v(G')$ to denote the partition of G' to which v belongs. We will use the notation $|\cdot|$ to denote the size of a set or the number of nodes in a subgraph.

2.3.2 Overview of Network Data

In this section, we describe our police dataset and the associated co-offender network as well as some interesting characteristics that we have noticed.

Police Dataset. Our dataset consists of gang-related arrest incidents gathered from August 2011 - August 2014 in Chicago as well as their immediate associates. This data set includes locations, dates, the links between the joint arrests, and the gang affiliation of the offenders. In Table 2.1, we summarize some of the important characteristics of the dataset.

Violent Crimes. In our dataset, the set $\mathcal{V}$ consists of the following crimes have been identified by the Chicago Police as violent crimes: homicide (first or second degree murder),

Table 2.1: Summary of Arrest Data

Name	**Value**
Number of records	64466
Violent offense	4450
Homicide	312
Criminal sexual assault	153
Robbery	1959
Aggravated assault	1441
Aggravated battery	896
Non violent offense	60016

Table 2.2: Network Properties

Name	**Values**
Vertices	9373
Edges	17197
Average degree	3.66
Average clustering	0.5
Transitivity	0.62
Connected components	1843
Largest connected component diameter	36
Largest connected component average path length	12.22
Largest connected component average clustering	0.63

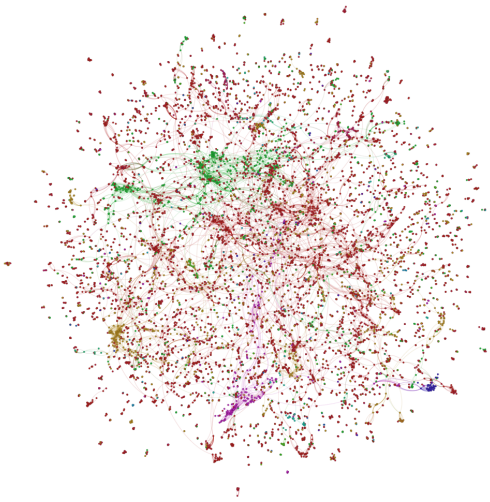


Fig. 2.1: The Gang Co-offender Network. Each Color Corresponds With a Different Gang.

criminal sexual assault, robbery, aggravated assault, and aggravated battery. All aforementioned offenses are also FBI "index" crimes as well. A key aspect about the violent crimes is that the dataset is highly imbalanced with much more arrests for non violent crimes vs. arrests for violent crimes (60016 vs. 4450).

Network Properties. From the arrest data, we were able to construct the *co-offender network*. In this network, the isolated vertices are eliminated due to the lack of structural information. A visualization of the network is depicted in Fig. 2.1 and we have included summary statistics in Table 2.2. In this network, we studied its degree distribution (Fig. 2.2). Unlike the degree distribution for other scale free social networks, the degree distribution for the offender network is *exponential* rather than *power law*. However, despite the degree distribution being similar to that of a random (E-R) or small world network topology [82], we noticed other characteristics that indicate differently. The co-offender network has a much higher average clustering coefficient than in a random network and does not follow the properties of the small world topology due to the relative high diameter and average shortest path (computed for the largest connected component.)

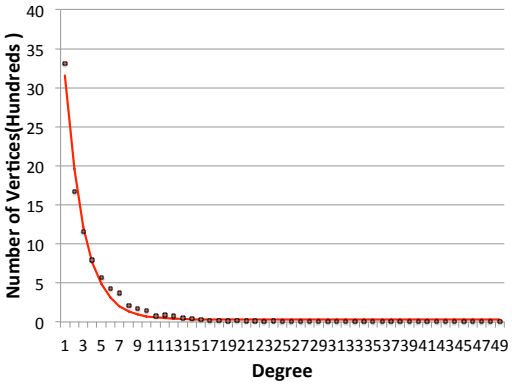


Fig. 2.2: Network Degree Distribution. The Exponential Function Fits to the Distribution ($R^2 = 0.77$).

Repeat Offenders. There are many instances of repeated offenses from the same offender. Fig. 2.3 shows the distribution of the repeated arrests for each individual in the dataset. This indicates that arrest records have utility in identifying future offenders.

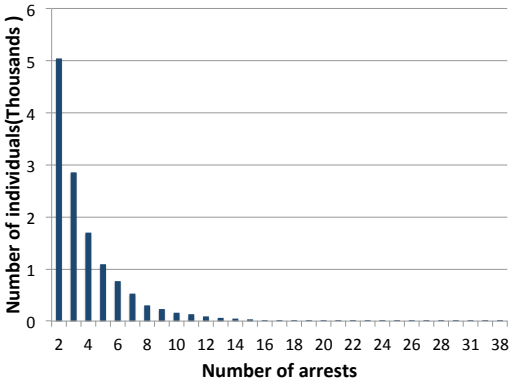


Fig. 2.3: Repeated Arrests. 12866 Instances of One-time Arrests Have Been Removed.

Seasonality of Crime. There is also a higher chance of criminal activities in different months of the year. Fig. 2.4 demonstrates some of these variations. As per police observations, both violent and non-violent crime incidents are lower in the winter months (Dec.-Feb.).

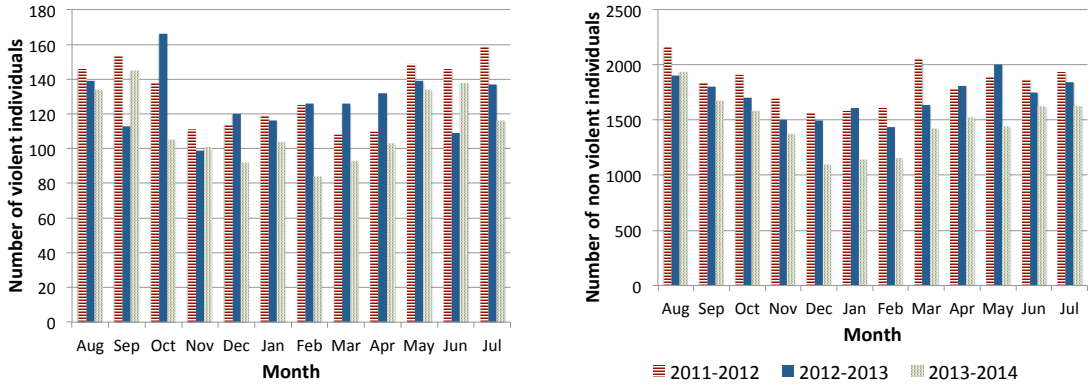


Fig. 2.4: Seasonality of Crime.

2.4 Identifying Violent Offenders

In this section, we describe our problem, some of the existing practical approaches used by law-enforcement, and our approach based on supervised learning with features primarily generated by the network topology.

2.4.1 Problem Statement

Given a co-offender network, $G = (V, E)$ and for each historical timepoint $t \in \tau = \{1, \ldots, t_{\max}\}$ and $v \in V$, we have the values of $arr_v^t, distr_v^t, beat_v^t$ and elements of the sets $\mathcal{V}_v^t, gang_v^t$, we wish to identify set $\{v \in V \textit{ s.t. } \exists t > t_{\max} \textit{ where } |\mathcal{V}_v^t| > 0\}$. In other words, we wish to find a set of offenders in our current co-offender network that commit a violent crime in the future.

2.4.2 Existing Methods

Here we describe two common techniques often used by law-enforcement to predict violent offenders. The first is a simple heuristic based on violent activities in the past. The second is a heuristic that was based on the findings of [48] which was designed to locate

future *victims* of violent crime. Both of these approaches are ad-hoc practical approaches that have become "best practices" for predicting violent offenders. However, we are not aware of any data-driven, formal evaluation of these methods in the literature.

Past Violent Activities (PVA). The first ad-hoc approach is quite simple: if an offender has committed a violent crime in the past, we claim that he will commit a violent crime in the future. An obvious variant of this approach is to return the set of violent offenders from the last Δt days. We note in practice, if police also have records of those who are incarcerated, and such individuals would be removed from the list (due to the different jurisdictions of police and corrections in the Chicago area, we did not have access to incarceration data - however discussed re-arrests observed in the data in the previous section).

Two-Hop Heuristic (THH). The two-hop heuristic is based on the result of [48] which investigated a social network of gunshot *victims* in Boston and found an inverse relationship between the probability of being a gunshot victim and the shortest path distance on the network to the nearest previous gunshot victim. Hence, THH returns all neighbors one and two hops away from previous violent criminals (see Algorithm 1 for details on the version we used in our experiments - which was the best-performing variant for our data). The Chicago Police have adopted a variant of this method to identify potential gang victims using a combination of arrest and victim data - the co-arrestee network of criminal gang members includes many individuals who are also victims of violent crime (this is a direct result of gang conflict). We note that victim information did not offer a significant improvement to our approach, except the trivial case that a homicide victim cannot commit any crime in the future.

2.4.3 Supervised Learning Approach

We evaluated many different supervised learning approaches including naive bayes (NB), logistic regression (LR), decision tree (DT), random forest (RF), neural network (NN), and

Algorithm 1 Two-Hop Heuristic

procedure TWOHOP(G) ▷ Offender network G.
  $R \leftarrow \{\}$ ▷ Identified violent offenders.
  $VICTIMS \leftarrow \{u \in G | is\_homicide\_victim(u)\}$
  for $v \in VICTIMS$ **do**
    $N \leftarrow N_v^1 \cup N_v^2$ ▷ Immediate neighbors
    $R \leftarrow R \cup \{u \in N \; s.t. \; \mathcal{V}_u = \emptyset\}$
  return R

support vector machines (SVM) on the same set of features for the nodes in the network that we shall describe in this section. We also explored combining these approaches with techniques for imbalanced data such as SMOTE [16] and Borderline SMOTE [32], however we do not report the results of Borderline SMOTE as it provided no significant difference from SMOTE. We group our features into four categories: 1) neighborhood-based (having to do with the immediate neighbors of a given node), 2) network-based (features that require the consideration of more than a nodes immediate and nearby neighbors), 3) temporal characteristics, and 4) geographic characteristics.

Neighborhood-Based Features

Neighborhood-based features are the features computed using each node and its first and/or second level neighbors in G – often with respect to some $C \subseteq \mathcal{V}$. The simplest such measure is the degree of vertex v – corresponding to the number of offenders arrested with v. We can easily extend this for some set of crimes of interest (C) where we look at all the neighbors of v who have committed a crime in C. This generalizes degree (as that is the case where $C = \emptyset$). In our experiments, we found the most useful neighborhood features to be in the case where $C = \mathcal{V}$ though standard degree ($C = \emptyset$) was also used. We also found that using

combinations of the following booleans based on the below definition also proved to be useful:

$$maj_v(C, i) = |\{u | u \in (\cup_i N_v^i) \cap V_C\}| \geq 0.5 \times |(\cup_i N_v^i)| \tag{2.1}$$

Intuitively, $maj_v(C, i)$ is true if at least half of the nodes within a network distance of i from node v have committed a crime in C and false otherwise. Using these intuitions, we explored the space of variants of these neighborhood-based features and list those we found to be best-performing in Table 2.3.

Table 2.3: Neighborhood-based Features

Name	**Definition**
Degree (w.r.t. C)	$\|\{u \| u \in N_v^1 \cap V_C\}\|$
Fraction of 1-hop neighbors committing a crime in C	$\|\{u \| u \in N_v^1 \cap V_C\}\|/\|N_v^1\|$
Fraction of 2-hop neighbors committing a crime in C	$\|\{u \| u \in N_v^2 \cap V_C\}\|/\|N_v^2\|$
Majority of 1-hop and 2-hop neighbors committing a crime in C	$maj_v(C, 1) \wedge maj_v(C, 2)$
Minority of 1-hop and majority of 2-hop neighbors comitting a crime in C	$\neg maj_v(C, 1) \wedge maj_v(C, 2)$

Network-Based Features

Network-based features fall into two sub-categories that we shall describe in this section: community-based and path-based.

Network-based community features. We use several notions of a node's community when engineering features: the connected component to which a node belongs, the gang to which a

Table 2.4: Network-based Features (Community)

Name	**Definition**
Component size when v is removed	$\lvert\mathsf{C}(\mathsf{C}_v(G) \setminus \{v\})\rvert$
Largest component size with a violent node after v is removed	$\max_{v' \in \mathsf{C}(\mathsf{C}_v(G)\{v\} \cap V_{\mathcal{V}}} \lvert X_{v'}\rvert \; where \; X_{v'} = \mathsf{C}_{v'}(\mathsf{C}_v(G)\{v\})$
Group size	$\lvert\mathsf{P}_v(G_{gang_v})\rvert$
Relationships within the group	$\lvert\{(u,v) \in E \; s.t. \; u,v \in \mathsf{P}_v(G_{gang_v})\}\rvert$
Number of violent members in the group	$\lvert\{v' \in \mathsf{P}_v(G_{gang_v}) \; s.t. \; \mathcal{V}_v \neq \emptyset\}\rvert$
Triangles in group	Number of triangles within subgraph $\mathsf{P}_v(G_{gang_v})$
Transitivity of group	$\frac{No. \; of \; triangles \; in \; \mathsf{P}_v(G_{gang_v})}{No. \; of \; \text{“}\vee\text{”}s \; in \; \mathsf{P}_v(G_{gang_v})}$
Group-to-group connections	$\lvert\{u \in \mathsf{P}_v(G_{gang_v}) \; s.t. \; \exists (u,w) \in E \; where \; w \notin \mathsf{P}_v(G_{gang_v})\}\rvert$
Gang-to-gang connections	$\lvert\{u \in G_{gang_v} \; s.t. \; \exists (u,w) \in E \; where \; w \notin G_{gang_v}\}\rvert$

node belongs, and what we will refer to as an individual's group. The connected component is simply based on the overall network structure, while the gang is simply the subgraph induced by the individuals in the network who belong to the same gang (the social network of node v's gang is denoted G_{gang_v}. A nodes group is defined as the partition he/she belongs to based on a partition of G_{gang_v} found using the Louvain algorithm [25]. We found in our previous work [51] and ensuing experience with the Chicago Police that the groups

Table 2.5: Network-based Features (Path)

Name	**Definition**
Betweenness (w.r.t. C)	$\sum_{u,w \in V_C} \frac{\sigma_v(u,w)}{\sigma(u,w)}$
Closeness (w.r.t. C)	$(\lvert V_C \rvert - 1) / \sum_{u \in V_C} d(u,v)$
Shell Number (w.r.t. C)	$\mathit{shell}_C(v)$ (see appendix for details)
Propagation (w.r.t. C)	1 if $v \in \Gamma_\kappa(V_\mathcal{V})$, 0 otherwise. (see appendix for details)

produced in this method were highly relevant operationally. In this work, we also examined other community finding methods (i.e. *Infomap*, and *Spectral Clustering*) and found we obtained the best results by using the Louvain algorithm. We provide our best performing network-based community features that we used in Table 2.4. Of particular interest, we found for individual v that features relating to the size of the largest connected component resulting v' removal of his/her connected component was useful. Another interesting pair of features we noted for both group and gang were the number of edges from members of that group/gang to a different group or gang. We hypothesize that the utility of these features is a result of conflicts between groups/gangs they are connected to as well as the spread of violence amongst different groups (i.e. if two groups are closely connected, one may conduct violent activities on behalf of the other).

Network-based path features. We looked at several features that leveraged the paths in the network by adopting three common node metrics from the literature: betweenness, closeness [26], and shell-number [63] as well as a propagation process based on a deterministic tipping model [30]. The features are listed in Table 2.5. We examined our modified definitions of closeness, betweenness, and shell number where C was a single element of $\mathcal{V}$, where $C = \mathcal{V}$ and where $C = \emptyset$ (which provides the standard definitions of these measures). Our intuition was that individuals nearer in the network to other violent individuals would

also tend to be more violent - and we found several interesting relationships such as that for closeness (where $C = V_{\mathcal{V}}$) discussed in section 2.5.1 when we run the classifier on each feature group. Shell number and the propagation process were used to capture the idea of the spread of violence (as shell number was previously shown to correspond with "spreaders" in various network epidemic models [37]). For the propagation process, we set the threshold (κ) equal to two, three, four, five, and six. Further details on shell number and the propogation process can be found in the appendix.

Geographic Features

Geographic features capture the information related to the location of a crime incident. The intuition is that the individuals who commit crimes in violent districts are more likely to become violent than the others. We found that the beat the individual has committed a crime in is an important feature for our problem. This is in accordance with previous well known literature in criminology [10, 57] which studies spatio-temporal modeling of criminal behavior. The complete list is shown in Table 2.6.

Table 2.6: Geographic Features

Name	**Definition**
District Frequency	$\lvert\{(t, v')\ s.t.\ arr^t_{v'} = \mathsf{true} \wedge \exists t'\ s.t.\ distr^t_{v'} = distr^{t'}_v\}\rvert$
Beat Frequency	$\lvert\{(t, v')\ s.t.\ arr^t_{v'} = \mathsf{true} \wedge \exists t'\ s.t.\ beat^t_{v'} = beat^{t'}_v\}\rvert$
Beat Violence	$\lvert\{(t, v')\ s.t.\ arr^t_{v'} = \mathsf{true} \wedge \mathcal{V}^t_{v'} \neq \emptyset \wedge \exists t'\ s.t.\ beat^t_{v'} = beat^{t'}_v\}\rvert$
District Violence	$\lvert\{(t, v')\ s.t.\ arr^t_{v'} = \mathsf{true} \wedge \mathcal{V}^t_{v'} \neq \emptyset \wedge \exists t'\ s.t.\ distr^t_{v'} = distr^{t'}_v\}\rvert$

Temporal Features

We considered couple of temporal features: average interval month and number of violent groups. Average interval time considers the average time duration of consecutive arrests of the offender. The other feature, which we examine, is number of violent groups appeared over time in the environment. We examined that the number of violent groups has been an important temporal aspect for identifying the violent criminals. The key intuition here is, if at least one member of the offender's groups (formed over time) is violent then we consider the offender as a part of that violent group. For an individual v, we define the partially ordered set $t_C^v = \{t \textit{ s.t. } arr_v^t = \mathsf{true} \wedge V_C^t \neq \emptyset\}$ (intuitively the set of the time points where v has committed at least on of the crimes in C.) We also define $\Delta_i^v(C) = t_i^v - t_{i-1}^v$ for each $t_i^v \in t_C^v$. Considering these definitions, we formally define the temporal features in Table 2.7.

Table 2.7: Temporal Features

Name	**Definition**
Average interval time (w.r.t. C)	$\sum_i \Delta_i^v(C)/\lvert t_C^v\rvert$
Number of violent groups	$\lvert\{t \; s.t. \; arr_v^t = \mathsf{true} \wedge$ $\exists v' \; s.t. \; arr_{v'}^t = \mathsf{true} \wedge$ $\mathcal{V}_v^t \neq \emptyset \wedge$ $v' \in N_v^t\}\rvert$

2.5 Experimental Results

In this section, we review the results of our experiments. We looked at two types: experiments where the entire co-offender network is known before-hand (Section 2.5.1) and experiments where the network is discovered over time (Section 2.5.2). The intuition

behind the experiments where the co-offender network is known is that the police often have additional information to augment co-arrestee data. This information can include informant reporting, observed individuals interacting by patrolmen, intelligence reporting, and information discovered on social media and the Internet. In our second type of experiment we discover the network over time in an effort to mimic real-world operations - however, we also show that this makes the problem more difficult as it reduces the power of neighborhood-based and network-based features. Based on our discussions with the Chicago Police, we believe that real-world results will most likely fall somewhere between these two experiments. Operationally, we will not have full arrest data, but the aforementioned augmenting data sources are available (even though we did not have access to them for our experiments).

2.5.1 Known Co-offender Network

In this experiment we assume that the entire offender network is known. In other words, to compute the features for each vertex v, we assume that the set $\mathcal{V}_v$ is unknown while the rest of the network is observable. In here we compared our approach with *THH* but not with the *PVA* as we do not utilize time. In each of the experiments described in this section, we conduct 10-fold cross validation. We consider the result of each approach as a set of nodes that the approach considers to be a set of potentially violent individuals. Our primary metrics are precision (fraction of reported violent individual who were actually violent in the dataset), recall (fraction of violent individuals in the dataset reported by the approach), F1 (the harmonic mean of precision and recall) and area under the curve. We conduct two types of experiments: first, we study classification performance using only features within a given category (neighborhood, network, temporal, and geographic), then we study the classification performance when the entire feature set is used but with various different classification algorithms and compare the result to *THH*.

Classification using single feature categories. Here we describe classification results using single feature categories. In this set of experiments, we use a random forest classifier (which we will later show provides the best performance of the classifiers that we examined). Fig. 2.5 shows the performance of RF for the described categories. The network-based features are highly-correlated to violent behavior with average F1 value of 0.72 compared to 0.63 for neighborhood, 0.21 for geographic, and 0.03 for temporal features. In Fig. 2.6, we show the performance of a feature from each category to classify violent vs. non violent crimes; the performance of each example is a good indicator of the performance of its category.

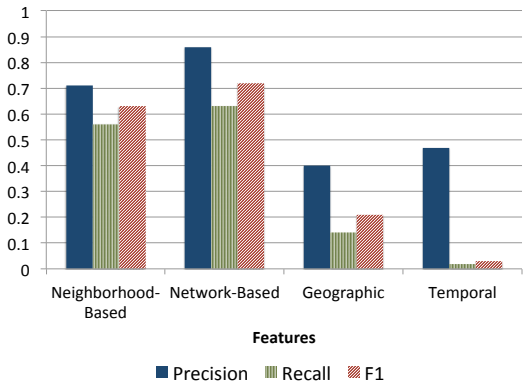
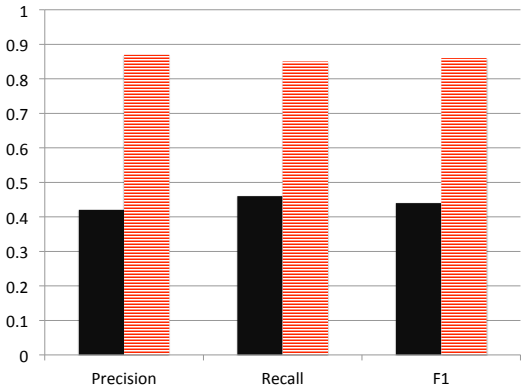
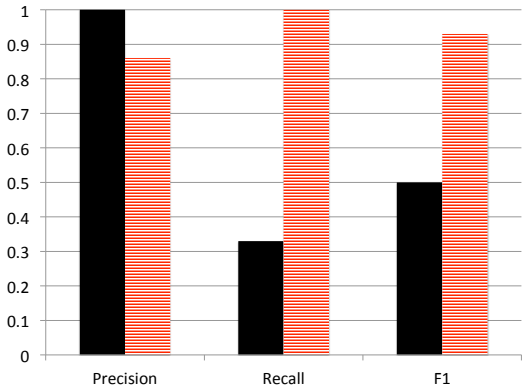


Fig. 2.5: Precision, Recall, and F1 Comparison Between Each Group of Features.

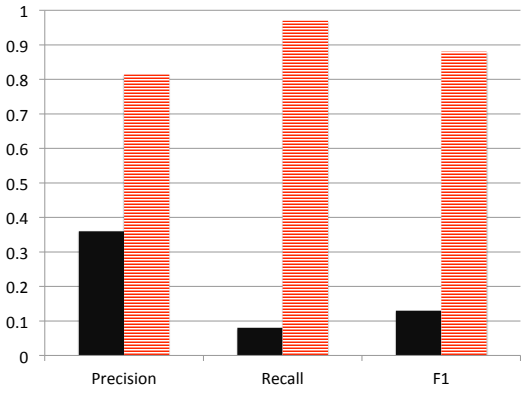
Classification comparison. Table 2.8 shows the performance of different classification algorithms. According to Table 2.8, RF provides the best performance (F1=0.83); we also note that using SMOTE for RF, did not improve this result. Fig. 2.7 shows that our algorithm outperforms *THH*. The performance of our features are also illustrated in Fig. 2.8. The area under the curve (AUC) of applying all features is 0.98 – a higher overall accuracy. The AUC for network-based, neighborhood-based, geographic, and temporal categories are 0.92, 0.91, 0.65, and 0.7 respectively. This indicates the importance of network features for this classification task.



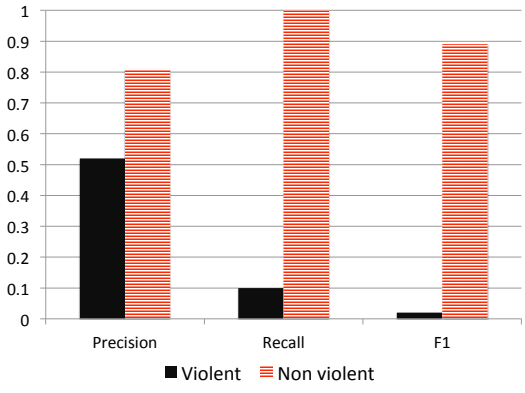
(a) Neighborhood-based: Minority of 1-hop and Majority of 2-hop Neighbors Committing a Crime in C.



(b) Network-based: Closeness (w.r.t. $\mathcal{V}$)



(c) Geographic: Beat Violence



(d) Temporal: Average Interval Months

Fig. 2.6: Example Features from Each Category.

2.5.2 Co-offender Network Emerges over Time

In this section, we present a more difficult experiment - where the co-arrestee network is discovered over time (by virtue of arrests). To simulate this phenomenon, we split our data into two disjoint sets: the first set for learning and identification, and the second one for measuring the performance. We do monthly split and start from February 2013. To illustrate

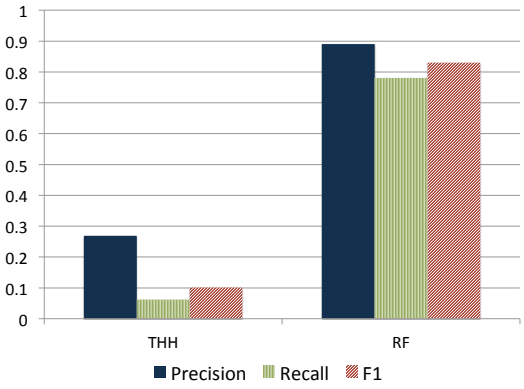


Fig. 2.7: Performance Comparison Between *THH* and *RF* in K-fold Cross Validation.

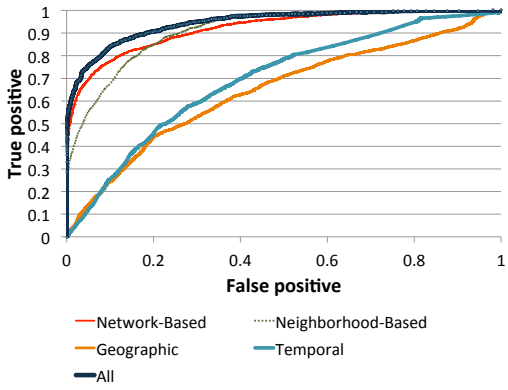


Fig. 2.8: ROC Curve for Each Feature Set.

the difficulty of this test, we show the number of nodes, edges, and violent individuals per month in Fig. 2.9. We note that in the early months, we are missing much of the graphical data (over 40% of nodes and edges in the first two months) - hence making many of our features less effective. However, as the months progress, there are less violent individuals to identify (due to the temporal nature of the dataset) - hence amplifying the data imbalance as time progresses.

In these experiments, we compared our approach using random forests with the full feature set to *THH* and *PVA*. We measure precision, recall, F1, number of true positives, and number of false positives and display the results in Fig.s 2.10 and 2.11. In *FRF* (Filtered

Table 2.8: K-fold Cross Validation

Method	Precision	Recall	F1
RF	0.89	0.78	0.83
RF w. SMOTE	0.86	0.78	0.82
NB	0.45	0.49	0.47
LR	0.68	0.49	0.57
DT	0.71	0.66	0.68
NN	0.64	0.57	0.6
SVM	0.73	0.2	0.31

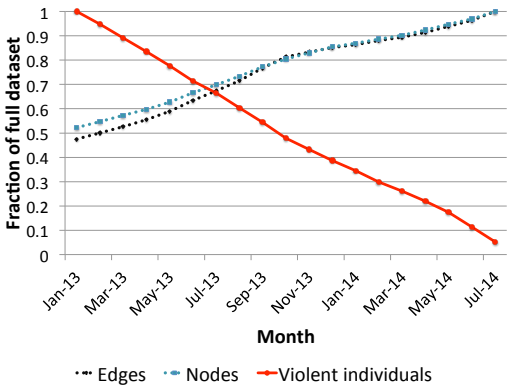
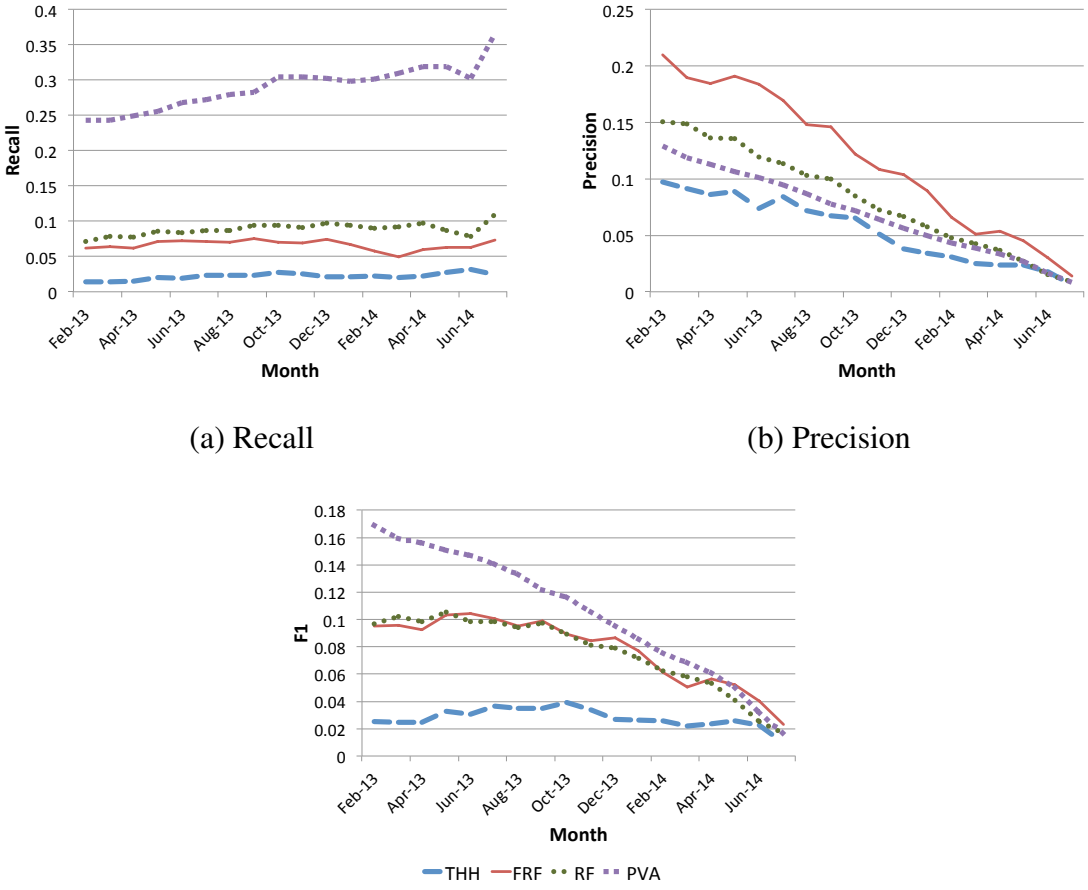


Fig. 2.9: Number of Nodes, Edges, and Violent Individuals over Time. More Training Data, Less Offenders to Identify.

Random Forest) we filter the offenders who have not committed any crime in the last 200 days. This simple heuristic increase the precision drastically while preserving the recall. The main advantage of our method, besides the high precision, is its ability to significantly reduce the population of potentially violent offenders when compared to PVA - which for each month had between 1813 and 3571 false positives. Fig. 2.11 compares the number of true and false positives instances for all the approaches for each month except PVA (PVA

was omitted due to readability because of the large amount of false positives). While the F1 measure for *PVA* is higher than that of the others, the large number of false positives prevents the law enforcement from using it effectively in practice. Furthermore, as time progresses, PVA likely rises in recall due to the drop in the number of violent criminals to predict.



(a) Recall

(b) Precision

(c) F1 score

Fig. 2.10: Performance of Different Approaches over Time

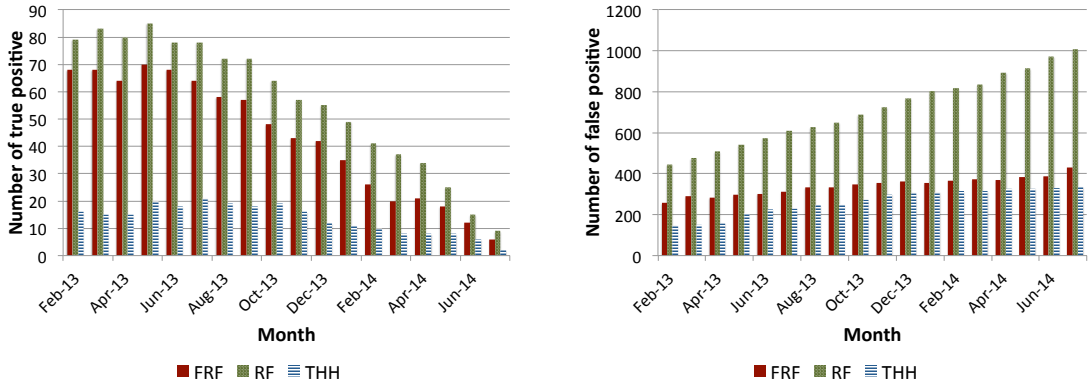


Fig. 2.11: Number of True and False Positive Instances.

2.6 Related Work

Though we believe that the prediction of violent offenders using co-offender social networks is new, there has previously been work on both co-offender networks in general as well as crime forecasting. In this section, we briefly review some of the relevant contributions in both of these areas.

There has been much previous work on co-offender networks. The earlier work that studied these special social networks primarily came from the criminology literature. For instance, [44] utilizes social network analysis techniques to study several case studies where the social network of the criminal organization was known. In [42], the authors study the stability of these networks change over time. More recently, graphical features derived from networks comprised of both offenders and victims has been shown to be related to the the probability of an individual becoming a victim of a violent crime [48, 50]. Previous work has also looked at the relationship between network structure and geography [49] and has leveraged both network and geographic features to predict criminal relationships [79] as well as influence gang members to dis-enroll [70]. There have also been several software tools developed for conducting a wide-range of analysis on co-offender networks including

CrimeFighter [55], CrimeLink [62], and ORCA [51]. However, our work departs from this is that we are looking to leverage the network topology and other features to identify violent offenders - which was not studied in any of the previous work.

There has also been a large amount of work on crime forecasting (i.e. [28, 41]) though historically, this work has relied on spatio-temporal modeling of criminal behavior [10, 57] or was designed to identify suspects for specific crimes [80, 46]. None of this previous work was designed to identify future violent offenders nor did it leverage social network structure.

2.7 Conclusion

In this chapter, we studied induction and deduction inferences, exploring the problem of identifying repeat offenders who will commit a violent crime. We showed a strong relationship between network-based features and whether a criminal will commit a violent offense providing an unbiased F1 score of 0.83 in our cross-validation experiment where we assumed that the underlying network was known. When we moved to the case where the network was discovered over time, our method significantly outperformed baseline approaches increasing precision and recall. In the next chapter, we study induction and abduction inferences.

Chapter 3

MISSING PERSON INTELLIGENCE SYNTHESIS TOOLKIT: A DATA-DRIVEN GEOSPATIAL ABDUCTIVE REASONING

3.1 Introduction

There are approximately hundred thousands missing person cases each and every year in the USA for the past twenty five years [14]. According to a review of missing and unidentified persons cases in 2008 [43] most cases resolve within a few days or week; however, there are instances that remain unsolved for decades or longer. In 2016, more than 647K people went missing from which approximately 3K were not located at all [14]. According to National Crime Information Center report, 76% of the total entries in 2017 are deceased [15].

The non-profit organization known as the Find Me Group (FMG), led by former law enforcement professionals, is dedicated to solving or resolving these cases. The group was founded by retired U.S. Drug Enforcement Agency (DEA) Special Agent J.E. "Kelly" Snyder in 2002 and consists of current and retired law enforcement officers with a wide-range of investigative expertise, including but not limited to linguistics, handwriting analysis, body language, missing person/homicide experience and search-and-rescue field management skills. The FMG has trained experts/sources that provide detailed location information where missing individuals can be found. Many of these experts have the ability to provide GPS coordinates to locate missing persons with a varying level of success. Their commitment and mission is to work collectively with law enforcement agencies to bring resolution to unresolved disappearances and homicides. The FMG focus/goal is to provide accurate location information in a timely manner and minimize the potential of finding the victim

deceased. Thirty canine handlers certified in tracking, scent and cadaver complements the FMG and has led to instances where the person in questions was located. This non-profit operates with limited resources (e.g., manpower) - so it must use its volunteer assets in a highly efficient manner.

This chapter introduces the Missing Person Intelligence Synthesis Toolkit (MIST) which leverages a data-driven variant of geospatial abductive inference [74]. This system takes search locations provided by a group of experts and rank-orders them based on the probability assigned to areas based on the prior performance of the experts taken as a group. We evaluate our approach compared to the current practices employed by the FMG and found it significantly reduces the search area. In 29 cases examined in our experiments (on real-world data provided by FMG), we found our approach to be able to reduce total search area by a total of 53 square miles for standard searches and by 55 square miles when dog team assets obtain a detection. This reduction is significant for the following reasons:

- **Reduction in time to locate missing persons.** In most of the cases, we achieved reduction of 1 to 15 and 2 to 56 square miles in search areas 1×1 and 2×2 respectively. As 3-5 square miles are searched on a typical day (terrain dependent), such a reduction can potentially increase the chance of a missing person being found alive.

- **Reduction in direct costs.** During a search, FMG spends approximately $\$2200$ per day. In all tests, our approach reduced the search area in the majority of cases which can be interpreted as a reduction in direct costs.

- **Reduction in indirect costs.** FMG relies extensively on volunteers to augment searches. During searches, these individuals often lose earnings from their day job or small business. As many volunteers also perform consulting or other services to law enforcement, longer searches lead to loss of revenue and opportunity. In one case, a volunteer estimated a loss of $15K. Again, our approach leads to a consistent

Table 3.1: Summary of the Results. Number of Cases, Average Reduction (mi^2) and Total Reduction (mi^2) in Search Area for the Two Approaches Considering Each Potential Location Is Centered in Two Different Size of Blocks.

Name	**Block Size**	**Cases**	**Avg Reduction**	**Reduction**
Double distance integer program (Section 3.6.1)	1×1	23	2.3	53
	2×2	24	7.62	183
Consideration of dog team detections (Section 3.6.2)	1×1	26	2.12	55
	2×2	29	9.28	269

reduction in the search area - hence reducing these indirect costs.

Specifically, we contribute an extension to geospatial abduction [74] that leverages historical data of individual experts. We also create new algorithms to learn the parameters of a geospatial abduction model from data based on integer programming. We then evaluate these algorithms on real-world data provided by the FMG under a variety of different settings. This approach learns the pattern of each reporter independently and is able to overcome outliers if any. It also performs well on limited data. Table 3.1 summarizes the results for the two proposed approaches considering each potential location is centered in 1×1 or 2×2 blocks.

We note that this research was done in collaboration with the FMG to ensure operational relevance. As such is the case, we also briefly describe our user interface for MIST. MIST needs a set of historical data including different cases and for each case, a set of potential locations (GPS coordinates) associated with reporters to learn the pattern of reporters. Then for any new case, it gets a set of potential locations and rank-orders them.

In this chapter, we formulate the problem of "finding missing person" with respect to information provided by FMG's experts, formally as a variant of the geospatial abduction

problem (GAP) [72]. GAP refers to the detection of unobserved partner locations (in this work, the location of a missing person) that best explain a set of observed phenomenon with known geographic information. To account for the key nuances of "finding missing person" problem though, we extended the GAP framework to better suit this domain. In particular, we extend the GAP formalism with a data-driven model - accounting for the previous performance of experts aiding in the missing person cases. We list the unique characteristics of our framework here. Later in the next section, we provide our technical approach to each.

1. **Explanation Size.** One key difference between "finding missing person" problem and other GAP instances, is that our explanation (the result of a GAP inference algorithm) only consists of a single related location (i.e., the location of the missing person) corresponding to the phenomenon under study. This differs from returning a set of k locations in the previously-introduced GAP formalisms. Consequently, here, an explanation will consist of a single point, which in turn lead us to explore a non-deterministic version of the original explanation.

2. **Distance Constraints.** In the original GAP formalism, each observed geospatial phenomenon is related to unobserved "partner" points through a distance constraint - (α,β) where α is the minimum distance between an observation and partner and β is the maximum distance. As described, this pair of constraints was the same for *all* observations. However, in the missing persons problem, each observation corresponds to a different domain expert - and hence has a different (α,β) constraint pair. Further, we study how this is best learned from data, as well as "soften" the constraint - assigning a probability of the partner point being less than α, between distances α and β, and greater than distance β from an observation.

3. **Uncertainty.** As we learn the (α,β) distance constraints for each observation and

associate corresponding probabilities from historical data, it makes sense that the inference step is treated probabilistically - which differs from the original deterministic GAP framework. Further, this enables us to rank the potential partner locations (again, as an explanation consists of one point, ranking search locations is more useful in a practical sense).

4. **Independent Observations.** In the original GAP framework, independence amongst the observations was not an assumption in the framework. However, FMG compartmentalizes the information from their law enforcement experts from one another in a manner to obtain independent reporting. Hence, we make this assumption in this chapter and it is supported by our experimental results.

We note there have been other data-driven approaches in the past for geospatial reasoning based on a historical data (see our summary of this line of research in Section 3.7). However, in general, these approaches rely on the sufficiently larger-sized corpus of training data as compared with geospatial abduction-based methods. For instance, the average number of cases reported on by an individual in our application is 3.66. It is 5.13 if we only consider the reporters that have participated more than once. Moreover, if the missing person's location is not within a few miles of the reported locations, MIST is not able to locate the missing person. This situation also cannot be handled by the FMG and is thus beyond the scope of this chapter.

Currently, the FMG uses a simple heuristic to rank-order potential search locations for a missing person (we describe this later in Section 3.4). Once ranked, FMG leverages a variety of assets. Fig. 3.1 depicts a recently searched area for a case. It represents a screen shot of the tracks from the GPS units that the dogs wear as well as the handheld units that the searchers wear. This shows several dog tracks and the human tracks. The green, dark blue, magenta represent three dogs, the grey and red represent two human searchers. The

teal track is a trailing dog, ascertaining a direction of travel. The straight lines tend to be humans and the rapidly changing direction lines are dogs as they grid around the humans. Fig. 3.2 shows real-world examples of how the FMG practices in an undisclosed location.

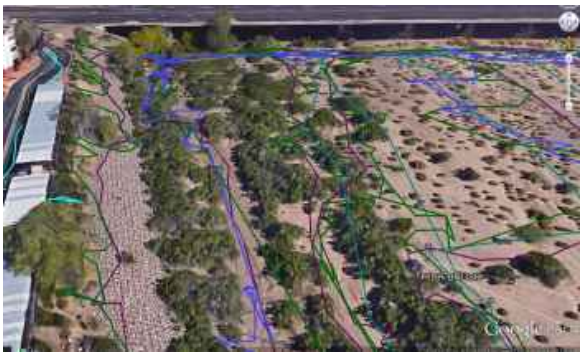


Fig. 3.1: Screen Shot of the Tracks from the GPS Units.

(a) (b)

Fig. 3.2: (a) Picture of the Search Area Taken from the Plane. (b) Search Team.

The rest of the chapter is organized as follows. In Section 3.2 we provide the technical preliminaries. We discuss our data-driven extension in Section 3.3. In Section 3.4, we detail our algorithmic approach. We introduce our dataset and conduct data analysis in Section 3.5. Next, we discuss the experimental results in Section 3.6. We review the related work in Section 3.7. We conclude the chapter by presenting future research directions.

3.2 Technical Preliminaries

In this section, we briefly explain geospatial abductive inference [74], and introduce our new (introduced in this chapter) data-driven probabilistic extension. We show how this extension was used to address the unique characteristics of the missing person location problem.

In general, *abduction* or *abductive inference* refers to a type of logic or reasoning to derive plausible explanations for a given set of facts [53]. Abduction has been extensively studied in medicine [53, 54], fault diagnosis [19], belief revision [47], database updates [34, 20] and AI planning [23]. Two major existing theories of abduction include logic-based abduction [24] and set-covering abduction [12]. Though none of the above papers takes into account spatial inference, [75] presents a logical formalism dealing with objects' spatial occupancy, while [61] describes the construction of a qualitative spatial reasoning system based on sensor data from a mobile robot.

Geospatial abduction problem (GAP) [72], on the other hand, refers to the problem of identifying unobserved partner locations (i.e., the location of a missing person) that best explain a set of the observed phenomenon with known geographic locations. *Geospatial abduction* was first introduced in [73] and later extended in [74, 71, 69, 68]. More formally, each GAP consists of three major elements [72]: (1) observations: a set of observations that explain the locations associated with the event under study (e.g., in this application, the locations reported by the domain experts), (2) distance constraints: a pair $(\alpha, \beta) \in \mathbb{R}$ corresponding to lower and upper bounds on the distances between observation and partner location and, (3) feasibility predicate: this allows to specify whether an area on the map is a potential location for a partner.

Next, we present the notations and definitions used throughout the chapter, and review the geospatial abduction framework of [72]. In the next section, we describe specialized

extensions that were necessary to study our problem. First, without loss of generality, we assume throughout the paper that a map (resp. space) is represented by a discrete two-dimensional grid of size $M \times N$, defined as follows:

Definition 3.2.1. (Space). *Given natural numbers M, N, the space $\mathcal{S}$ is the set* $[1, \ldots, M] \times [1, \ldots, N]$.

Associated with the space is a *distance function* $d : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}^+$ that satisfies the normal distance axioms: $d(p_i, p_i) = 0$, $d(p_i, p_j) = d(p_j, p_i)$, and $d(p_i, p_j) \leq d(p_i, p_q) + d(p_q, p_j)$.

Note that we use o to represent the observer (source of information) and p_o to represent the location he/she reported (which differs slightly from the original framework). From these observations (reports), the corresponding unobserved phenomenon is the actual location of the missing person. In the original framework, the explanation consisted of geographic locations that were located at least distance α and no more than distance β away from each observation. In this work, we generalize this notion by providing α,β pair for each observer - denoted α_o,β_o.

Definition 3.2.2. (Feasibility Function). *A feasibility function* feas *is defined as* feas $: \mathcal{S} \rightarrow \{$*True*, *False*$\}$.

A key use for the feasibility function here is for an initial reduction of the search space by the FMG. This is due to the fact that missing person reports often span a large area and an initial reduction is necessary for practical reasons. For example, in this chapter, the distances between found locations and last-seen locations are in range (0.026, 863.4) miles. Moreover, the median is 2.7 miles which need the search area of 29 square miles. Also, according to a report in UK [11], 66% of cases are within 5 miles of their home which is in an area of about 78 square miles. Searching the whole area is thus impossible in practice due to the resource limitation. An obvious future direction would be to utilize a probabilistic variant of the feasibility function - which would assign a prior probability to a location for a missing

person. However, in this application, it is unclear where such a distribution would come from. Further, as the search space is relatively large when compared to FMG resources, the deterministic version of this definition is more appropriate for operational reasons.

Due to the resource constraints and large areas for which reports are spread, FMG only searches areas for which there is a report (i.e., potential locations for any missing person) by its experts. As we will describe in Section 3.4, they search a 1×1 mile square surrounding a location reported by an observer. As such is the case, we shall assume the following feasibility function throughout this chapter:

$$\mathsf{feas}(p) = \begin{cases} \mathsf{True} & \text{if } p \in \mathcal{O} \\ \mathsf{False} & \text{otherwise} \end{cases} \tag{3.1}$$

Unless otherwise noted, we shall assume the above function is used for feasibility and hence the subset of the space considered will be the points in $\mathcal{O}$.

We now provide the important definition of an *explanation*. Intuitively, for a given set of points $\{p_1, \ldots, p_{|\mathcal{O}|}\}$ reported by observers in $\mathcal{O}$, an explanation is a set of points $\mathcal{E}$ such that every point in this set is feasible and for every observation, there is a point in $\mathcal{E}$ that is at least α units away from the observation, but no more than β units from the

Definition 3.2.3. (α,β) Explanation). *Suppose $\mathcal{O}$ is the set of observations, $\mathcal{E}$ is a finite set of points in $\mathcal{S}$, and $0 \leq \alpha, \beta \leq 1$ are two real numbers. $\mathcal{E}$ is said to be an (α, β) explanation of $\mathcal{O}$ iff:*

- *$p \in \mathcal{E}$ implies that $\mathsf{feas}(p) = \mathsf{True}$, i.e., all points in $\mathcal{E}$ are feasible.*
- *$(\forall o \in \mathcal{O})(\exists p \in \mathcal{E})\ \alpha \leq d(p, o) \leq \beta$, i.e., every observation is neither too close nor too far from some point in $\mathcal{E}$.*

Thus, an (α,β) explanation is a set of points. Each point must be feasible and every observation must have an analogous point in the explanation which is neither too close nor too far.

Again, we note that here an explanation will consist of a single point - the location of the missing person (found location). Hence, this deterministic definition of an explanation will not suffice - as in practice there will often not exist an explanation for a given problem instance. As such is the case, we extended this framework using a data-driven approach.

3.3 Data-driven Extensions

Here, we describe our data-driven probabilistic extension to the original GAP formalism. The framework extensions in this section were not previously introduced and are new in this chapter. In order to do so, we first introduce some preliminary notation. For point $p \in \mathcal{S}$, the random variable $\mathcal{P}_p$ denotes that the missing person was found at point p, so this is either true or false. We will use $\mathcal{P}_p$ as shorthand for $\mathcal{P}_p = \mathsf{True}$. For observer $o \in \mathcal{O}$ the random variable O_o can be assigned to one of the points in p. Based on this notation, we define an *explanation distribution*.

Definition 3.3.1 (Explanation Distribution). *Given a set of observers $\mathcal{O}$ and a set of reported locations by each observer $p_1, \ldots, p_o, \ldots, p_{|\mathcal{O}|}$, an **explanation distribution** is a probability distribution over all points in $\mathcal{S}$ - directly addressing characteristic 3 of this application (see Section 3.1). This distribution assigns the probability of a missing person being located at each point conditioned on the observers reporting their respective locations. Formally, it is written as $Pr(\mathcal{P}_p | \bigwedge_{o \in \mathcal{O}} \mathsf{O}_o = p_o)$.*

The key intuition is that if we are able to compute an explanation distribution, we can then rank-order points in the space by probability - and hence conserve search resources. Note that the explanation distribution is over all points - implying that there is precisely one location. While generalizations that allow for more than one location are possible in such a probabilistic framework, we keep the size at one due to the first characteristic of our problem (as described in Section 3.1).

In this chapter, we make an assumption of *distance primacy* meaning the distance constraints (α_o, β_o) relate the $\mathcal{P}_p$ with $\bigwedge_{o \in \mathcal{O}} \mathsf{O}_o = p_o$. Hence, we introduce another random variable, $\mathfrak{R}^{\beta_o}_{p,p'}$ which is true if $d(p, p') \leq \beta_o$ and false otherwise. Note that in the remainder of this section, we will use one distance constraint (β) for sake of brevity - though this idea can be extended for multiple distance constraints (as per characteristic 2 from Section 3.1). In fact, we leverage multiple distance constraints in our optimization procedure for parameter selection introduced later. Hence, by distance primacy, we have the following relationships.

$$Pr(\mathcal{P}_p | \bigwedge_{o \in \mathcal{O}} \mathsf{O}_o = p_o) = Pr(\mathcal{P}_p | \bigwedge_{o \in \mathcal{O}} \mathfrak{R}^{\beta}_{p,p_o}) \tag{3.2}$$

According to the Bayes' Theorem, this is equivalent to the following.

$$\frac{Pr(\mathcal{P}_p) \times Pr(\bigwedge_{o \in \mathcal{O}} \mathfrak{R}^{\beta}_{p,p_o} | \mathcal{P}_p)}{Pr(\bigwedge_{o \in \mathcal{O}} \mathfrak{R}^{\beta}_{p,p_o})} \tag{3.3}$$

However, by characteristic 4, we assume that the observers report information independently (conditioned on the location where the missing person is actually located), which gives us the following.

$$\frac{Pr(\mathcal{P}_p) \times \prod_{o \in \mathcal{O}} Pr(\mathfrak{R}^{\beta}_{p,p_o} | \mathcal{P}_p)}{Pr(\bigwedge_{o \in \mathcal{O}} \mathfrak{R}^{\beta}_{p,p_o})} \tag{3.4}$$

Due to our application, we will not consider the prior probability $Pr(\mathcal{P}_p)$ as each missing person case occurs in a different geographic location - and due to the wide range of cases that span multiple countries, data supporting a realistic, informed prior is highly sparse. As such, we will treat this prior probability as a uniform distribution over all locations. Further, for notational simplicity, we shall use the notation ρ^{β}_o for the quantity $Pr(\mathfrak{R}^{\beta}_{p,p_o} = \mathsf{True} | \mathcal{P}_p = \mathsf{True})$. Therefore, we can rank points in the space based on the explanation distribution by simply considering their log-likelihood computed as follows:

$$\sum_{\substack{o \in \mathcal{O} \\ d(p,p_o) \leq \beta}} \log(\rho^{\beta}_o) + \sum_{\substack{o \in \mathcal{O} \\ d(p,p_o) > \beta}} \log(1 - \rho^{\beta}_o) \tag{3.5}$$

Hence, the inference step for this problem is straight-forward provided we know the values β and ρ_o^β for each observer $o \in \mathcal{O}$ (or similar parameters if considering more than one distance constraint). If we know the value β we can then compute ρ_o^β based on a corpus of historical data concerning the accuracy of reporter o. Given a corpus of previous cases for the observer C_o where the found location was p^c and the location reported by the observer was p_o^c, we can compute ρ_o^β as follows:

$$\rho_o^\beta = \frac{|\{c \in C_o \textit{ s.t. } d(p^c, p_o^c) \leq \beta\}|}{|C_o|} \tag{3.6}$$

Hence, we also adjust ρ_o^β to account for volume of the reporter's history to provide the effect of regularization. Note that the quantity $|C_o|$ will be small for reporters with a limited case history. Considering η_o as the portion of total number of cases in which observer o has participated, to the total number of cases, and ϵ as a non-negative parameter, we define $\rho_o^{\beta,\epsilon}$ as follows:

$$\rho_o^{\beta,\epsilon} = \rho_o^\beta - \epsilon \times (1 - \eta_o) \tag{3.7}$$

The situation is further complicated with multiple distance constraints. We propose an optimization approach to this problem in the next section.

3.4 Algorithmic Approach

In this section, we present our algorithmic approach to special case of geospatial abductive inference. First, we explain the method that FMG currently uses. Then, we provide our proposed optimization approach to solve the problem.

3.4.1 Existing Method

The FMG uses the following method to explore the missing person location. Given the reported locations provided by different observers, FMG initially creates a search area (grid) as follows. First, they draw building blocks (or boxes) of size 1×1 mile centered at each

reported location (note that depending on the situation, these boxes may overlap). Then, they search the entire grid in the following order. First, they search the larger areas which are created of the overlapping boxes, and if the missing person was not found, they explore the remaining boxes in the order of the observers' history (how well they did in the past). The whole process is repeated by extending the size of boxes to 2×2 miles, if the missing person was not located. Note that, we use the same grid in our proposed methods.

3.4.2 Proposed Methods

As described, for simplicity, we first elaborate on the required steps to calculate the best β_o for each observer. Then, we extend the idea for multiple distance constraints. Let $[\beta_o]$ be the set of possible error radii. Note that for C_o cases where observer reported a location, there are at most $|C_o|$ possible values for β_o. Hence, our goal is to select as a set of these distance constraints - one for each observer. We do this through an integer program - where for each observer $o \in \mathcal{O}$ and each associated distance constraint $\beta_o \in [\beta_o]$ we have an indicator variable X_{o,β_o} that is 1 if we use that value and zero otherwise. We shall refer to this as the *single distance integer program*. Hence, we find an assignment of values to these indicator variables in order to maximize the following quantity:

$$F_1 = \sum_{c\in C}\sum_{o\in\mathcal{O}}\sum_{\beta\in[\beta_o]}\Big[\delta_\beta(p^c,p_o^c)\times\log\rho_o^\beta\times X_{o,\beta}+ (1-\delta_\beta(p^c,p_o^c))\times\log(1-\rho_o^\beta)\times X_{o,\beta}\Big] \tag{3.8}$$

subject to the following constraints:

$$\forall X_{o,\beta}\in\{0,1\} \tag{3.9}$$

$$\forall o,\ \sum_{\beta\in[\beta_o]} X_{o,\beta}\le 1 \tag{3.10}$$

$$\sum_{o}\sum_{\beta\in[\beta_o]} X_{o,\beta} = k \tag{3.11}$$

where k is a cardinality that limits the number of reporters (which is set to a natural number in the range $1, \ldots, |\mathcal{O}|$), and $\delta_\beta(x, y)$ is defined as:

$$\delta_\beta(x,y) = \begin{cases} 1 & \text{if } d(x,y) \leq \beta \\ 0 & \text{if } d(x,y) > \beta \end{cases} \tag{3.12}$$

However, this equation will result in tendency toward selecting the largest distance constraints. This has the effect of not only maximizing the probability of the locations where the missing person was found, but also can increase the probability of other locations. Intuitively, we want to also minimize the following quantity:

$$\begin{aligned} F_2 = \sum_{c \in C} \sum_{o \in \mathcal{O}} \sum_{p \in \{\mathcal{S} \setminus p^c\}} \sum_{\beta \in [\beta_o]} \Big[& \delta_\beta(p, p_o^c) \times \log \rho_o^{'\beta} \times X_{o,\beta} + \\ & (1 - \delta_\beta(p, p_o^c)) \times \log(1 - \rho_o^{'\beta}) \times X_{o,\beta} \Big] \end{aligned} \tag{3.13}$$

Therefore, the objective function we seek to optimize is

$$L_1 = \max(F_1 - F_2) \tag{3.14}$$

Theorem 3.4.1. *Number of variables in single distance integer program is* $O(\mathit{avg}(|C_o|) \cdot |\mathcal{O}|)$.

Proof. *For any* $o \in \mathcal{O}$*, there are at most* $|C_o|$ *possible distance constraints. Considering* $\mathit{avg}(|C_o|)$ *as average number of cases reported by each reporter, the total number of variables is* $O(\mathit{avg}_o(|C_o|) \cdot |\mathcal{O}|)$.

We extend the previous formulation by allowing the objective function to find a pair of distance constraints for each reporter. We have experimentally found diminishing returns on performance (and in many cases increased complexity) with more than two constraints.

This will give us the *double distance integer program* as follows:

$$F_1' = \sum_{c\in C}\sum_{o\in\mathcal{O}}\sum_{\alpha\in[\beta_o]}\sum_{\substack{\beta\in[\beta_o]\\ \beta\geq\alpha}} \Big[\delta_\alpha(p^c,p_o^c)\times\log\rho_o^{\alpha,\epsilon}\times X_{o,\alpha,\beta}+$$
$$\Big(1-\delta_\alpha(p^c,p_o^c)\Big)\times\delta_\beta(p^c,p_o^c)\times\log\Big(\rho_o^{\beta,\epsilon}-\rho_o^{\alpha,\epsilon}\Big)\times X_{o,\alpha,\beta}+$$
$$(1-\delta_\beta(p^c,p_o^c))\times\log\Big(1-\rho_o^{\beta,\epsilon}\Big)\times X_{o,\alpha,\beta}\Big]$$

subject to the following constraints:

$$\forall X_{o,\alpha,\beta}\in\{0,1\}$$
$$\forall o,\ \sum_{\alpha,\beta\in[\beta_o]} X_{o,\alpha,\beta}\leq 1$$

Note that we have limited the selection of α for $\rho_o^{\alpha,\epsilon} > 0.5$ due to high confident distance selection. Likewise, we use the following objective function, to avoid bias toward selecting the largest β's.

$$L_2 = \max(F_1' - F_2') \tag{3.15}$$

where F_2' is defined as follows:

$$F_2' = \sum_{c\in C}\sum_{o\in\mathcal{O}}\sum_{\alpha\in[\beta_o]}\sum_{\substack{\beta\in[\beta_o]\\ \beta\geq\alpha}} \Big[\sum_{p\in\{\mathcal{S}\setminus p^c\}}\delta_\alpha(p,p_o^c)\times\log\rho_o^{'\alpha,\epsilon}\times X_{o,\alpha,\beta}+$$
$$\Big(1-\delta_\alpha(p,p_o^c)\Big)\times\delta_\beta(p,p_o^c)\times\log(\rho_o^{'\beta,\epsilon}-\rho_o^{'\alpha,\epsilon})\times X_{o,\alpha,\beta}+$$
$$\Big(1-\delta_\beta(p,p_o^c)\Big)\times\log\Big(1-\rho_o^{'\beta,\epsilon}\Big)\times X_{o,\alpha,\beta}\Big] \tag{3.16}$$

Theorem 3.4.2. *Number of variables in double distance integer program is* $O(\mathsf{avg}(|C_o|)^2\cdot|\mathcal{O}|)$.

Proof. *For any* $o\in\mathcal{O}$*, there are at most* ${}^{|C_o|(|C_o|-1)}/_2$ *possible distance constraints while choosing at most two. Considering* $\mathsf{avg}(|C_o|)$ *as average number of cases reported by each reporter, the total number of variables is* $O(\mathsf{avg}_o(|C_o|)^2\cdot|\mathcal{O}|)$.

While we obtained a significant reduction in the area searched by setting the cardinality constraint $k = \mathcal{O}$, we found that varying it would often lead to further improvement. We gradually increased the number of observers from one to the total number of observers and each time, we learned the distance constraints for the last added observers. In this method of optimization, we may choose a specific number of points in each iteration. The number of points added with each iteration can be determined based on available resources.

We also defined two heuristic to discriminate points with the same probability. In each iteration, we chose the point with highest probability. If there were more than one point, we applied the following heuristics: (1) we chose the point which had the maximum summation of the priors of the reporters in its 1×1 mile. (2) we chose the points which had the most of the reported locations by the reporters, in its 1×1 mile.

Algorithm 2 is a specific variant of restricted model. In this algorithm, in each iteration one point (i.e., representative of a 1×1 mile) is selected. Though we note that this can easily be adjusted in practice. If the area size we are able to search is larger than number of observers, we sort the representatives based on their probabilities. Then, we apply two heuristics to rank them (similar to Lines 11-19). To better understand how Algorithm 2 works, we will give an example next.

Example 3.4.1. *A 5×6 search grid with four observers is shown in Fig. 3.3. The description of each subfigure is as follows: (a) Given the observations $o_1 = (2, 2)$, $o_2 = (3, 4)$, $o_3 = (5, 2)$, and $o_4 = (6, 5)$, we would like to pick top 5 locations to search. (b) In the first iteration, best observer and its β is selected by Algorithm 2. Here, o_2 is picked and each cell is ranked based on that. Thus, $(3, 4)$ is ranked first. (c) In the next iteration, the second top observer is selected (o_3) as shown in red. Therefore, cells are prioritized based on the two observers o_2 and o_3. The cell $(4, 3)$ is picked as the second search point. (d) In the third iteration, o_1 is picked and the cell $(3, 3)$ is selected as the next searching point. (e) In the fourth iteration, all reporters play role in prioritizing the locations and the point $(4, 2)$ is*

Algorithm 2 Iterative Search Resource Allocation

1: **procedure** OPT-POINT-BY-POINT(A, c, $\mathcal{S}$, ρ) ▷ Train set A, Test case c
2: List $R = \emptyset$ ▷ Output
3: **for** $k \in [1, |\mathcal{O}_c|]$ **do** ▷ k is a constant value of the constraint
4: Find assignment of variables that optimize (3.15) w.r.t. (3.9 - 3.11)
5: $RP \leftarrow$ Order by (3.5) ▷ Ranked points RP
6: $RP \leftarrow RP \setminus R$
7: Pick $P \subseteq RP$ with largest probabilities
8: **if** P includes one point **then**
9: $R = R \cup P$
10: **else**
11: $p \leftarrow Heuristic(P)$
12: $R = R \cup \{p\}$
13: **return** R

picked. (f) In the last iteration, since there is no more observers to add, the next cell with highest probability is selected.

Theorem 3.4.3. *The time complexity of the algorithm (2) is* $O(\mathit{avg}(|C_{\mathcal{O}}|) \cdot \mathit{avg}(|\mathcal{O}_c|)^3 \cdot \mathit{avg}(|C_o|)^2)$.

Proof. *Running objective function Eq (3.15) takes* $O(\mathit{avg}(|C_{\mathcal{O}}|) \cdot \mathit{avg}(|\mathcal{O}_c|) \cdot \mathit{avg}(|C_o|)^2 \cdot \mathit{avg}(|\mathcal{S}|))$ *time where* $C_{\mathcal{O}}$ *is the set of cases that includes at least one of the observers from the test case. Space size is* $O(\mathit{avg}(|C_o|))$*; therefore, the objective function run time can be simplified to* $O(\mathit{avg}(|C_{\mathcal{O}}|) \cdot \mathit{avg}(|\mathcal{O}_c|)^2 \cdot \mathit{avg}(|C_o|)^2)$. *Line 5 also takes* $O(\mathit{avg}(|C_o|)^2)$ *and the running time of the remainder is* $O(\mathit{avg}(|\mathcal{O}_c|)^2)$. *It is limited by* $|\mathcal{O}_c|$ for *loop in line 3. Hence, the running time is* $O(\mathit{avg}(|C_{\mathcal{O}}|) \cdot \mathit{avg}(|\mathcal{O}_c|)^3 \cdot \mathit{avg}(|C_o|)^2)$.

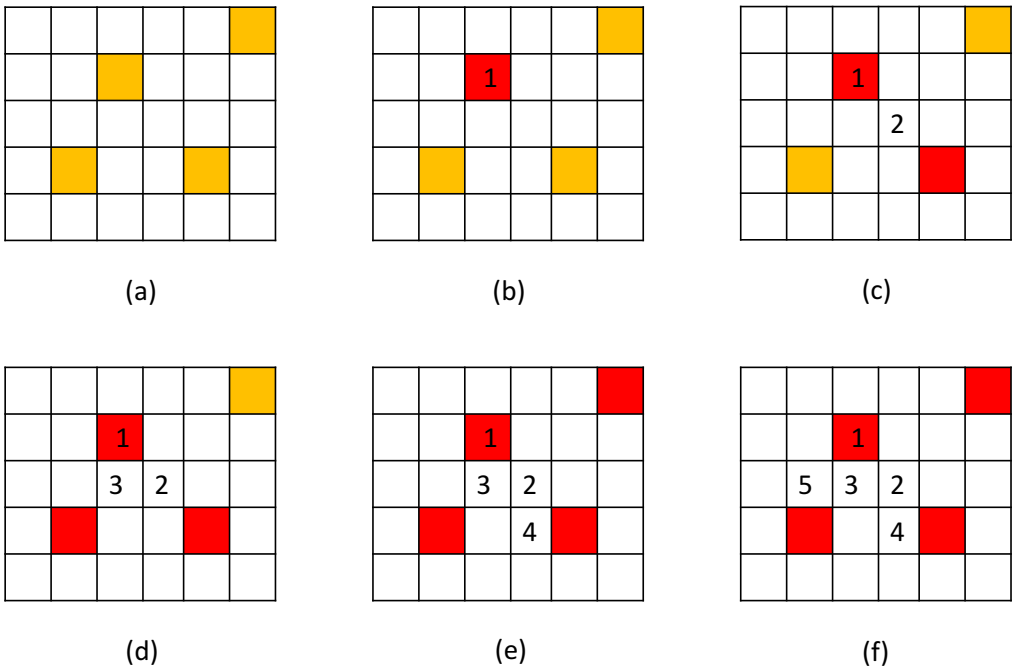


Fig. 3.3: A Toy Example Demonstrating How Algorithm 2 Works. Orange Cells Represent the Observations (Reported Locations) by the Observers (Reporters). Red Cells Represent the Observations That Are Picked by Our Algorithm and Numbers Depict the Corresponding Prioritized Cells to Search.

3.5 Missing Person Dataset

In this section, we describe our dataset and briefly discuss the observation made from our initial data analysis.

3.5.1 Overview

Our dataset includes cases (i.e., missing persons), found status (alive/deceased), found location (latitude and longitude), age and reason for disappearance as well as the potential locations (latitude and longitude) associated with the reporters/experts. Note we assume that the reporters/experts have considered all different aspects of the missing person including

age, gender, potential reason and health condition. Therefore, potential locations reflect different existing limitations for a given case. The description of this dataset is summarized in Fig. 3.4. About 86% of the FMG cases found deceased. Note that in some cases, we are aware of reports, but do not have the found location (p_o^c). In this work, we only have 29 cases with the known found locations used for the experiments. However, for the data analysis, the entire dataset is applied.

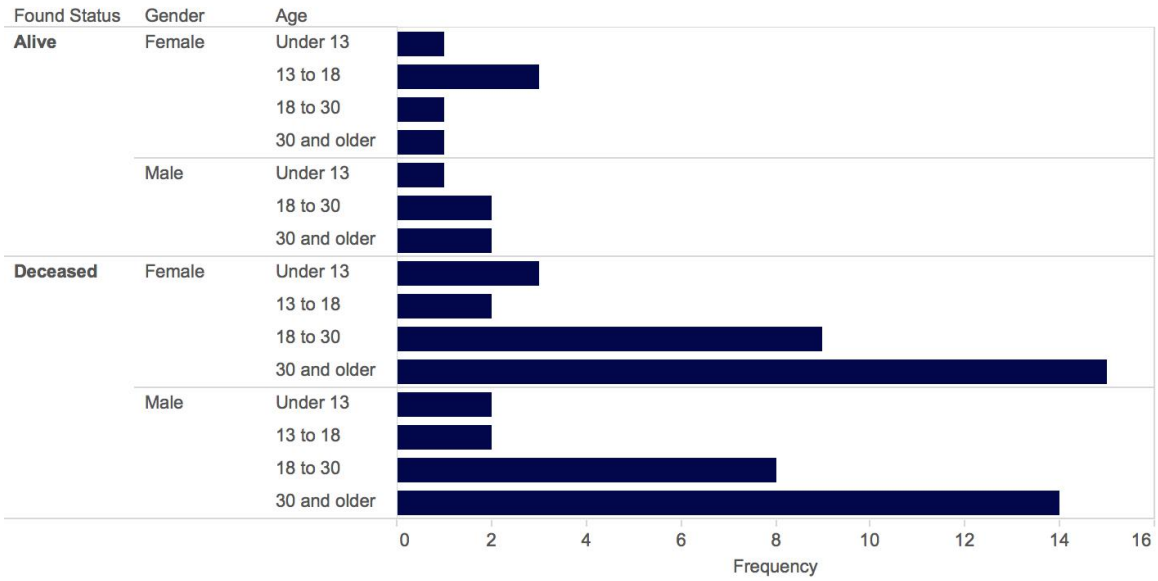


Fig. 3.4: Description of the Dataset

3.5.2 Data Analysis

The dataset consists of cases distributed all over the world. We split the U.S.-based cases into 4 regions, *west*, *midwest*, *northeast* and *south*, according to the United States Census Bureau. We further grouped together all cities outside the U.S. into one single category, namely, *international*. The distribution of cases across different regions is demonstrated in Fig. 3.5. As it is shown in the Fig., the west is dominated by Arizona and California, due to the large focus of FMG on these two states.

There are several known reasons of disappearance associated with the cases in our dataset including, *accidental*, *bipolar*, *drowning*, *foul play*, *natural*, *runaway*, *self-inflicted*, *staged* and *undetermined*. According to Fig. 3.6, 'foul play' is the dominant reason for disappearance. There are also different number of reporters for each case. The distribution of reporters with respect to the number of cases in which they participated is shown in Fig. 3.7a.

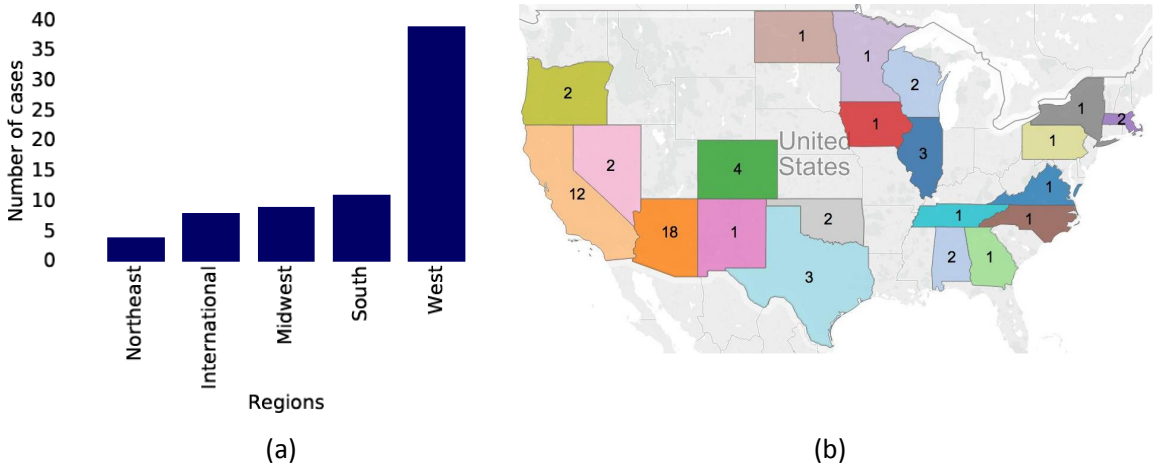


Fig. 3.5: Distribution of the Cases Across Different Regions of the US and International.

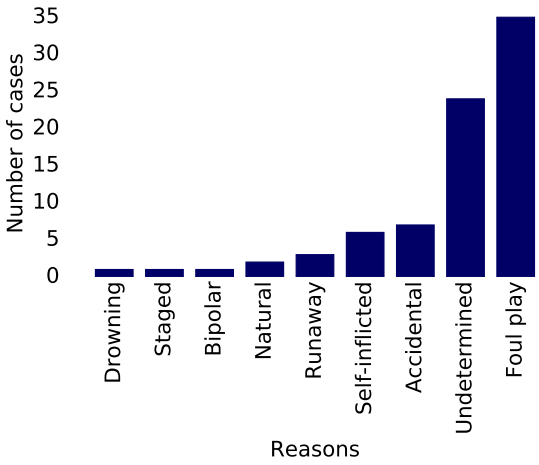
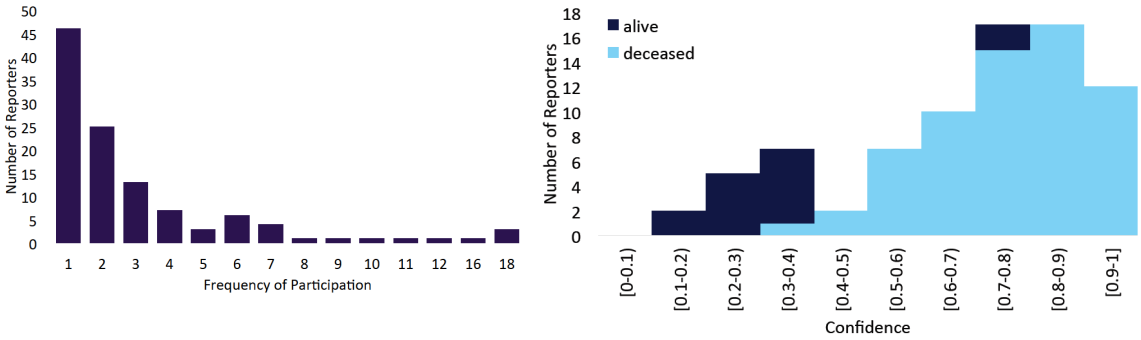


Fig. 3.6: Distribution of the Cases with Respect to the Probable Reasons.



(a) (b)

Fig. 3.7: a. Distribution of Frequency of Participation. b. Distribution of All Reporters with Respect to Their Confidence Values.

For the rest of our data analysis, we need to introduce some preliminary notation. We use the random variable g_A to denote if the missing person is found alive or not, so it is either true or false. We shall use $Pr(g_A = \mathsf{True}|o \text{ stated Alive})$ to denote the confidence of the observer o in reporting *Alive*. This confidence value shows the portion of the cases for which o has reported the missing person is *Alive* and the person was found *Alive*, to the total number of cases for which o has reported *Alive*. Likewise, we compute the confidence of o in reporting *Deceased*. The distribution of the reporters with respect to their confidence values is demonstrated in Fig. 3.7b. According to the Fig., most reporters' confidence values belong to the ranges of [0.3,0.4) for *alive* and [0.8,0.9) for *deceased* statuses.

We also define the ratio r_A as follows:

$$r_A = \frac{Pr(g_A = \mathsf{True}|\text{observer } o \text{ stated } \textit{Alive})}{Pr(g_A = \mathsf{True})} \tag{3.17}$$

This ratio demonstrates how much the observer o outperformed the prior probability $Pr(g_A = \mathsf{True})$ on *Alive*. Similarly, we use r_D for *Deceased* cases. The distributions of the reporters with respect to r_A and r_D are shown in Fig. 3.8. We note that as most are found dead, it is harder for the reporters to outperform the prior on *Deceased* compared to the *Alive*.

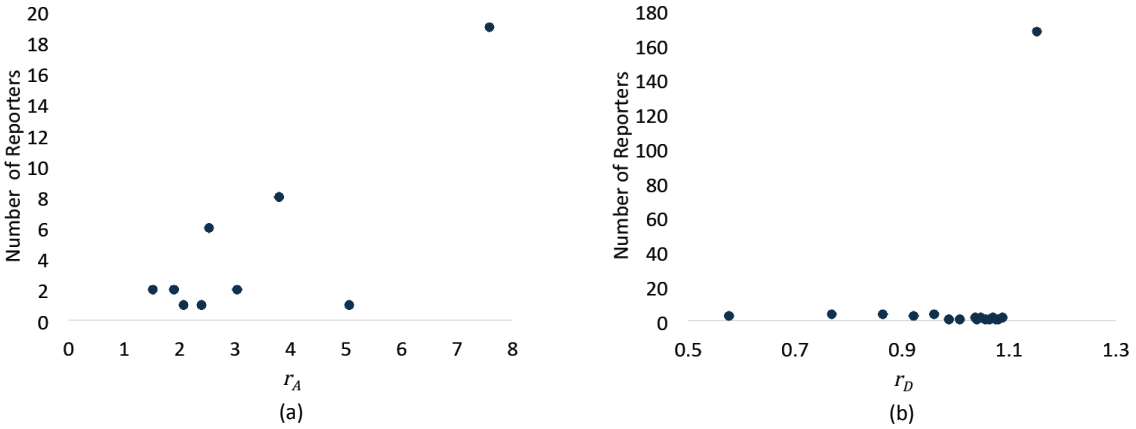


Fig. 3.8: The Distributions of the Reporters with Respect to r_A and r_D.

3.6 Experimental Results

This section reports on the experiments conducted to validate our approach. We note that the individual cases themselves are not related - hence we are justified in using leave-one-out cross validation in our experiments. Specifically, for each case in the experiments, we learn a *different* model using all of the other cases. We first compare the methods for restricted (without dog) and unrestricted (with dog) searches and then discuss the sensitivity of the parameter.

3.6.1 Area Reduction

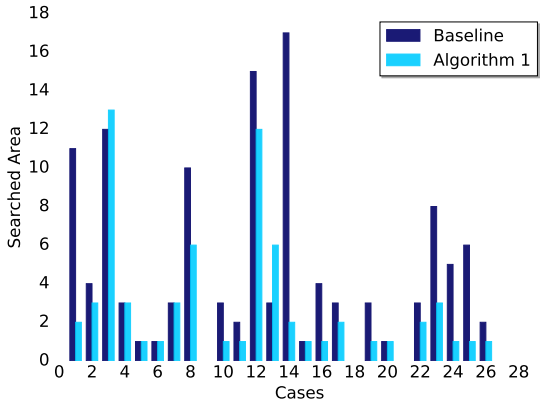
In this section, we examine how our approach can be used to reduce the area searched by the Find Me Group (FMG) over the baseline. Note that we use the same search grid for both approaches (double distance integer program with Algorithm 1 and $\epsilon = 0.05$) and the baseline (described in Section 3.4.1). The difference is the strategy for searching the grid. The FMG strategy is to find the overlaps between the reported locations and divide the search grid to the contiguous areas. The larger the size of each area is, the higher priority it gains during search. Fig. 3.9 shows the reduction of area based on our approach (double

distance integer program with Algorithm 1 and $\epsilon = 0.05$) when compared to the baseline. We examine this with grid squares of 1×1 miles and 2×2 miles. In grid squares of 1×1 miles, the missing person was located for 23 cases. Our approach achieved area reduction in 15 cases - reducing the search area by 3.8 square miles on average. In the 2 cases where our method caused the search area to increase, the increase was by 2 square miles on average. This contrasts with the cases where the area was reduced - reducing the search area by up to 15 square miles. For the 23 cases, the average and total reduction was 2.3 and 53 mile square respectively ($t(23) = 1.93, p < 0.03$). We also calculated the probability of locating missing person by searching the same size of areas randomly in Fig. 3.10. The average probability of our approach and baseline are 0.23 and 0.44 respectively (see Fig. 3.10a).

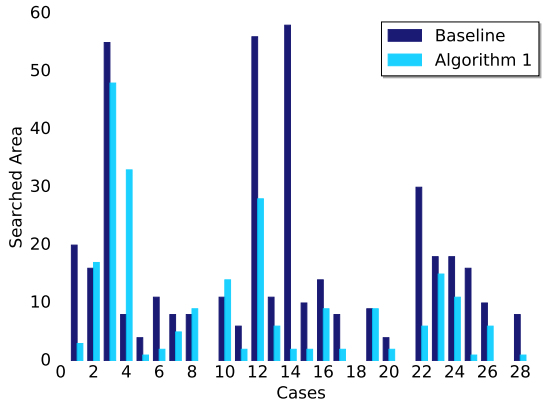
We also examined cases where the size of the grid squares was 2×2 miles. In the 24 cases, the area reduction achieved was in 19 cases using our method, and by an average equals to 11.21 square miles. Further, in the 4 cases, our method caused an increase in the search area, however, the increase was 7.5 square miles on average. Our method outperformed the baseline in area reduction with an average and total of 7.62 and 183 mile square, respectively ($t(24) = 1.88, p < 0.03$). The average probabilities of locating missing person while searching the same size of areas randomly are 0.23 and 0.42 of our approach and baseline, respectively (see Fig. 3.10b).

3.6.2 Consideration of Dog Team Detections

The experiments in the previous section illustrated how our approach could reduce the search area over the baseline for standard grid settings. However, in the events that a dog team detects evidence of the missing person, it may lead to a continued search outside of the assigned grid square. These searches can lead to FMG personnel examining up to a mile outside a designated location. In this section, we consider a grid square settings in the last section, but also allow for an additional mile outside the square to mimic the effect of

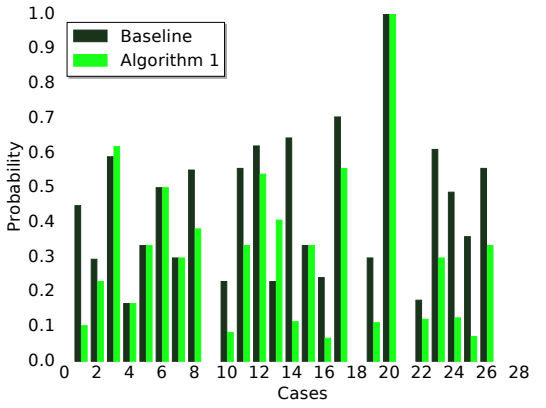


(a) Search Area with 1×1 Mile per Observation

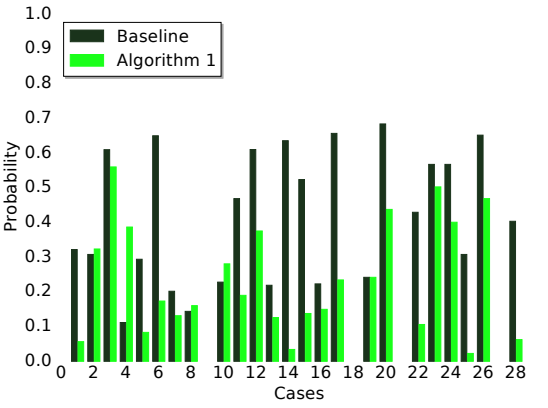


(b) Search Area with 2×2 Miles per Observation

Fig. 3.9: Searched Area until the Missing Person Is Located (Baseline and Algorithm 1).

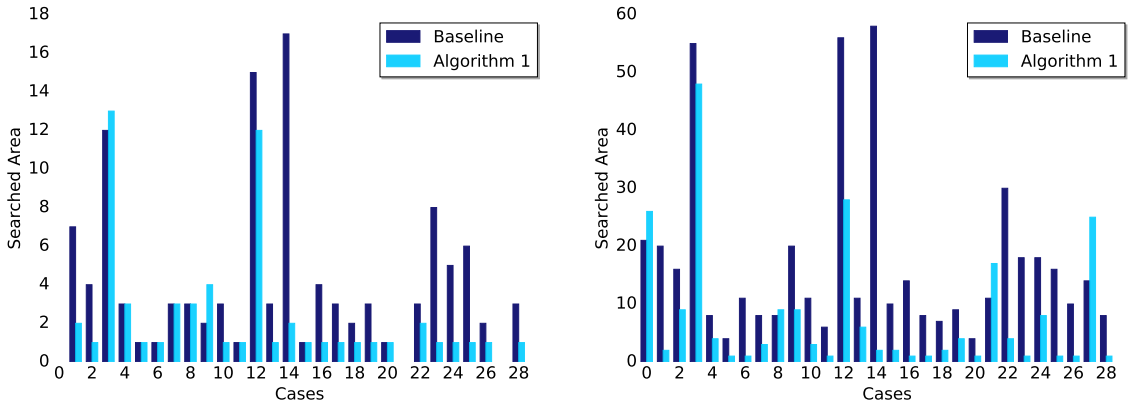


(a) Search Area with 1×1 Mile per Observation



(b) Search Area with 2×2 Miles per Observation

Fig. 3.10: Probability of Locating Missing Person for the Searched Area Demonstrated in Fig. 3.9.



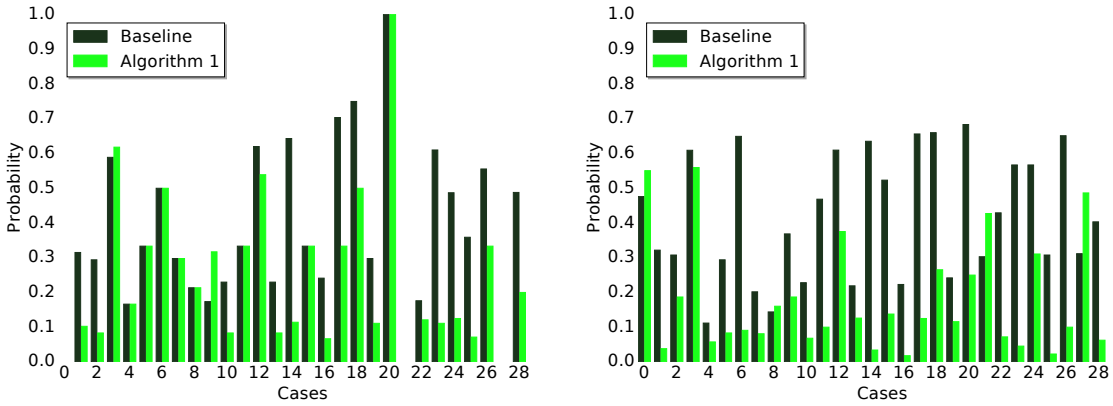
(a) Search Area with 1×1 Mile per Observation

(b) Search Area with 2×2 Miles per Observation

Fig. 3.11: Searched Area with Dogs Allowed to Explore 1 Mile Beyond the Grid (Baseline and Algorithm 1).

the dog search team following such a lead. Fig. 3.11 demonstrates the reduction of area based on our approach (double distance integer program with Algorithm 1 and $\epsilon = 0.05$) when compared to the baseline. We investigate the area reduction with grid squares of 1×1 miles and 2×2 miles. According to Fig. 3.11a, in the 26 cases where the missing person was located, our approach achieved area reduction in 16 cases - reducing the search area by 3.625 square miles on average. In the 2 cases where our method caused the search area to increase, the increase was only 1.5 square miles on average. This contrasts with the cases where the area was reduced - reducing the search area by up to 15 square miles. Our method outperformed the baseline in area reduction with an average and total of of 2.12 and 55 mile square ($t(26) = 2.06, p < 0.02$). We also calculated the probability of locating missing person by searching the same size of areas randomly in Fig. 3.12. The average probability for our approach and baseline is 0.18 and 0.42, respectively (see Fig. 3.12a).

We examined cases where the size of the grid squares was 2×2 miles. In the 29 cases,



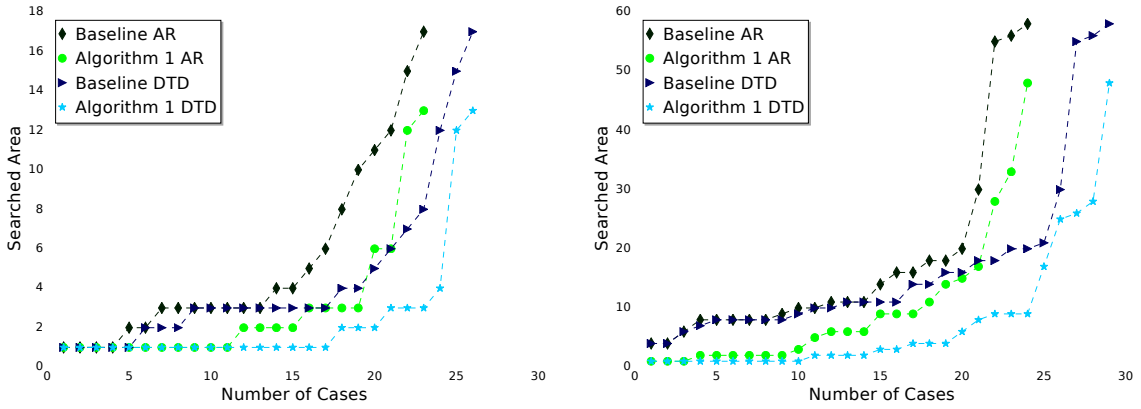
(a) Search Area with 1×1 Mile per Observation

(b) Search Area with 2×2 Miles per Observation

Fig. 3.12: Probability of Locating Missing Person for the Searched Area Demonstrated in Fig. 3.11. Dogs Are Allowed to Explore 1 Mile Beyond the Grid (Algorithm 1).

the area reduction achieved by our method was in 25 cases, and on average by 11.68 square miles. In the 4 cases where our method caused the search area to increase, the increase was 5.75 square miles on average. This contrasts with the cases with the reduced search area by up to 56 square miles. Our method outperformed the baseline in area reduction with an average and total of 9.28 and 269 mile square ($t(29) = 2.7, p < 0.005$). The average probabilities of locating missing person while searching the same size of areas randomly are 0.18 and 0.42 for our approach and baseline respectively as it is shown in Fig. 3.12b.

We also compared the results in Fig. 3.9 and 3.11. In both search area sizes (1×1 and 2×2 miles), Algorithm 1 searches the smaller area and is ahead of the baseline as it is shown in Fig. 3.13.



(a) Search Area with 1×1 Mile per Observation

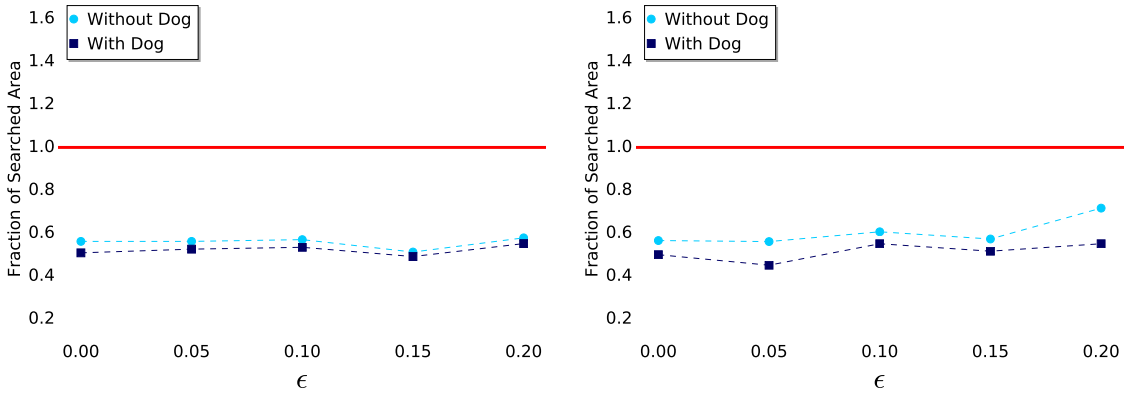
(b) Search Area with 2×2 Miles per Observation

Fig. 3.13: Comparison of Searched Area over All Cases in Both Area Reduction (AR) and Consideration of Dog Team Detection (DTD) Using Algorithm 1 and the Baseline.

3.6.3 Parameter Sensitivity

We compare different values of ϵ in both double distance integer programs (iterative search resource allocation and non-iterative program). The impact of changing the parameter ϵ is shown in Fig. 3.14. We plot the fraction of area searched by our method over the baseline, against the ϵ, for both grid sizes of 1×1 and 2×2. We note that while the extreme values of ϵ (i.e. 0.0 and 0.20) negatively effected the performance of both approaches, we achieved relatively stable results for intermediate values - noting that the best performance was for ϵ equal to 0.05 - which we used in the experiments.

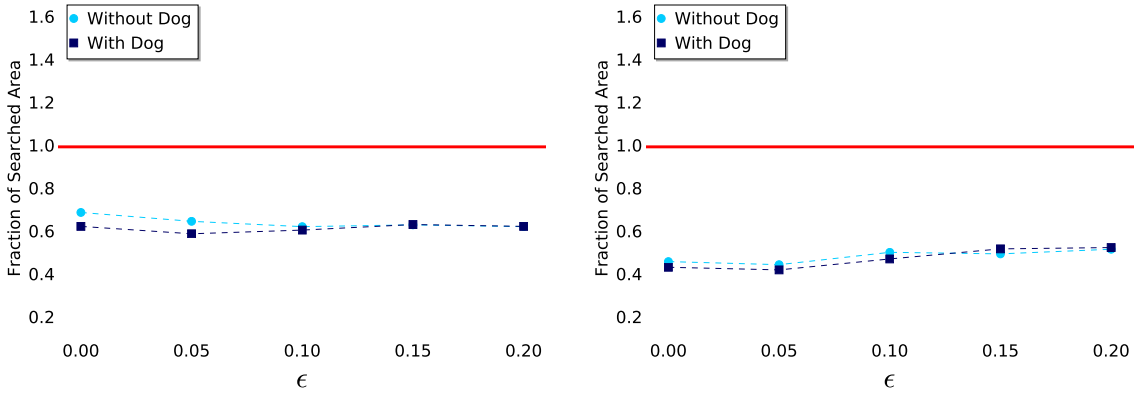
We also studied the performance of our optimization approach without algorithm 1 (i.e. prioritize locations by equation 5 after selecting the values for β_o through optimization of 19 with regards to Lines 9-11). The results are depicted in Fig. 3.15. The behavior of the algorithm for different settings of ϵ were similar to that found with Algorithm 1, the reduction in search area was generally less.



(a) Search Area with 1×1 Mile per Observation (**Algorithm 1**)

(b) Search Area with 2×2 Miles per Observation (**Algorithm 1**)

Fig. 3.14: Fraction of Total Area Searched Across All Cases with the Iterative Search Resource Allocation Approach over the Baseline.



(a) Search Area with 1×1 Mile per Observation ***Not*** **Using Algorithm 1**

(b) Search Area with 2×2 Miles per Observation ***Not*** **Using Algorithm 1**

Fig. 3.15: Fraction of Total Area Searched Across All Cases by the Double Distance Integer Programming Approach (***Not*** **Using Algorithm 1**) over the Baseline.

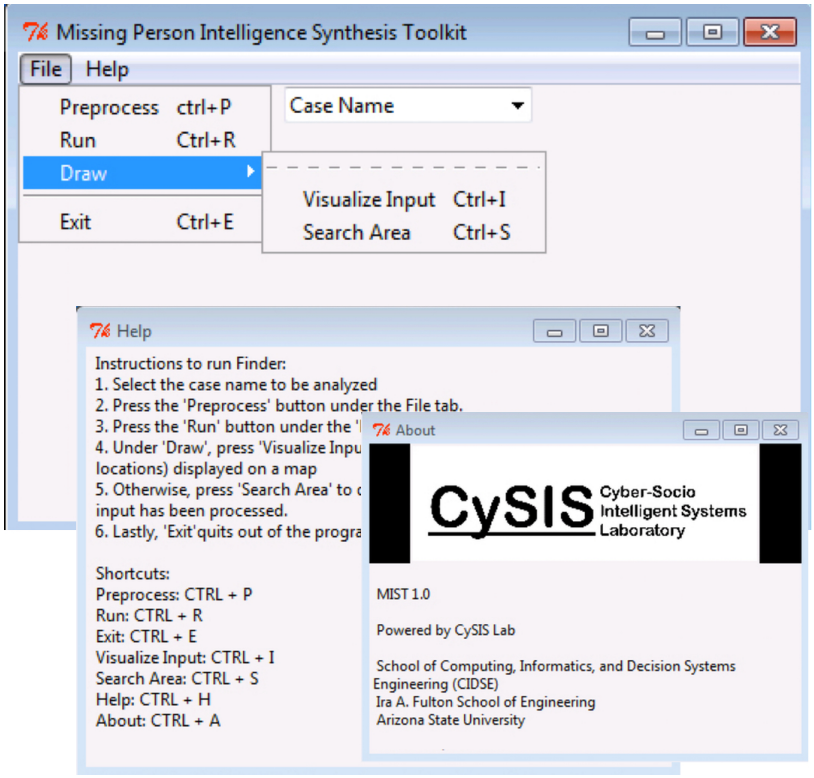


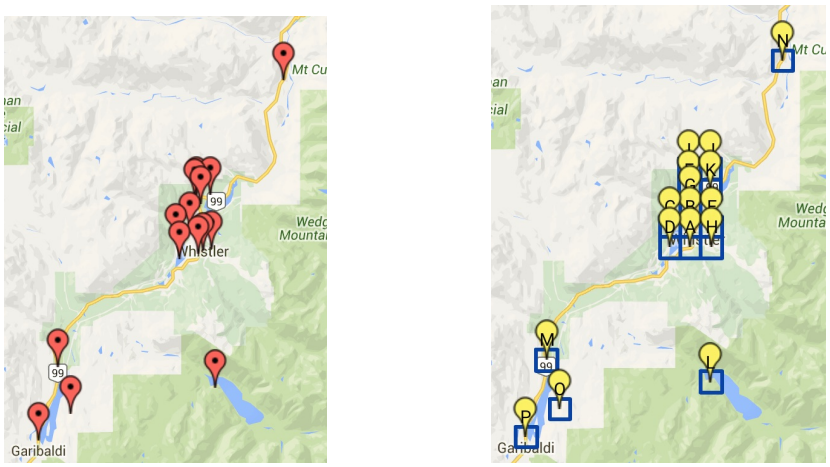
Fig. 3.16: The MIST User Interface.

3.6.4 User Interface

We have created a user interface using the TKinter library and provided Google map and Google earth visualizations in HTML and KML formats. As displayed in Fig. 3.16, users can easily use the interface to run new missing person cases using case information and reporters historical data. Fig. 3.17 shows an example of the input and outout visualization by MIST. At the time of this writing, we have provided results of MIST to support an active case with FMG. FMG found the result consistent with their experiences.

3.7 Related Work

Recently, there has been some work [68, 69, 71, 73, 39] addressing geospatial abductive inference which was first introduced in [74]. In [68] for example, authors studied the case



(a) Observations (b) Ordered Search Areas

Fig. 3.17: An Example of Input (a) and Output (b) by MIST.

of geospatial abduction where there is an explicit adversary who is interested in ensuring that the agent does not detect the partner locations in an attempt to simulating the real-world scenario of insurgents who conduct IED (improvised explosive device) attacks. Another work [69], has adopted geospatial abduction to develop a software tool which applies geospatial abduction to the environment of Afghanistan, to look for insurgent high-value targets, supporting insurgent operations. The work of [71] introduced a variant of the GAPs called region-based GAPs (RGAPs) which deals with the multiple possible definitions of the subregions of the map. Finally, spatial cultural abductive reasoning engine which solves spatial abductive problems was developed in [73]. Aside from introducing GAP, the work of [74] demonstrated the accuracy of proposed framework on a real-world dataset of insurgent IED attacks against US forces in Iraq. Further, the work of [39] proposed a technique to reduce the computational cost of point-based GAPs. They presented an exact algorithm for the natural optimization problem of point-based GAPs. Geospatial abduction problems are related to facility location [76] and sensor placement problems [40] in that they identify a set of geo-locations to optimize a cost or reward function. However, there are

key differences amongst these various frameworks that arise from the difference between explanation and optimization. Interested readers can refer to [72] for further discussion on this topic.

Similarly, [2] presents a specific aspect of the well-known qualification problem, namely spatial qualitative reasoning approach, which aims at investigating the possibility of an agent being present at a specific location at a certain time to carry out an action or participate in an event, given its known antecedents. This work is different from both above papers and our study, as it takes on purely logical approach to formalizing spatial qualifications, while our work and other aforementioned studies use geometric and probabilistic techniques. Further, the framework of this chapter is tailored specifically for the missing person problem.

Looking beyond geospatial abduction, recent research has demonstrated that GPS (positional) data could be used to learn rich models of human activity [58, 59, 60, 27]. For example, [58, 59, 60], modeled the human interactions and intentions in a fully relational multi-agent setting. They used raw GPS data from a real-world game of capture the flag and Markov logic- a statistical-relational language. Whereas [27] developed a model to simulate the behaviors associated with insurgent attacks, and their relationship with geographic locations and temporal windows.

At first glance, one may think our work is similar to [40], in that they identify a set of geo-locations to optimize a cost or reward function. However, as described, there are key differences amongst these various frameworks that arise from the difference between explanation and optimization.

Finally, compared to the conference version [65], the experiments are entirely revised and new experiments are also conducted. A new heuristic used to enhance our algorithmic approach. The number of cases in the experimental results has been increased. More details on the dataset is also provided.

3.8 Conclusion

In this chapter, we studied induction and abduction inferences and introduced the Missing Person Intelligence Synthesis Toolkit (MIST) which leverages a data-driven variant of geospatial abductive inference. MIST can rank-order the set of search locations provided by a group of experts. The experimental results showed that our approach is able to reduce the total search area by a total of 53 square miles for standard searched and by 55 square miles when dog team assets obtain a detection. This reduction will make FMG locating missing persons faster while saving in direct and indirect cost. At the time of writing this manuscript, FMG has started to use MIST to support 2 missing-persons cases.

This work can be extended in several directions including utilizing a probabilistic variant of the feasibility function, incorporating other features such as missing person's corresponding region, age, gender into the model. In next chapter, we aim to use these reasoning approaches to identify pathogenic social media accounts.

Chapter 4

UNSUPERVISED FRAMEWORK TO DETECT PATHOGENIC SOCIAL MEDIA ACCOUNTS

4.1 Introduction

The spread of harmful mis-information in social media is a pressing problem. We refer accounts that have the capability of spreading such information to viral proportions as "Pathogenic Social Media" (PSM) accounts. These accounts include terrorist supporters accounts, water armies, and fake news writers. These organized groups/accounts spread messages regarding certain topics. They might be multiple people that tweet/retweet through multiple accounts to promote/degrade an idea. This can influence public opinion. Identifying PSM accounts has important applications to countering extremism [36, 4], the detection of water armies [18, 17, 81] and fake news campaigns [31, 35, 4]. In Twitter, many of these accounts are social bots.

The PSM accounts that propagate information are key to a malicious information campaign and detecting them is critical to understanding and stopping such campaigns. However, this is difficult in practice. Existing methods rely on message content [45], network structure [13] or a combination of both [77, 21, 22]. However, reliance on information of this type leads to two challenges. First, network structure is not always available. For example, the Facebook API does not make this information available without permission of the users (which is likely a non-starter for PSM accounts). Second, the use of content often necessitates the training of a new model for a previously unobserved topics. For example, PSM accounts taking part in elections in the U.S. and Europe will likely leverage different types of content. In this chapter, we propose a method based on causal analysis to

avoid these very problems. The main requirement is an activity log of user's activities and timestamp. Further, as our method does not rely on data used in previous approaches, it is inherently complementary – which allows for future combined methods.

In this chapter, we aim to find PSM users in the *viral cascades*. As viral cascades are so rare, the users that cause them are suspicious accounts. To address these issues, we leverage causal analysis [78, 38]. We developed, implemented, and evaluated a framework for identifying PSM accounts. This chapter makes the following contributions:

- We proposed a PSM detection framework that does not leverage network structure, cascade path information, content and user's information.
- We observed that PSM accounts have higher causality values.
- We introduced a series of causality-based metrics for identifying PSM users - which alone can achieve precision of 0.66.
- We introduced an unsupervised label propagation framework that, when combined with our causal metrics, provide a precision of 0.75. We showed that our framework significantly outperforms random method (0.11), the content-based bot detection (0.13), all features (0.16), and Sentimetrix [77] (0.11).
- Our framework is able to find the more important PSM accounts in comparison with the baseline methods. The larger the cascade is, the more important its PSM accounts are and our model can capture those cascades better.

The rest of the chapter is organized as follows. In Section 4.2, we describe our framework that leverages causal analysis and label propagation. Then we present the algorithms in Section 4.3. This is followed by a description of our dataset in Section 4.4. In Section 4.5, the causality analysis is discussed. Then we describe our implementation and discuss our results in Section 4.6. Finally, related work is reviewed in Section 4.7.

4.2 Technical Approach

4.2.1 Technical Preliminaries

Throughout this chapter we shall represent cascades as an "action log" ($Actions$) of tuples where each tuple $(u, m, t) \in Actions$ corresponds with a user $u \in U$ posting message $m \in M$ at time $t \in T$, following the convention of [29]. We assume that set M includes posts/repost of a certain original tweet or message. For a given message, we only consider the first occurrence of each user. We define $Actions_m$ as a subset of $Actions$ for a specific message m. Formally, we define it as $Actions_m = \{(u', m', t') \in Actions \; s.t. \; m' = m\}$.

Definition 4.2.1. (m-participant). *For a given $m \in M$, user u is an m-**participant** if there exists t such that $(u, m, t) \in Actions$.*

Note that the users posting tweet/retweet in the early stage of cascades are the most important ones since they play a significant role in advertising the message and making it viral. For a given $m \in M$, we say m-participant i "precedes" m-participant j if there exists $t < t'$ where $(i, m, t), (j, m, t') \in Actions$. Thus, we define *key users* as a set of users adopting a message in the early stage of its life span. We formally define *key user* as follows:

Definition 4.2.2. (Key User). *For a given message m, m-participant i, and $Actions_m$, we say user i is a **key user** iff user i precedes at least ϕ fraction of m-participants (formally: $|Actions_m| \times \phi \leq |\{j | \exists t' : (j, m, t') \in Actions_m \wedge t' > t\}|$, $(i, m, t) \in Actions_m$), where $\phi \in (0, 1)$.*

The notation $|\cdot|$ denotes the cardinality of a set. All messages are not equally important. That is, only a small portion of them gets popular. We define *viral messages* as follows:

Definition 4.2.3. (Viral Messages). *For a given threshold θ, we say that a message $m \in M$ is **viral** iff $|Actions_m| \geq \theta$. We use M_{vir} to denote the set of viral messages.*

The Definition 4.2.3 allows us to compute the prior probability of a message (cascade) going viral as follows:

$$\rho = \frac{|M_{vir}|}{|M|} \tag{4.1}$$

We also define the probability of a cascade m going viral given some user i was involved as:

$$p_{m|i} = \frac{|\{m \in M_{vir}\, s.t.\, i\, is\, a\, key\, user\}|}{|\{m \in M\, s.t.\, i\, is\, a\, key\, user\}|} \tag{4.2}$$

We are also concerned with two other measures. First, the probability that two users i and j tweet or retweet viral post m chronologically, and both are key users. In other words, these two users are making post m viral.

$$p_{i,j} = \frac{\begin{array}{c}|\{m \in M_{vir}|\exists t, t'\, where\, t < t'\, and \\ (i,m,t),(j,m,t') \in Actions\}|\end{array}}{|m \in M|\exists t, t'\, where\, (i,m,t),(j,m,t') \in Actions|} \tag{4.3}$$

Second, the probability that key user j tweets/retweets viral post m and user i does not tweet/retweet earlier than j. In other words, only user j is making post m viral.

$$p_{\neg i,j} = \frac{\begin{array}{c}|\{m \in M_{vir}|\exists t'\, s.t.\, (j,m,t') \in Actions\, and \\ \nexists t\, where\, t < t',\ (i,m,t) \in Actions\}|\end{array}}{\begin{array}{c}|\{m \in M|\exists t'\, s.t.\, (j,m,t') \in Actions\, and \\ \nexists t\, where\, t < t',\ (i,m,t) \in Actions\}|\end{array}} \tag{4.4}$$

Knowing the action log, we aim to find a set of pathogenic social media (PSM) accounts. These users are associated with the early stages of large information cascades and, once detected, are often deactivated by a social media firm. In the causal framework, we introduce a series of causality-based metrics for identifying PSM users.

4.2.2 Causal Framework

We adopt the causal inference framework previously introduced in [78, 38]. We expand upon that work in two ways: (1.) we adopt it to the problem of identifying PSM accounts and (2.) we extend their single causal metric to a set of metrics. Multiple causality measurements

provide a stronger determination of significant causality relationships. For a given viral cascade, we seek to identify potential users who likely *cause* the cascade viral. We first require an initial set of criteria for such a causal user. We do this by instantiating the notion of Prima Facie causes to our particular use case below:

Definition 4.2.4. (Prima Facie Causal User). *A user u is a prima facie causal user of cascade m iff: User u is a key user of m, $m \in M_{vir}$, and $p_{m|u} > \rho$.*

For a given cascade m, we will often use the language *prima facie causal user* to describe user i is a prima facie cause for m to be viral. In determining if a given prima facie causal user is causal, we must consider other "related" users. In this chapter, we say i and j are m-related if (1.) i and j are both prima facie causal users for m, (2.) i and j are both key users for m, and (3.) i precedes j. Hence, we will define the set of "related users" for user i (denoted $R(i)$) as follows:

$$R(i) = \{j \; s.t. \; j \neq i \; , \exists m \in M \; s.t. \; i, j \; are \; m - related\} \tag{4.5}$$

Therefore, $p_{i,j}$ in (4.3) is the probability that cascade m goes viral given both users i and j, and $p_{\neg i,j}$ in (4.4) is the probability that cascade m goes viral given key user j tweets/retweets it while key user i does not tweet/retweet m or precedes j. The idea is that if $p_{i,j} - p_{\neg i,j} > 0$, then user i is more likely a cause than j for m to become viral. We measure *Kleinberg-Mishra causality* ($\epsilon_{K\&M}$) as the average of this quantity to determine how causal a given user i is as follows:

$$\epsilon_{K\&M}(i) = \frac{\sum_{j \in R(i)} (p_{i,j} - p_{\neg i,j})}{|R(i)|} \tag{4.6}$$

Intuitively, $\epsilon_{K\&M}$ measures the degree of causality exhibited by user i. Additionally, we find it useful to include a few other measures. We introduce *relative likelihood causality* (ϵ_{rel}) as follows:

$$\epsilon_{rel}(i) = \frac{\sum_{j \in R(i)} S(i,j)}{|R(i)|} \tag{4.7}$$

$$S(i,j) = \begin{cases} (\frac{p_{i,j}}{p_{\neg i,j}+\alpha}) - 1, & p_{i,j} > p_{\neg i,j} \\ 0, & p_{i,j} = p_{\neg i,j'} \\ 1 - (\frac{p_{\neg i,j}}{p_{i,j}}), & \text{otherwise} \end{cases} \tag{4.8}$$

where α is infinitesimal. Relative likelihood causality metric assesses the relative difference between $p_{i,j}$ and $p_{\neg i,j}$. This helps us to find new users that may not be prioritized by $\epsilon_{K\&M}$. We also find that if a user is mostly appearing after those with the high value of $\epsilon_{K\&M}$, then it is likely to be a PSM account. One can consider all possible combinations of events to capture this situation. However, this approach is computationally expensive. Therefore, we define $\mathcal{Q}(j)$ as follows:

$$\mathcal{Q}(j) = \{i \; s.t. \;\; j \in R(i)\} \tag{4.9}$$

Consider the following example:

Example 4.2.1. *Consider two cascades (actions)* $\tau_1 = \{\text{A, B, C, D, E, F, G, H}\}$ *and* $\tau_2 = \{\text{N, M, C, A, H, V, S, T}\}$ *where the capital letters signify users. We aim to relate key users while* $\phi = 0.5$ *(Definition 4.2.2). Table 4.1 shows the related users* $R(.)$ *for each cascade. Note that the final set* $R(.)$ *for each user, is the union of all sets from the cascades. Set* $Q(.)$ *for the users of Table 4.1 are presented in Table 4.2.*

Accordingly, we define *neighborhood-based causality* (ϵ_{nb}) as the average $\epsilon_{K\&M}(i)$ for all $i \in Q(j)$ as follows:

$$\epsilon_{nb}(j) = \frac{\sum_{i \in \mathcal{Q}(j)} \epsilon_{K\&M}(i)}{|\mathcal{Q}(j)|} \tag{4.10}$$

The intuition behind this metric is that accounts who are retweeting a message that was tweeted/retweeted by several causal users are potential for PSM accounts. We also define the *weighted neighborhood-based causality* (ϵ_{wnb}) as follows:

$$\epsilon_{wnb}(j) = \frac{\sum_{i \in \mathcal{Q}(j)} w_i \times \epsilon_{K\&M}(i)}{\sum_{i \in \mathcal{Q}(j)} w_i} \tag{4.11}$$

Table 4.1: Related Users $R(.)$ (4.5) of Cascades $\tau_1 = \{\text{A, B, C, D, E, F, G, H}\}$ and $\tau_2 = \{\text{N, M, C, A, H, V, S, T}\}$

User	R_{τ_1}	R_{τ_2}	R
A	{B, C, D, E, F}	{H, V}	{B, C, D, E, F, H, V}
B	{C, D, E, F}	{}	{C, D, E, F}
C	{D, E, F}	{A, H, V}	{A,D, E, F, H, V}
D	{E, F}	{}	{E, F}
E	{F}	{}	{F}
N	{}	{M, C, A, H, V}	{A, C, H, M, V}
M	{}	{C, A, H, V}	{A, C, H, V}
H	{}	{V}	{V}

The intuition behind the metric ϵ_{wnb} is that the users in $\mathcal{Q}$ may not have the same impact on user j and thus different weights w_i are assigned to each user i with $\epsilon_{K\&M}(i)$.

4.2.3 Problem Statements

Our goal is to find the potential PSM accounts from the cascades. Assigning a score to each user and applying threshold-based algorithm is one way of selecting users. In the previous section, we defined causality metrics where each of them or combination of them can be a strategy for assigning scores. Users with high values for causality metrics are more likely to be PSM accounts - later we demonstrate the relationship between these measurements and the real world by identifying accounts deactivated eventually.

Problem 1. (Threshold-based Problem). *Given a causality metric ϵ_k where $k \in \{K\&M, rel, nb, wnb\}$, parameter θ, set of users U, we wish to identify set $\{u \; s.t. \; \forall u \in U, \; \epsilon_k(u) \geq \theta\}$.*

We find that considering a set of cascades as a hypergraph where users of each cascade

Table 4.2: Set $\mathcal{Q}(.)$ of Users Table 4.1 in (4.9)

User	Total
A	{C, N, M}
B	{A}
C	{A, B, N, M}
D	{A, B, C}
E	{A, B, C, D}
N	{}
M	{N}
H	{A, C, N, M}

are connected to each other can better model the PSM accounts. The intuition is that densely connected users with high values for causality are the most potential PSM accounts. In other words, we are interested in selecting a user if (1.) it has a score higher than a specific threshold or (2.) it has a lower score but occurs in the cascades where high score users occur. Therefore, we define the *label propagation* problem as follows:

Problem 2. (Label Propagation Problem). *Given a causality metric* ϵ_k *where* $k \in \{K\&M, rel, nb, wnb\}$*, parameters* θ*,* λ*, set of cascades* $\mathcal{T} = \{\tau_1, \tau_1, ..., \tau_n\}$*, and set of users* U*, we wish to identify set* $\mathcal{S} : \mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_l, ..., \mathcal{S}_{|U|}$ *where* $\mathcal{S}_l = \{u | \forall \tau \in \mathcal{T}, \forall u \in (\tau \backslash \mathcal{S}_{l-1}), \epsilon_k(u) \geq (H_\tau^l - \lambda)\}$ *and* $H_\tau^l = \{min(\epsilon_k(u)) \; s.t. \; \forall u \in \tau \wedge u \in \bigcup_{l' \in [1,l)} \mathcal{S}_{l'}\}$.

4.3 Algorithms

4.3.1 Algorithm for Threshold-based Problems

To calculate causality metrics, we use map-reduce programming model. In this approach, we select users with causality value greater than or equal to a specific threshold. We refer to

this approach as the *Threshold-based Selection Approach*.

4.3.2 Label Propagation Algorithms

Label propagation algorithms [85, 6, 56] iteratively propagate labels of a seed set to their neighbors. All nodes or a subset of nodes in the graph are usually used as a seed set. We propose a Label Propagation Algorithm (Algorithm 3) to solve problem 2. We first take users with causality value greater than or equal to a specific threshold (i.e. 0.9) as the seed set. Then, in each iteration, every selected user u can activate user u' if the following two conditions are satisfied: (1.) u and u' have at least one cascade (action) in common and (2.) $\epsilon_k(u') \geq \epsilon_k(u) - \lambda, \lambda \in (0, 1)$. Note that, we set a minimum threshold such as 0.7 so that all users are supposed to satisfy it. In this algorithm, inputs are a set of cascades (actions) $\mathcal{T}$, causality metric ϵ_k and two parameters θ, λ in $(0, 1)$. This algorithm is illustrated by a toy example:

Example 4.3.1. *Consider three cascades* $\{\{$A, B, G$\}, \{$A, B, C, D, E, G, H, I$\}, \{$E, H, I$\}\}$ *as shown in hypergraph Fig. 4.1. Let us consider the minimum acceptable value as 0.7; in this case, users* C *and* E *would not be activated in this algorithm. Assuming two parameters* $\theta = 0.9,\ \lambda = 0.1$*, both users* A *and* G *get activated (Fig. 4.1a). Note that an active user is able to activate inactive ones if (1.) it is connected to the inactive user in the hypergraph, (2.) score of the inactive user meets the threshold. In the next step, only user* B *will be influenced by* G *(*$0.82 \geq 0.92 - 0.1$*) as it is shown in Fig. 4.1b. Then, user* D *will be influenced by user* B *(*$0.73 \geq 0.82 - 0.1$*). In the next step (Fig. 4.1d), the algorithm terminate since no new user is adopted. As it is shown, user* I *and* H *are not influenced although they have larger values of* ϵ *in comparison with user* D.

Proposition 1. *Given a set of cascades* $\mathcal{T}$*, a threshold* θ*, parameter* λ*, and causality values* ϵ_k *where* $k \in \{K\&M, rel, nb, wnb\}$*, ProSel returns a set of users* $\mathcal{R} = \{u | \epsilon_k(u) \geq \theta$ *or* $\exists u'\ s.t.\ u', u \in \tau, \epsilon_k(u) \geq \epsilon_k(u') - \lambda\ and\ u'\ is\ picked\}$*. Set* $\mathcal{R}$ *is equivalent to the set*

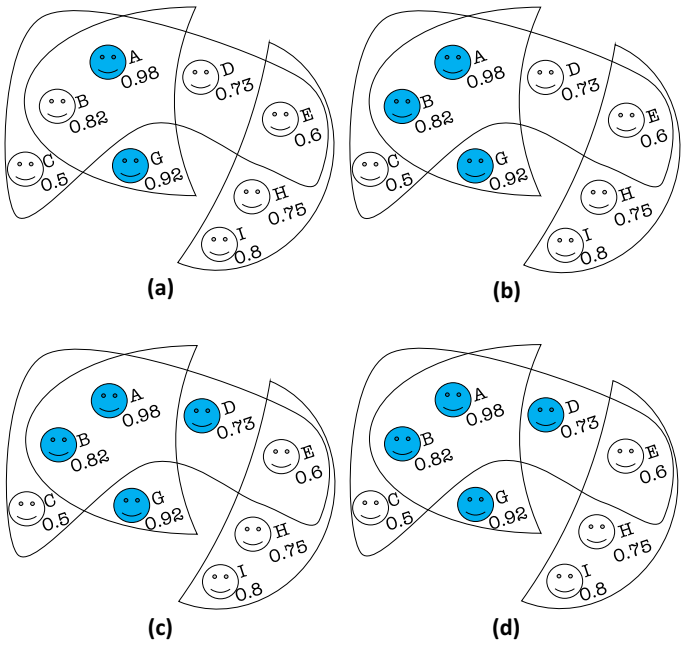


Fig. 4.1: A Toy Example of Algorithm ProSel. Blue Faces Depict Active Users.

$\mathcal{S}$ *in Problem 2.*

4.4 ISIS Dataset

Our dataset consists of ISIS related tweets/retweets in Arabic gathered from Feb. 2016 to May 2016. The dataset includes tweets and the associated information such as user ID, re-tweet ID, hashtags, content, date and time. About 53M tweets are collected based on the 290 hashtags such as Terrorism, State of the Islamic-Caliphate, Rebels, Burqa State, and Bashar-Assad, Ahrar Al-Sham, and Syrian Army. In this chapter, we only use tweets (more than 9M) associated with viral cascades. The statistics of the dataset are presented in Table 4.3 discussed in details below.

Cascades. In this chapter, we aim to identify PSM accounts - which in this dataset are mainly social bots or terrorism-supporting accounts that participate in viral cascades. The tweets that have been retweeted from 102 to 18,892 times. This leads to more than 35k

Algorithm 3 Label Propagation Algorithm (*ProSel*)

procedure PROSEL($\mathcal{T}, \epsilon_k, \theta, \lambda$)

$\mathcal{S} = \{(u, \epsilon_k(u)) | \forall u \in U, \epsilon_k(u) \geq \theta\}$

$\mathcal{R} = \mathcal{S}$

$H = \emptyset$

while $|\mathcal{S}| > 0$ **do**

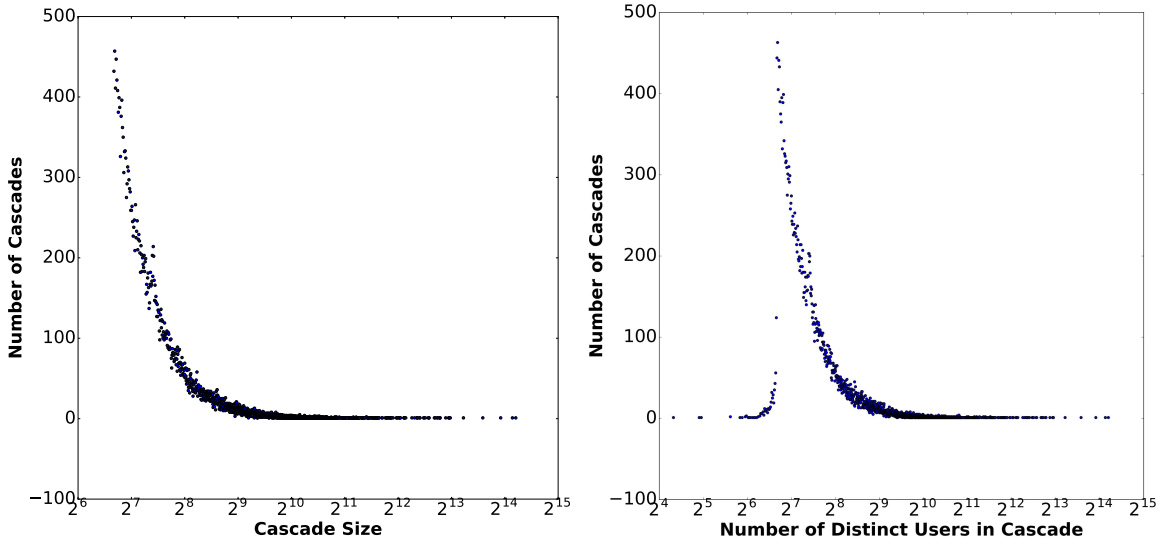
$H' = \{(\tau, \epsilon_m) | \forall (\tau, \epsilon) \in H,\ \epsilon_m = min(\epsilon, min(\{\epsilon' = \mathcal{S}_u\ s.t.\ \forall u \in \tau \wedge u \in \mathcal{S}\}))\}$

$H = H' \cup \{(\tau, \epsilon_m) | \forall \tau \in \mathcal{T} \wedge \tau \notin H',\ \epsilon_m = min(\{\epsilon = \mathcal{S}_u\ s.t.\ \forall u \in \tau \wedge u \in \mathcal{S}\})\}$

$\mathcal{S} = \{(u, \epsilon) | \forall \tau \in \mathcal{T},\ \forall u \in \tau,\ u \notin \mathcal{R},\ \epsilon_k(u) \geq (H_\tau - \lambda)\}$

$\mathcal{R} = \mathcal{R} \cup \mathcal{S}$

return $\mathcal{R}$



(a) All Users (b) Distinct Users

Fig. 4.2: Distribution of Cascades vs Cascade Size

Table 4.3: Statistics of the Dataset

Name	**Values**
Tweets	9,092,978
Cascades	35,251
Users	1,249,293
Generator users	8,056

cascades which are tweeted or retweeted by more than 1M users. The distribution of the number of cascades vs cascade size is illustrated in Fig. 4.2a. There are users that retweet their own tweet or retweet a post several times, we only consider the first tweet/retweet of each user for a given cascade. In other words, duplicate users are removed from the cascades, which make the size of the viral cascades from 20 to 18,789 as shown in Fig. 4.2b. The distribution of the cascades over the cascade life span is illustrated in Fig. 4.3. Cascades took from 16 seconds to more than 94 days to complete.

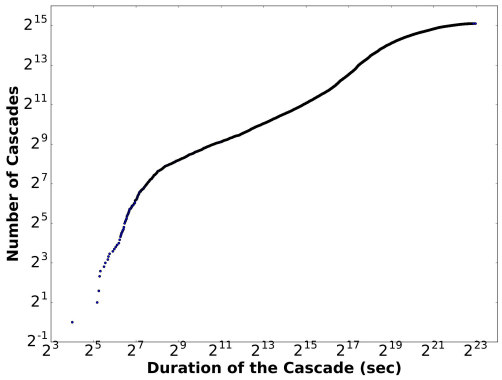


Fig. 4.3: Cumulative Distribution of Duration of Cascades.

Users. There are more than 1M users that have participated in the viral cascades. Fig. 4.4 demonstrates the cumulative distribution of the number of times a user have participated in

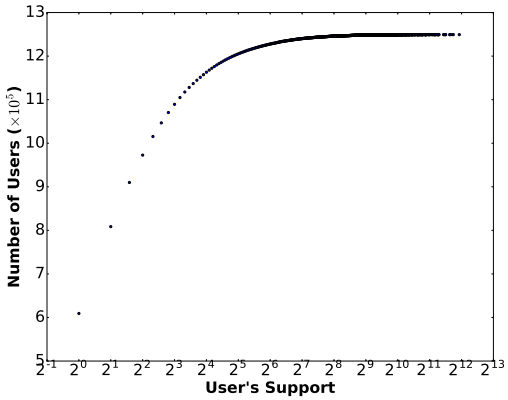


Fig. 4.4: Cumulative Distribution of User's Occurrence in the Dataset.

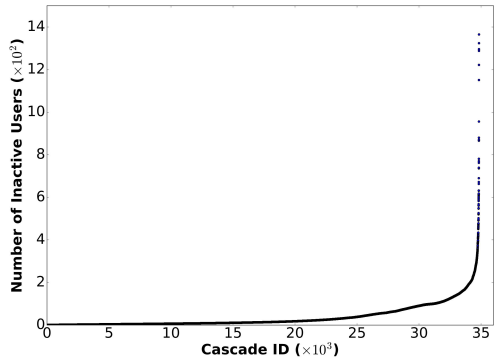


Fig. 4.5: Total Inactive Users in Every Cascade

the cascades. As it is shown, the larger the support value is, the less number of users exists. Moreover, users have tweeted or retweeted posts from 1 to 3,904 times and on average each user has participated more than 7 times.

User's Current Status. We select *key users* that have tweeted or retweeted a post in its early life span - among first half of the users (according to Definition 4.2.2, $\phi = 0.5$), and check whether they are active or not. Accounts are not active if they are suspended or deleted. More than 88% of the users are active as shown in Table 4.4. The statistics of the generator users are also reported. Generator users are those that have initiated a viral cascade. As shown, 90% of the generator users are active as well. Moreover, there are a

Table 4.4: Status of a Subset of the Users in Dataset

Name	**Active**	**Inactive**	**Total**
Users	723,727	93,770	817,497
Generator users	7,243	813	8,056

significant number of cascades with hundreds of inactive users. The number of inactive users in every cascade is illustrated in Fig. 4.5. Inactive users are representative of automatic and terrorism accounts aiming to disseminate their propaganda and manipulate the statistics of the hashtags of their interest.

Generator Users. In this part, we only consider users that have generated (started) the viral tweets. According to Table 4.4, there are more than 7K active and 800 inactive generator users. That is, more than 10% of the generator users are suspended or deleted, which means they are potentially automated accounts. The distribution of the number of tweets generated by generator users shows that most of the users (no matter active and inactive) have generated a few posts (less than or equal to 3) while only a limited number of users are with a large number of tweets.

4.5 Causality Analysis

Here we examine the behavior of the causality metrics. We analyze users considering their current account status in Twitter. We label a user as active (inactive) if the account is still active (suspended or deleted).

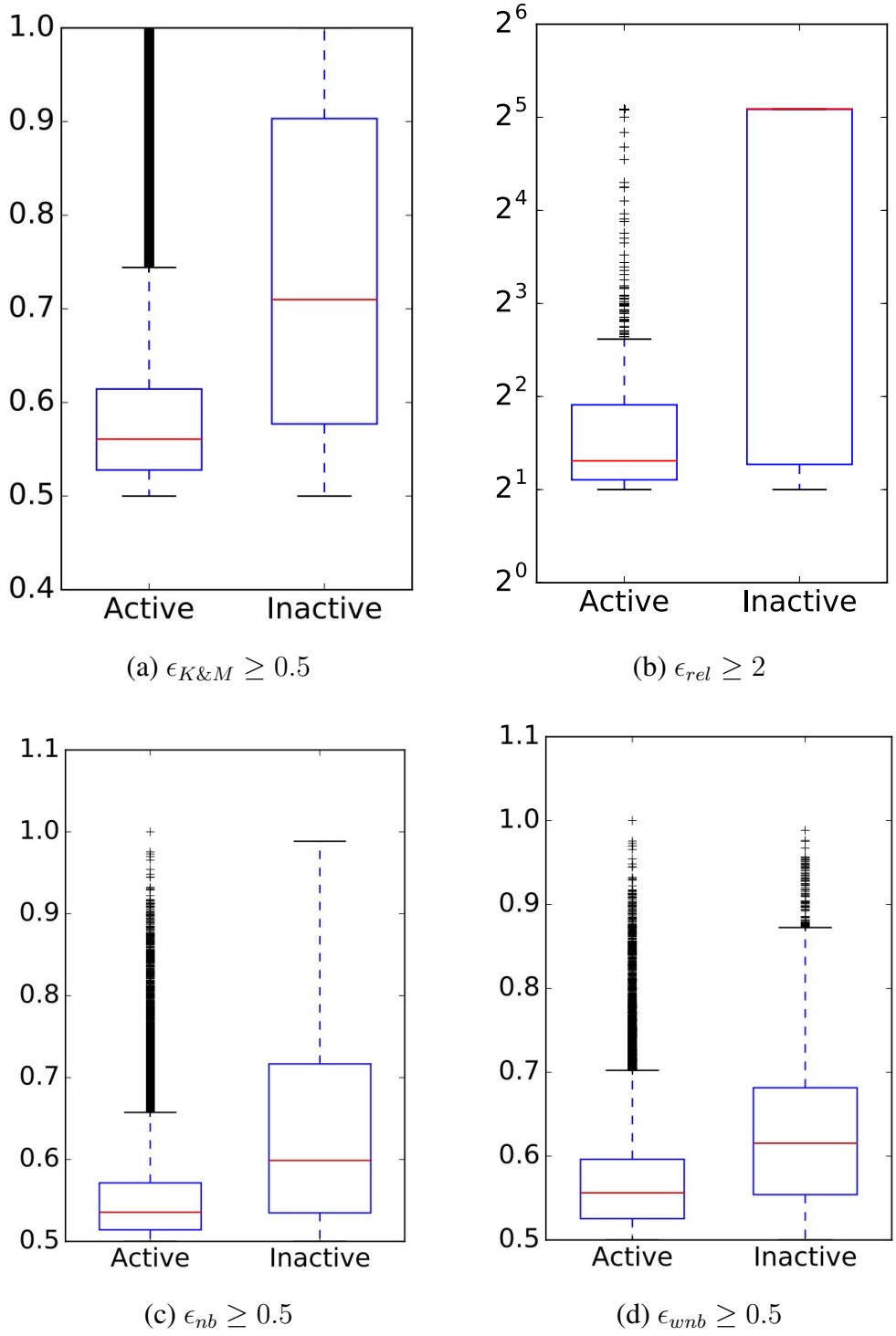
Kleinberg-Mishra Causality. We study the users that get their causality value of $\epsilon_{K\&M}$ greater than or equal to 0.5. As expected, inactive users exhibit different distribution from active users (Fig. 4.6a). We note that significant differences are present - more than 75% of the active users are distributed between 0.5 and 0.62, while more than 50% of the inactive

users are distributed from 0.75 to 1. Also, inactive users have larger values of mean and median than active ones. Note that number of active and inactive users are 404,536 and 52,452. This confirms that this metric is a good indicator to discriminate PSM users from the normal users.

Relative Likelihood Causality. This metric magnifies the interval between every pairs of the probabilities that measures the causality of the users; therefore, the values vary in a wide range. Fig. 4.6b displays the distribution of users having relative likelihood causality of greater than or equal to two. In this metric, 1,274 users get very large values. For the sake of readability, very large values are replaced with 34.0 in Fig. 4.6b. More than 50% of the inactive users get values greater than 32, while the median of active users is 2.48. More than 75% of the active users are distributed in the range of $(2, 4)$. Note that number of active and inactive users in this figure are 3,563 and 1,041, respectively. That is, using this metric and filtering users with the relative likelihood greater than a threshold, leads to the good precision. For example, the threshold in Fig. 4.6b is set to 2 - the precision is more than 0.22 for inactive class. Considering users with a very large value leads to the precision of more than 0.5 and uncovering a significant number of PSMs - 638 inactive users.

Neighborhood-Based Causality. We study the users that get their causality value of ϵ_{nb} greater than or equal to 0.5. As expected, inactive users exhibit different distribution from active users as shown in Fig. 4.6c. Also, inactive users are mostly distributed in the higher range and have larger values of mean and median than active ones. More than 75% of the active users are distributed between 0.5 and 0.6, while more than 50% of the inactive users are distributed from 0.6 to 1. Therefore, increasing the threshold results in the higher precision for the PSM users. Note that the number of active and inactive users are 85,864 and 10,165.

Weighted Neighborhood-Based Causality. This metric is the weighted version of the previous metric (ϵ_{nb}). We assign weight to each user in proportion to her participation rate



(a) $\epsilon_{K\&M} \geq 0.5$

(b) $\epsilon_{rel} \geq 2$

(c) $\epsilon_{nb} \geq 0.5$

(d) $\epsilon_{wnb} \geq 0.5$

Fig. 4.6: Distribution of Various Causality Metrics for Active and Inactive Users.

in the viral cascades. Fig. 4.6d shows the distribution of users with ϵ_{wnb} greater than or equal to 0.5. This metric also displays different distribution for active and inactive users. More than 75% of the active users are distributed between 0.5 and 0.6, while more than 50% of the inactive users are distributed from 0.6 to 1. Note that the number of active and inactive users of ϵ_{wnb} are 52,346 and 16,412. In other words, this metric achieves the largest precision compared to other metrics, 0.24. Clearly, increasing the threshold results in the higher precision for the PSMs.

4.6 Results and Discussion

We implement our code in Scala Spark and Python 2.7x and run it on a machine equipped with an Intel Xeon CPU (1.6 GHz) with 128 GB of RAM running Windows 7. We set the parameter ϕ to label key users 0.5 (Definition 4.2.2). Thus, we are looking for the users that participate in the action before the number of participants gets twice.

In the following sections, first we look at the existing methods. Then we look at two proposed approaches (see Section 4.3): (1) *Threshold-based Selection Approach* - selecting users based on a specific threshold, (2) *Label Propagation Selection Approach* - selecting by applying Algorithm 3. The intuition behind this approach is to select a user if it has a score higher than a threshold or has a lower score but occurs in the cascades that high score users exist. We evaluate methods based on true positive (True Pos), false positive (False Pos), precision, the average (Avg CS) and median (Med CS) of cascade size of the detected PSM accounts. Note that in our problem, precision is the most important metric. The main reason is labeling an account as PSM means it should be deleted. However, removing a real user is costly. Therefore, it is important to have a high precison to prevent removing real user.

4.6.1 Existing Method

Here we use the approach proposed by the top-ranked team in the DARPA Twitter Bot Challenge [77]. We consider all features that we could extract from our dataset. Our features include tweet syntax (average number of hashtags, average number of user mentions, average number of links, average number of special characters), tweet semantics (LDA topics), and user behaviour (tweet spread, tweet frequency, tweet repeats). We apply three existing methods to detect PSM accounts: 1) *Random* selection: This method achieves the precision of 0.11. This also presents that our data is imbalanced and less than 12% of the users are PSM accounts. 2) *Sentimetrix*: We cluster our data by DBSCAN algorithm. We then propagate the labels from 40 initial users to the users in each cluster based on the similarity metric. We use Support Vector Machines (SVM) to classify the remaining PSM accounts [77]. 3) *Classification* methods: In this experiment, we use the same labeled accounts as the previous experiment and apply different machine learning algorithms to predict the label of other samples. We group features based on the limitations of access to data into three categories. First, we consider only using content information (*Content*) to detect the PSM accounts. Second, we use content independent features (*No content*) [77] to classify users. Third, we apply all features (*All features*) to discriminate PSM accounts. The best result for each setting is when we apply Random Forest using all features. According to the results, this method achieves the highest precision of 0.16. Note that, most of the features used in the previous work and our baseline take advantage of both content and network structure. However, there are situations that the network information and content do not exist. In this situation, the best baseline has the precision of 0.15. We study the average (Avg CS) and median (Med CS) of the size of the cascades in which the selected PSM accounts have participated. Table 4.5 also illustrates the false positive, true positive and precision of different methods.

Table 4.5: Existing Methods - Number of Selected Users as PSM

Method	**False Pos**	**True Pos**	**Precision**	**Avg CS**	**Med CS**
$Random$	80,700	10,346	0.11	**289.99**	**184**
$Sentimetrix$	640,552	77,984	0.11	261.37	171
$Content$	292,039	43,483	0.13	267.66	174
$Nocontent$	357,027	63,025	0.15	262.97	172
$Allfeatures$	164,012	31,131	**0.16**	273.21	176

4.6.2 Threshold-based Selection Approach

In this experiment, we select all the users that satisfy the thresholds and check whether they are active or not. A user is *inactive*, if the account is suspended or closed. Since the dataset is not labeled, we label inactive users as PSM accounts. We set the threshold for all metrics to 0.7 except for relative likelihood causality (ϵ_{rel}), which is set to 7. We conduct two types of experiments: first, we study user selection for a given causality metric. We further study this approach using the combinations of metrics.

Single Metric Selection. In this experiment, we attempt to select users based on each individual metric. As expected, these metrics can help us filter a significant amount of active users and overcome the data imbalance issue. Metric $\epsilon_{K\&M}$ achieves the largest recall in comparison with other metrics. However, it has the largest number of false positives. Table 4.6 shows the performance of each metric. The precision value varies from 0.43 to 0.66 and metric ϵ_{wnb} achieves the best value. Metric ϵ_{rel} finds the more important PSM accounts with average cascade size of 567.78 and median of 211. In general, our detected PSM accounts have participated in the larger cascades in comparison with baseline methods. We also observe that these metrics cover different regions of the search area. In other words,

they select different user sets with little overlap between each other. The common users between any two pairs of the features are illustrated in Table 4.7. Considering the union of all metrics, 36,983 and 30,353 active and inactive users are selected, respectively.

Table 4.6: Threshold-based Selection Approach - Number of Selected Users Using Single Metric

Method	**False Pos**	**True Pos**	**Precision**	**Avg CS**	**Med CS**
$All\ features$	164,012	31,131	**0.16**	273.21	176
$No\ content$	357,027	63,025	0.15	262.97	172
$\epsilon_{K\&M}$	36,159	27,192	0.43	383.99	178
ϵ_{rel}	693	641	0.48	**567.78**	**211**
ϵ_{nb}	2,268	2,927	0.56	369.46	183.5
ϵ_{wnb}	7,463	14,409	**0.66**	311.84	164

Table 4.7: Threshold-based Selection Approach - Number of Common Selected Users Using Single Metric

Status	**Active**			**Inactive**		
Method	ϵ_{rel}	ϵ_{nb}	ϵ_{wnb}	ϵ_{rel}	ϵ_{nb}	ϵ_{wnb}
$\epsilon_{K\&M}$	404	1,903	6,992	338	2,340	11,748
ϵ_{rel}		231	175		248	229
ϵ_{wnb}			1,358			1,911

Combination of Metrics Selection. According to Table 4.7, most of the metric pairs have more inactive users in common than active users. In this experiment, we discuss if using the combination of these metrics can help improve the performance. We attempt to select

users that satisfy the threshold for at least three metrics. We get 1,636 inactive users out of 2,887 selected ones, which works better than $\epsilon_{K\&M}$ and ϵ_{rel} while worse than ϵ_{nb} and ϵ_{wnb}. In brief, this approach achieves precision of 0.57. Moreover, the number of false positives (1,251) is lower than most of the other metrics.

4.6.3 Label Propagation Selection Approach

In label propagation selection, we first select a set of users that have a high causality score as seeds, then ProSel selects users that occur with those seeds and have a score higher than a threshold iteratively. Also, the seed set in each iteration is the selected users of the previous iteration. The intuition behind this approach is to select a user if it has a score higher than a threshold or has a lower score but occurs in the cascades that high score users occur. We set the parameters of *ProSel Algorithm* as follows: $\lambda = 0.1$, $\theta = 0.9$, except for relative likelihood causality, where we set $\lambda = 1$, $\theta = 9$. Table 4.8 shows the performance of each metric. Precision of these metrics varies from 0.47 to 0.75 and ϵ_{wnb} achieves the highest precision. Metrics ϵ_{rel} with average cascade size of 612.04 and ϵ_{nb} with median of 230 find the more important PSM accounts. Moreover, detected PSM accounts have participated in the larger cascades compared with threshold-based selection. This approach also produces much lower number of false positives compared to threshold-based selection. The comparison between this approach and threshold-based selection is illustrated in Fig. 4.7. From the precision perspective, label propagation method outperforms the threshold-based one.

The number of common users selected by any pair of two metrics are also illustrated in Table 4.9. It shows that our metrics are powerful to cover different regions of the search area and identify different sets of users. In total, 10,254 distinct active users and 16,096 inactive ones are selected.

Table 4.8: Label Propagation Selection Approach - Number of Selected Users

Method	**False Pos**	**True Pos**	**Precision**	**Avg CS**	**Med CS**
$All\ features$	164,012	31,131	**0.16**	273.21	176
$No\ content$	357,027	63,025	0.15	262.97	172
$\epsilon_{K\&M}$	9,305	14,176	0.60	390.52	179
ϵ_{rel}	561	498	0.47	**612.04**	216
ϵ_{nb}	1,101	1,768	0.62	403.55	**230**
ϵ_{wnb}	1,318	4,000	**0.75**	355.24	183.5

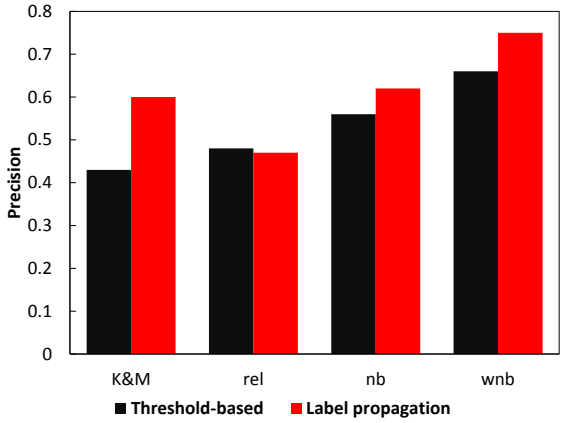


Fig. 4.7: Comparison Between Threshold-based and Label Propagation Selection Approaches for the Inactive Class

4.7 Related Work

To the best of our knowledge, this chapter represents the first unsupervised approach on PSM detection. The majority of previous work was based on three fundamental assumptions. First, *the information of the network is known* [77, 29, 7, 1]. This assumption may not hold in reality. Second, they are language dependent [77, 45]. Third, the majority of botnet detection algorithms focused on bots in general. That is, they did consider the bots

Table 4.9: Label Propagation Selection Approach - Number of Common Selected Users

Status	**Active**			**Inactive**		
Method	ϵ_{rel}	ϵ_{nb}	ϵ_{wnb}	ϵ_{rel}	ϵ_{nb}	ϵ_{wnb}
$\epsilon_{K\&M}$	289	581	1,122	168	1,019	2,788
ϵ_{rel}		15	6		180	102
ϵ_{nb}			151			833

equally [45, 22] where in this work, we identify PSM accounts that spread viral information. Here, we review related work on identifying automatic accounts and terrorist groups. Aside from the bot detection work, our work can be compared with detection of water armies.

Identifying Automatic Accounts. Due to the importance of the issue, DARPA conducted the Twitter bot detection challenge to identify and eliminate influential bots [77]. In this challenge, all teams applied supervised or semi-supervised learning approaches using the diverse sets of features. Most of the previous work extracted different sets of features (tweet syntax and semantics, temporal behavior, user profile, and network features) and conducted supervised or semi-supervised approaches [77, 22, 45]. On the other hand, here, we focus on situations where neither network information nor account related attributes and user profile information are available. Our appproach is also independent of content and language.

Analysis of Terrorist Groups and Detection of Water Armies. Terrorist groups use social media for propaganda dissemination [3]. Benigni et al. [7] conducted vertex clustering and classification to find Islamic Jihad Supporting Community on Twitter. Abdokhodair et al. [1] studied the behaviors and characteristics of Syrian social botnet. Chen et al. [18] found that within the context of news report comments, user-specific measurements can distinguish water army from normal users. Similarly, in [17], Chen et al. applied user behavior and domain-specific features to detect water armies. Our work is different from them since

these methods also applied features related to the accounts and network (follower/followee). However, we do not have any network information and account-related features.

4.8 Conclusion

In this chapter, we studied induction and abduction reasonings on large-scale dataset. we conducted a data-driven study on the pathogenic social media accounts especially terrorist supporters, automatic accounts and bots. We proposed unsupervised causality based framework to detect these groups. Our approach identifies these users without using network structure, cascade path information, content and user's information. We believe our technique can be applied in the areas such as detection of water armies and fake news campaigns.

Chapter 5

SUPERVISED AND SEMI-SUPERVISED FRAMEWORKS TO DETECT PATHOGENIC SOCIAL MEDIA ACCOUNTS

5.1 Introduction

In previous chapter, we proposed causality metrics which are able to detect most of pathogenic social media (PSM) accounts imprecisely (high recall and low precision) or precisely small portion of it (high precision and low recall). In this chapter, our goal is to detect precisely larger portions. In other words, reducing the rate of false positive accounts. We expand on the previous work in [66] and propose graph-based metrics to distinguish PSM accounts from normal users within a short time around their activities. Our new metrics combined with our causal ones can achieve high precision 0.90, while increasing the recall from 0.22 to 0.49. We propose supervised and semi-supervised approaches and then show our proposed methods outperform the ones in the literature. In summary, the major contributions of this chapter are itemized as follows:

- We propose supervised and semi-supervised PSM detection frameworks that do not leverage network structure, cascade path information, content and user's information.
- We introduce graph-based framework using the cascades and propose a series of scalable metrics to identify PSM users. We apply this framework to more than 722K users and 35K cascades.
- We propose a deep neural network framework which achieves AUC of 0.82. We show that our framework significantly outperforms Sentimetrix [77] (0.74), causal-

ity [66] (0.73), time-decay causality [5] (0.66), and causal community detection-based classification [5] (0.6).

- We introduce a self-training semi-supervised framework that can capture more than 29K PSM users with the precision of 0.81. We only used 600 labeled data from training and development sets. Moreover, if a supervisor is involved in the training loop, the proposed algorithm is able to capture more than 80K PSM users.

The rest of the chapter is organized as follows. In Section 5.2, we describe our framework that leverages causal metrics, graph-based metrics. We present the algorithms in Section 5.3. This is followed by a description of our dataset in Section 5.4. Then we describe our implementation and discuss our results in Section 5.5. Finally, related work is reviewed in Section 5.6.

5.2 Technical Approach

5.2.1 Graph-based Framework

User-Message Bipartite Graph. Here, we denote $Actions$ as a bipartite graph $G_{u-m}(U, M, E)$, where users U and messages M are disjoint sets of vertices. There is an annotated link from user u to message m if u has tweeted/retweeted m and is annotated by occurrence time t (see Fig. 5.1). In other words, every edge in graph G_{u-m} is associated with one tuple $(u, m, t) \in Actions$. For a given node $u \in U(m \in M)$, the set $\mathcal{N}_u = \{m' \in M \; s.t. \; (u, m') \in E\}(\mathcal{N}_m = \{u' \in U \; s.t. \; (u', m) \in E\})$ is the set of immediate neighbors of u (m). We also define $U^v \subset U$ which is the set of verified users (often celebrities). We indicate $U^v_m = \{u|(u, m, t) \in Actions, u \in U^v\}$ as a set of verified users that have re/tweeted message m.

As for the edges, we examine different metrics such as Jaccard similarity between users, and rank of a user in a message which is defined as $Rank_{(u,m)} = |\{(u', m, t') \in$

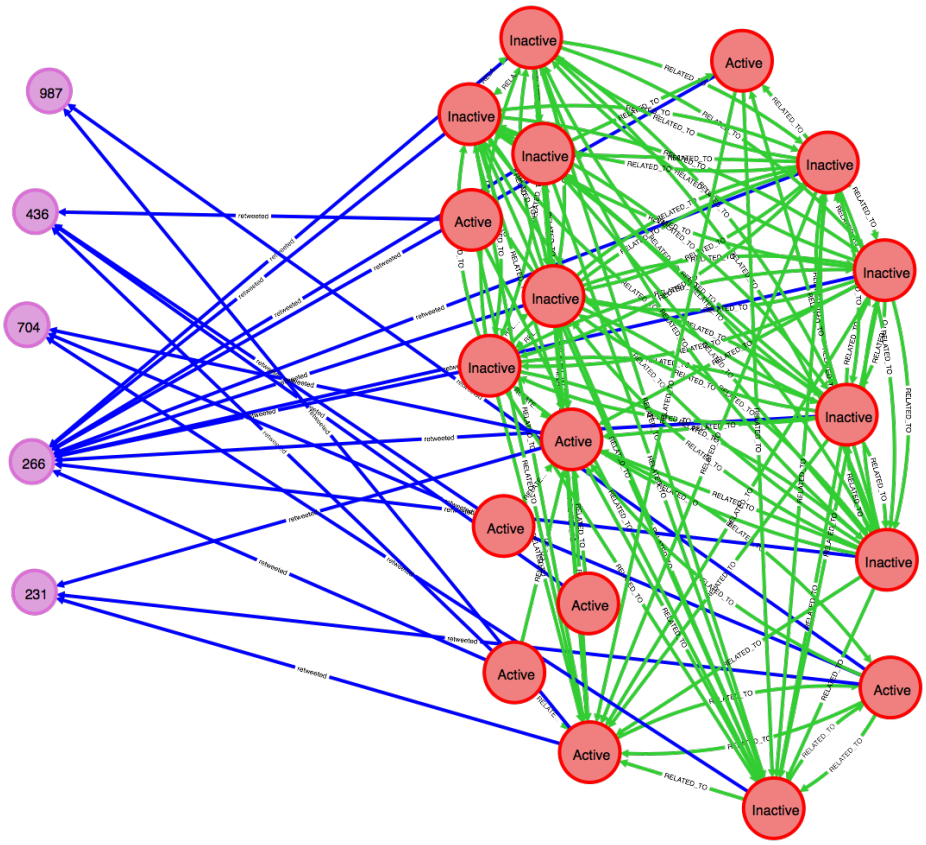


Fig. 5.1: User-message Bipartite Graph and User Graph. Red and Purple Nodes Represent Users and Messages Respectively. Users Are Labeled by Their Current Status (Active, Not Found, Suspended) and Messages With the Length of the Cascade (Degree). Blue and Green Edges Represent User-message and User-user Relationships.

$Actions|(u, m, t) \in Actions, t' < t\}|$. We also define normalized rank as:

$$NR(u, m) = 1 - \frac{Rank_{(u,m)}}{\mathcal{N}_m} \tag{5.1}$$

Our intuition behind rank metric $Rank_{(u,m)}$ is that the earlier a user has participated in spreading a message, the more important the user is. In this regard, we can also define the exponential decay of the time as:

$$\mathcal{T}_u^m = \exp(-\gamma \Delta t_u^m) \tag{5.2}$$

where $\Delta t_u^m = \{t|(u,m,t) \in Actions\} - \min(\{t'|(u',m,t') \in Actions\})$, and γ is a constant. This metric prioritizes based on the retweeting time of the message. In other words, this metric assigns different weights to different time points of a given time interval, inversely proportional to their duration from start of the cascade, i.e., smaller duration is associated with higher weight.

Using all these information, we then annotated users U based on their local and network characteristics such as degree, and PageRank. We also consider function $\mathcal{F} \in \{sum, \max, \min, avg, med, std\}$ to calculate statistics such as minimum, mean, median, maximum, and standard deviation based on their one-hop or two-hops neighbors. For example, for a given user u, mean of re/tweeted message's PageRank of user u proved to be among top predictive metrics according to our experiments. Using these intuitions, we explored the space of variants features and list those we found to be best-performing in Table 5.1.

User Graph. We represent a directed weighted user graph $G(V', E')$ where the set of nodes V' corresponds with key users. There is a link between two users if they are both key users of at least one message. There is a link from i (j) to j (i) if the number of times that "i appears before j and both are key users" is equal to or larger (smaller) than the case when "j appears before i", see Fig. 5.1. For a given node i, the set $N_i^{out} = \{i' \in V' \; s.t. \; (i,i') \in E'\}$ ($N_i^{in} = \{i' \in V' \; s.t. \; (i',i) \in E'\}$)- the set of outgoing (incoming) immediate neighbors of i. The weight of edges is determined as a variant of co-occurences of the key user pairs:

$$\mathcal{CO}_{i,j} = \frac{\begin{array}{c}|\{m|i,j \; are \; key \; users, \; \exists t, t' \; where \; t < t', \\ (i,m,t),(j,m,t') \in Actions\}|\end{array}}{\min(|\{m|i \; is \; a \; key \; user\}|, |\{m|j \; is \; a \; key \; user\}|)} \tag{5.3}$$

Using $\mathcal{CO}_{i,j}$, we then propose a weighted co-occurrence score for user i as:

$$\mathcal{CO}^w_{i,N_i} = \frac{\sum_{j \in N_i}(abs(\delta_{i,j}) + 1) \times \mathcal{CO}_{i,j}}{\sum_{j \in N_i}(abs(\delta_{i,j}) + 1)} \tag{5.4}$$

where $abs(\cdot)$ denotes the absolute value of the input. The differences between ordered joint

Table 5.1: User-message Bipartite Graph-based Metrics

Name	**Definition**
Degree	$D_v = \|\{v' \|(v,v') \in E \vee (v',v) \in E\}\|$
Cascade size statistics	$CS_{u,\mathcal{F}} = \mathcal{F}_{m \in \mathcal{N}_u} D_m$
PageRank	$PR(v) = \frac{1-d}{N} + d \sum_{v' \in \mathcal{N}_v} \frac{PR(v')}{L(v')}$
Message's PageRank statistics	$PS_{u,\mathcal{F}} = \mathcal{F}_{m \in \mathcal{N}_u} PR(m)$
Number of verified users	$Vr_m = \|\{u \|(u,m) \in E, u \in U^v\}\|$
Jaccard similarity statistics	$JS_{u,\mathcal{F}} = \mathcal{F}_{u' \in U} \frac{\|\mathcal{N}_u \cap \mathcal{N}_{u'}\|}{\|\mathcal{N}_u \cup \mathcal{N}_{u'}\|}$
Intersection statistics	$IS_{u,\mathcal{F}} = \mathcal{F}_{u' \in U} \|\mathcal{N}_u \cap \mathcal{N}_{u'}\|$
Normalized rank statistics	$NRS_{u,\mathcal{F}} = \mathcal{F}_{m \in \mathcal{N}_u} NR(u,m)$
$\mathcal{T}$ statistics	$\mathcal{T}S_{u,\mathcal{F}} = \mathcal{F}_{m \in \mathcal{N}_u} \mathcal{T}_u^m$
Verified users in the cascades statistics	$U^v S_{u,\mathcal{F}} = \mathcal{F}_{m \in \mathcal{N}_u} \|U_m^v\|$

occurrences $\delta_{i,j}$ is also defined as:

$$\begin{aligned} \delta_{i,j} = & \\ & |\{m | \exists t, t' \; s.t. \; t < t', (i,m,t), (j,m,t') \in Actions\}| \\ & \quad - |\{m | \exists t, t' \; s.t. \; t > t', (i,m,t), (j,m,t') \in Actions\}| \end{aligned} \tag{5.5}$$

The list of user graph-based metrics extracted from graph G is shown in Table 5.2. We further calculate the probability of "user j appears after user i" as:

$$P_{(j,i)} = \frac{\begin{array}{c} |\{m \in M_{vir} | \exists t, t' \, where \, t < t' \, and \\ (i,m,t), (j,m,t') \in Actions\}| \end{array}}{|\{m | (j,m,t) \in Actions\}|} \tag{5.6}$$

The average probability that user i appears before its related users $R(i)$ is also a good indicator for identifying PSM accounts:

$$CM_i = \frac{\sum_{R(i)} P_{(j,i)}}{|R(i)|} \tag{5.7}$$

We aim to evaluate users from different perspectives and these metrics have shown to be helpful for evaluating users and detecting PSM accounts.

Table 5.2: User Graph-based Metrics

Description	**Definition**
Degree	$\lvert N_i^{out} \rvert$
Outgoing co-occurrence score statistics	$\mathcal{COS}_{i,\mathcal{F}}^{out} = \mathcal{F}_{j \in \mathcal{N}_i^{out}} \mathcal{CO}_{i,j}$
Incoming co-occurrence score statistics	$\mathcal{COS}_{i,\mathcal{F}}^{in} = \mathcal{F}_{j \in \mathcal{N}_i^{in}} \mathcal{CO}_{i,j}$
Weighted co-occurrence score	$\mathcal{CO}_{i,N_i^{out}}^{w}$
Number of outgoing verified users	$\lvert \{j \lvert j \in N_i^{out}, j \in U^v\} \rvert$
Number of incoming verified	$\lvert \{j \lvert j \in N_i^{in}, j \in U^v\} \rvert$
Triangles	Number of triangles
Clustering coefficient	$CC_i = \frac{\lvert \{(j,k) \lvert j,k \in N_i, (j,k) \in E'\} \rvert}{\lvert N_i \rvert \times (\lvert N_i \rvert - 1)}$

5.2.2 Problem Statement

Our goal is to find the potential PSM accounts from the cascades. In the previous section, we discussed causality metrics, and defined diverse set of features using both user-message bipartite and user graphs where these metrics can discriminate the users of interest.

Problem. (Early PSM Account Detection). *Given Action log Actions, causality and structural metrics, we wish to identify set of key users that are PSM accounts.*

5.3 PSM Account Detection Algorithm

We employ supervised, and semi-supervised approaches for detecting PSM accounts. Proposed metrics are scalable and can be calculated efficiently using map-reduce programming model and storing data in a graph-based database. To such aim, we used Neo4j to store data and calculated most of the structural metrics using Cypher query language [83].

5.3.1 Supervised Learning Approach

We evaluate several supervised learning approaches including logistic regression (LR), naive bayes (NB), k-nearest neighbors (KNN) and random forest (RF) on the same set of features. We also develop a dense deep neural network structure using Keras. As for the deep neural network and in order to find the best architecture and hyperparamters, we utilize the random search method. Many model structures were tested and Fig. 5.2 illustrates the best architecture.

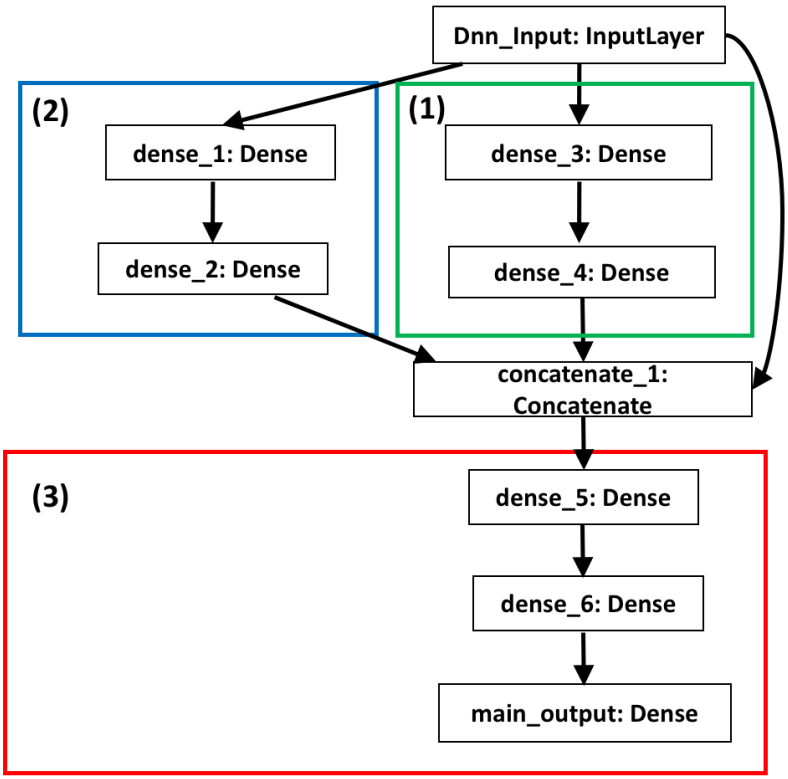


Fig. 5.2: The Proposed Deep Neural Net Structure

As we can see from Fig. 5.2, the proposed deep neural net, in fact, consists of three dense deep neural net structures. The first two structures are of the same, but the activation functions of their layers are different. The intuition is that we aimed to capture the most

useful information from the input data and our experiments show the ReLU and Sigmoid activation functions can contribute to this. Specifically, these two structures are aimed to filter the noises in the input data and prepare clean inputs to feed into the third structure. In this regard, the outputs of these two structures along with the input data are concatenated into one vector and this vector is fed into another dense deep neural net. Finally, the output of this structure is fed into a regular output layer. To avoid overfitting, we used dropout method. In the proposed framework, the binary cross entropy loss function is minimized and the best optimizers are reported as Adam and Adagrad.

5.3.2 Self-training Semi-supervised Learning Approach

Semi-supervised algorithms [77, 84] use unlabeled data along with the labeled data to better capture the shape of the underlying data distribution and generalize better to new samples. Here, we propose a *Weighted Self-training Algorithm* (WSET) shown in Algorithm 4 to address such problem. We start with small amount of labeled training data and iteratively add users with high confidence scores from unlabeled data to the training set. Lets denote labeled data $L = \{\mathbf{u}_i, l_i\}$ and unlabeled data $U = \{\mathbf{u}_j\}$. Labeled data is split to training set L_t and development set L_d. We then iteratively train a classifier using training set and predict the confidence scores for development set and unlabeled data. Based on the *confidence score* obtained from *development set*, a *threshold* is determined. We then select *all samples* from *unlabaled data* that *satisfy the threshold*. Next, those samples are *removed* from *unlabeled set* and are *added* to the *training set*. The termination condition is determined based on at most θ_{tr} drop in accuracy on the development set or minimum number of selected users by algorithm.

There are still two main questions that need to be answered:

Q1. Should all training samples be weighted equally?

Algorithm 4 Weighted Self-training Algorithm (WSeT)

procedure WSET($L = \{\mathbf{u}_i, l_i\}, U = \{\mathbf{u}_j\}, \alpha, \beta, \theta_{pr}, \theta_{tr}$)
  Split L to training set L_t and development set L_d
  $L_t.w_c = 1$
  $it = 1$
  $m =$ Train a classification model using L_t
  $L_d.p =$ confidence score p using m of L_d
  $c =$ accuracy of model m on L_d
  $c' = c$
  while $c' >= c - \theta_{tr}$ **do**
    $U.p =$ confidence score p using m of U
    Update L_t and U by Algorithm 5 ($L_t, L_d, U, \alpha, \beta, \theta_{pr}, it$)
    $m =$ Train a classification model using L_t
    $L_d.p =$ confidence score p using m of L_d
    $c' =$ accuracy of model m on L_d
    $it = it + 1$
  return L_t

Q2. How should a threshold be determined for adding unlabeled data to the labeled set?

Since the prediction mistake reinforces itself, and the prediction error increases by number of iterations, the way we choose samples is of importance. According to our experiments, all training samples should not be weighted equally. We found the *exponential decay weighting* approach as the most efficient one (see Q1). Considering a sample with confidence p_l associated to a specific label l in iteration it, the exponential decay weighting approach is defined as:

$$\exp(-\beta \times it \times (\frac{1}{1-p_l})) \tag{5.8}$$

where β is a parameter. To answer the second question, we pick the threshold to have the minimum precision of θ_{pr} on development set in each iteration. Since the precision decreases as the algorithm iterates, the threshold is required to be adjusted in order to make sure the top ranked and qualified samples are picked up. Mathematically, the updated threshold in each iteration is defined as follows:

$$\theta_{pr} - \alpha \times (it - 1) \tag{5.9}$$

where α is a parameter, $\alpha \in [0, \frac{1}{it-1}], it > 0$. We pick 0.005 for the experiments. If $it = 1$, the threshold is equal to θ_{pr}. As the number of iteration increases the threshold is updated according to the product of α and iteration number it. This approach can make sure that we are picking samples with acceptable confidence. Algorithm 5 presents our approach for updating labeled and unlabaled datasets.

Algorithm 5 Update Weighted Self-training Datasets Algorithm (UpDWSeT)

procedure UPDWSET($L_t, L_d, U, \alpha, \beta, \theta_{pr}, it$)
  $S = \emptyset$
  for $l \in [True, False]$ **do**
    $thr = FindPrecisionThreshold(L_d, \theta_{pr} - \alpha \times (it - 1), label = l)$
    $S = S \cup \{\mathbf{u} \in U | \mathbf{u}.p \geq thr\}$
  $U = U - S$
  $S.w_c = \exp(-\beta \times it \times (\frac{1}{1-p}))$
  $L_t = L_t \cup S$
  return L_t, U

5.4 ISIS Dataset

Our dataset is the same as previous chapter. The dataset includes tweets and the associated information such as user ID, re-tweet ID, hashtags, number of followers, number of followee, content, date and time. About 53M tweets are collected based on the 290 hashtags such as State of the Islamic-Caliphate, and Islamic State. In this chapter, we only use tweets (more than 9M) associated with viral cascades. Dataset is labeled based on their status in Nov. 2018 on Twitter. Accounts are not active if they are suspended or deleted. Less than 24% of the users are inactive. Inactive users are representative of automatic and terrorism accounts aiming to disseminate their propaganda and manipulate the statistics of the hashtags of their interest.

5.5 Results and Discussion

We implement part of our code in Scala Spark and Python 2.7x and run it on a machine equipped with an Intel Xeon CPU (2 processors of 2.4 GHz) with 256 GB of RAM running Windows 7. We also implement most of structural metrics in Cypher query language. We create the graphs using Neo4j [83] on a machine equipped with an Intel Xeon CPU (2 processors of 2.4 GHz) with 520 GB of RAM. We set the parameter ϕ as 0.5 to label key users. That is, we are looking for the users that participate in the *action* before the number of participants gets twice.

In the following sections, first we look at the baseline methods. Then we address the performance of two proposed approaches (see Section 5.3): (1) *Supervised Learning Approach*: applying different supervised learning methods on proposed metrics, (2) *Self-training Semi-supervised Learning Approach*: selecting users by applying Algorithm 4. The intuition behind this approach is to select users with the high probability of being either PSM or non-PSM (normal user) from unlabeled data and then adding them to the

Table 5.3: Statistics of the Datasets Used in Experiments.

Name	PSM accounts	Normal accounts	Total
$\mathcal{A}$	19,859	65,417	85,276
$\mathcal{B}$	137,248	585,396	722,644

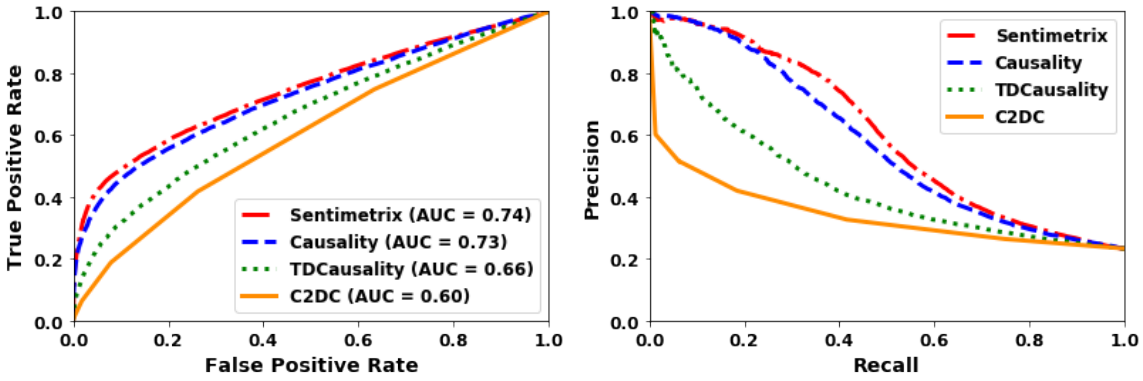
training set in order to improve the performance. We evaluate methods based on both Precision-Recall and Receiver Operating Characteristics (ROC) curves. Note that in all experiments, the training, development, and test sets are imbalanced with more normal user accounts than PSM accounts. The statistics of the datasets are presented in Table 5.3. Dataset $\mathcal{A}$ is randomly selected from dataset $\mathcal{B}$ using sklearn library [52]. Note that, all random selections of data in the experiments have been done using sklearn library. We repeated the experiments 3 times and picked the median output. It is worth to mention that the variance among the results was negligible. In this problem, our goal is to achieve high precision while maximizing the recall. The main reason is labeling an account as PSM means it should be deleted. However, removing a normal user is costly. Therefore, it is important to have a high precision to prevent removing the normal users.

5.5.1 Baseline Methods

We have compared our results with existing work for detecting PSM accounts [66, 5] or bots [77].

Causality. This paper presents a set of causality metrics and unsupervised label propagation model to identify PSM accounts [66]. However, since our approach is supervised, we only use the causality metrics and evaluate its performance in a supervised framework.

C2DC. This approach uses time decay causal community detection-based classification to detect PSM accounts [5]. We also considered time decay causal metrics with random forest



(a) ROC Curve

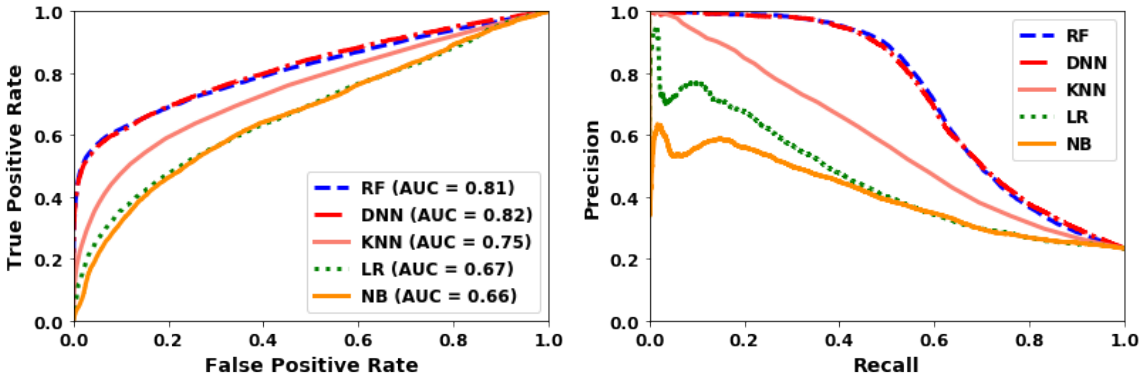
(b) Precision-Recall Curve

Fig. 5.3: Performance of the Baseline Methods on Dataset $\mathcal{A}$

as another baseline method (TDCausality).

Sentimetrix. This approach is proposed by the top-ranked team in the DARPA Twitter Bot Challenge [77]. We consider all features that we could extract from our dataset. Our features include tweet syntax (average number of hashtags, average number of user mentions, average number of links, average number of special characters), tweet semantics (LDA topics), and user behaviour (tweet spread, tweet frequency, tweet repeats). The proposed method starts with a small seed set and propagate the labels. As we have enough labeled dataset for the training set, we use random forest as the learning approach.

We use dataset $\mathcal{A}$ to evaluate different approaches. Fig. 5.3a shows the precision-recall curve for these methods. As it is shown, in the supervised framework, *Sentimetrix* outperforms all approaches in general. Also, *Causality* is a comparable approach with Sentimetrix with the constraint that the precision is no less than 0.9 as illustrated in Fig. 5.3b. Note that, most of the features used in the previous bot detection work take advantage of content and network structure of users. However, this is not the case in our proposed metrics and approach.



(a) ROC Curve

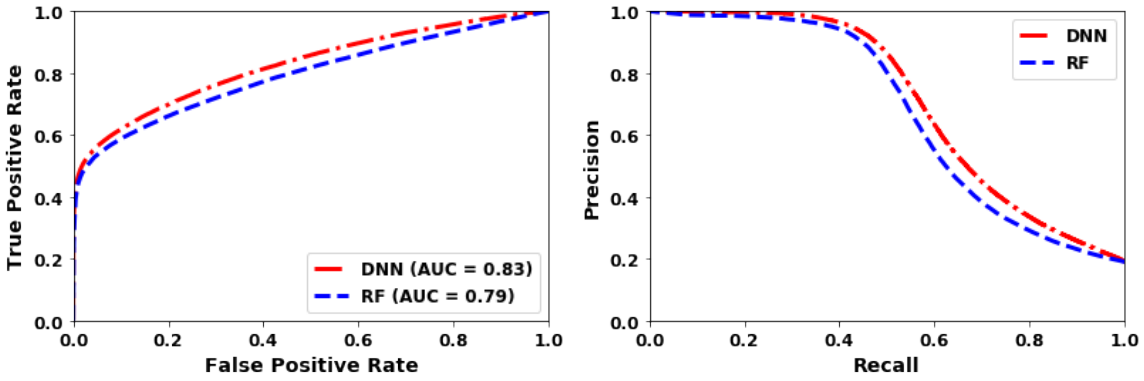
(b) Precision-Recall Curve

Fig. 5.4: Performance of Different Supervised Approaches Using Proposed Metrics on Dataset $\mathcal{A}$

5.5.2 Supervised Learning Approach

In this section, we describe the classification results using proposed metrics with different learning approaches. We used both datasets for this experiment. First, we use the same dataset as we used in baseline experiments ($\mathcal{A}$). Then, we use dataset $\mathcal{B}$ for comparing top methods which is 8.5 times larger than dataset $\mathcal{A}$.

Fig. 5.4a shows the ROC curve for different approaches. As it is shown, deep neural network achieved the highest area under the curve. Note that, the deep neural network is comparable with random forest as it is shown in Fig. 5.4b on Dataset $\mathcal{A}$. The proposed approaches could improve the recall from 0.22 to 0.49 with the precision of 0.9. According to the random forest, top features are from all categories including causality metrics: ϵ_{nb}, ϵ_{wnb}, user-message bipartite graph-based metrics: user's PageRank $PR(u)$, median of retweeted message's PageRank $PS_{u,med}$, degree D_u, mean of verified users in his messages $U^v S_{u,mean}$, $\mathcal{T}S_{u,med}$, $\mathcal{T}S_{u,mean}$, median of length of the cascades $CS_{u,med}$, user graph-based metrics: weighted co-occurrence score $\mathcal{CO}^w_{u,N_u}$.



(a) ROC Curve

(b) Precision-Recall Curve

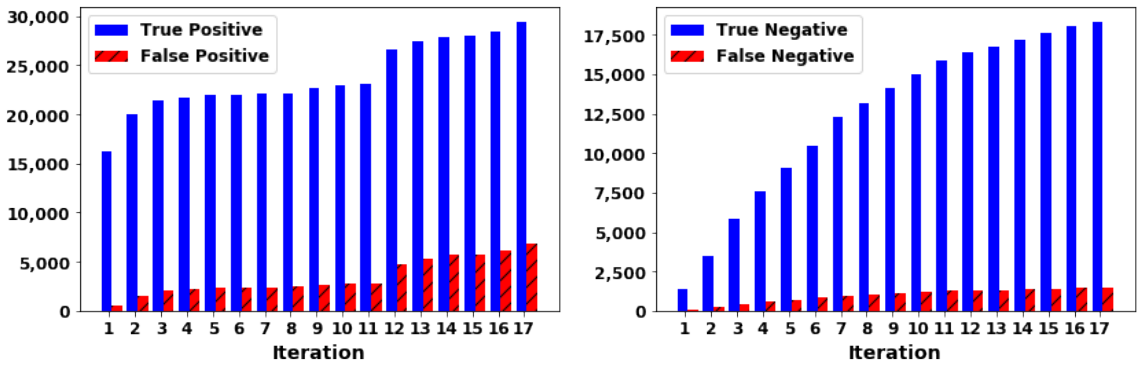
Fig. 5.5: Performance of the Top Two Supervised Approaches Using Proposed Metrics on Dataset $\mathcal{B}$

In Fig. 5.5b, we probe the performance of the top two supervised approaches on larger dataset $\mathcal{B}$. Deep neural network is able to achieve the recall of 0.48 with the precision of 0.9. It is also able to achieve AUC of 0.83 on this dataset (Fig. 5.5a).

5.5.3 Self-training Semi-supervised Learning Approach

In this experiment, we randomly select 300 PSM and 300 normal users from dataset $\mathcal{B}$ for training and development sets and the rest of the dataset was considered as unlabeled data. We conduct two types of experiments:

WSeT Algorithm. In this experiment, we evaluate the self-training semi-supervised approach using Algorithm 4. In this approach, we iteratively update the training set and the termination condition is accuracy of the model on the development set. We set the parameters as $\theta_{pr} = 1$, $\alpha = 0.05$, $\theta_{tr} = 0.03$. We use a random forest classifier to train the model. The cumulative number of true positive and false positive is shown in Fig. 5.6a. With using 300 PSM accounts as seed set, WSeT can find 29,440 PSM accounts with the precision of 0.81. Note that, we can stop the algorithm earlier. In this case, the precision



(a) PSM Users

(b) Normal Users

Fig. 5.6: Cumulative Number of Selected Users Using WSeT Algorithm on Dataset $\mathcal{B}$

varies from 0.97 to 0.81. Fig. 5.6b illustrates cumulative number of selected users as normal users by WSeT. As it is shown, the number of true negatives (selected normal accounts as normal users) is 18,343 with precision of 0.93.

Supervised WSeT Algorithm. In previous experiment, we assume that the supervisor checks accounts labeled as PSM by WSeT at the end. However, this process can be done iteratively. Here, we assume that the supervisor evaluates the *PSM labeled accounts by WSeT* in each iteration and verify if they are either true or false positive. Therefore, these labels along with the non-PSM labeled accounts by WSeT are fed into WSeT. According to our results, the number of true positive increases to 80,652 with the precision of more than 0.8. That is, using this approach we can increase the number of true positive PSM accounts 2.7 times.

5.6 Related Work

In summary, majority of previous work was based on three fundamental assumptions which make them different from our work. First, *the information of the network is known* [77, 29, 7, 1]. This assumption may not hold in reality. Second, *they are language-dependent*

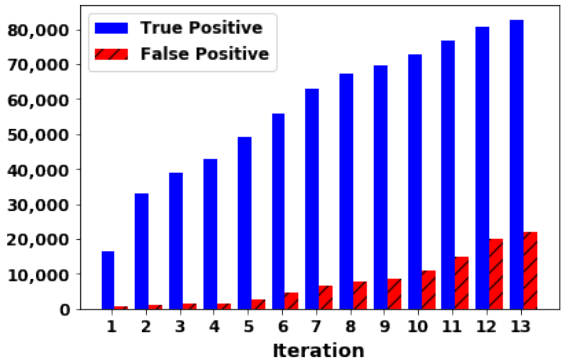


Fig. 5.7: Cumulative Number of Selected Users as PSM Accounts Using Supervised WSeT Algorithm on Dataset $\mathcal{B}$

[77, 45]. Third, *the majority of botnet detection algorithms focused on bots in general.* That is, they did only consider the bots *equally* [45, 22] while in this work, we identify PSM accounts that spread harmful viral information. Below, we distinguish our work from the literature in details.

Identifying PSM accounts. Compared to previous PSM account detection work in [66, 5], we propose graph-based features and unlike the unsupervised learning approach in [66], we expand out both the supervised and semi-supervised learning methods. We also develop a deep neural network and show how our proposed approaches improve them significantly.

Identifying Automatic Accounts. DARPA conducted the Twitter bot detection challenge to identify and eliminate influential bots [77]. Most of the previous work extracted different sets of features (tweet syntax, tweet semantics, temporal behavior, user profile, friends and network features) and conducted supervised or semi-supervised approaches [77, 21, 22]. However, without using content, and network structure, they perform poorly. Also, some of the features such as tweet semantics depend on the language. It is yet a challenge to apply these features to other languages such as Arabic. Moreover, unlike our proposed method, they use friendship (follower/followee) network structure and account related attributes.

Analysis of Terrorist Groups and Detection of Water Armies. Terrorist groups use social

media for propaganda dissemination [3]. Benigni et al. [7] conducted vertex clustering and classification to find Islamic Jihad Supporting Community on Twitter. Abdokhodair et al. [1] studied the behaviors and characteristics of Syrian social botnet. Chen et al. [18] found that within the context of news report comments, user-specific measurements can distinguish water army from normal users. Our work is different from them since these methods used features related to the accounts and network.

5.7 Conclusion

In this chapter, we conducted a data-driven study of inductive and abductive reasonings on the pathogenic social media accounts. We proposed supervised and semi-supervised frameworks to detect these users. We achieved the precision of 0.9 with F1 score of 0.63 using the supervised framework. In semi-supervised framework, we are able to detect more than 29K PSM users by using only 600 labeled data for training and development sets with the precision of 0.81. Our approaches identify these users without using network structure, cascade path information, content information. We believe our technique can be applied in areas such as detection of water armies and fake news campaigns. In the future, the combination of the supervised approach with the semi-supervised and unsupervised approaches can provide us with a complete pipeline for identifying PSM accounts.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This dissertation focuses on the relationships between different types of reasoning, known as deductive, inductive, and abudctive, and applying them to address real-world problems. While data-driven variants of these forms of reasoning have been applied separately, there is less work on how these approaches relate and lend themselves to practical applications. Therefore, as the first step in this thesis, we proposed a method for identifying potentially violent criminal gang members using *inductive* and *deductive* reasonings. In induction, we leveraged features derived from the co-arrestee network of criminal gang members and other meta-data to create a supervised model. In deduction, we used the best model to identify the potential violent individuals from existing criminal individuals. It is observed that our method outperforms the ones in the literature.

We then focused on *geospatial abductive inference method* to address the missing person problem. We developed a data-driven variant of geospatial abductive inference for finding missing persons. For a given missing person case, and its potential locations reported by experts, we developed a toolkit to rank-orders the search area. The experimental results showed that our approach is able to reduce the total search area for standard searched and when dog team assets obtain a detection.

Utilizing *induction* and *abduction* reasonings, we then studied the pathogenic social media accounts. We proposed a causality-based unsupervised method to detect these accounts. Considering an extremist cascade as an agent, we aimed to find if an agent is a pathogenic user. We introduced an unsupervised causality-based framework as well as label

propagation approach. This approach identified these users without using network structure, cascade path information, content and user's information. We evaluated our approach using Twitter dataset.

In Chapter 5, we expand upon the causal inference framework with graph-based metrics in order to distinguish PSM accounts from normal users to increase precision and recall. We thus proposed both supervised and semi-supervised approaches without taking the network structure, cascade path information and content into account. Results on a real-world dataset from Twitter highlighted the advantage of our proposed frameworks.

6.2 Future Work

This thesis can be extended in several following directions:

- Working with the police to identify other sources of data to build a more complete social network of the offenders for the research work in Chapter 2.
- Utilizing a probabilistic variant of the feasibility function, incorporating other features such as missing person's corresponding region, age, gender into the proposed model in Chapter 3.
- Applying our technique in Chapter 4 in the areas such as detection of water armies and fake news campaigns.
- Combining the proposed semi-supervised approach with unsupervised one in Chapter 4 in order to find the initial seed set for the semi-supervised approach.

BIBLIOGRAPHY

[1] N. Abokhodair, D. Yoo, and D. W. McDonald. Dissecting a social botnet: Growth, content and influence in twitter. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 839–851. ACM, 2015.

[2] B. Akinkunmi and P. C. Bassey. A Logic of Spatial Qualification Using Qualitative Reasoning Approach. *International Journal of Artificial Intelligence & Applications*, 4(2):45, 2013.

[3] S. Al-khateeb and N. Agarwal. Examining botnet behaviors for propaganda dissemination: A case study of isil's beheading videos-based propaganda. In *Data Mining Workshop, 2015 IEEE International Conference on*, pages 51–57, 2015.

[4] H. Allcott and M. Gentzkow. Social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research, 2017.

[5] H. Alvari, E. Shaabani, and P. Shakarian. Early identification of pathogenic social media accounts. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 169–174. IEEE, 2018.

[6] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of the 17th international conference on World Wide Web*, pages 895–904. ACM, 2008.

[7] M. Benigni and K. M. Carley. From tweets to intelligence: Understanding the islamic jihad supporting community on twitter.

[8] J. A. Bertetto. Countering criminal street gangs: Lessons from the counterinsurgent battlespace. *Journal Article— Nov*, 15(4):30am, 2012.

[9] A. A. Braga, D. M. Hureau, and A. V. Papachristos. Deterring gang-involved gun violence: measuring the impact of boston?s operation ceasefire on street gang behavior. *Journal of Quantitative Criminology*, 30(1):113–139, 2014.

[10] P. Brantingham and P. Brantingham. Crime pattern theory. In *Environmental criminology and crime analysis*, pages 78–93. Willan, 2008.

[11] U. M. P. Bureau. Missing persons: Data and analysis 2012–2013. *London: National Crime Agency*, 2014.

[12] T. Bylander, D. Allemang, M. C. Tanner, and J. R. Josephson. The Computational Complexity of Abduction. *Artif. Intell.*, 49(1-3):25–60, 1991.

[13] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 15–15. USENIX Association, 2012.

[14] N. C. I. Center. 2016 ncic missing person and unidentified person statistics, 2017.

[15] N. C. I. Center. 2017 ncic missing person and unidentified person statistics, 2018.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[17] C. Chen, K. Wu, V. Srinivasan, and K. Bharadwaj. The best answers? think twice: online detection of commercial campaigns in the cqa forums. In *Advances in Social Networks Analysis and Mining, 2013 IEEE/ACM International Conference on*, pages 458–465, 2013.

[18] C. Chen, K. Wu, V. Srinivasan, and X. Zhang. Battling the internet water army: Detection of hidden paid posters. In *Advances in Social Networks Analysis and Mining, 2013 IEEE/ACM International Conference on*, pages 116–120, 2013.

[19] L. Console, L. Portinale, and D. T. Dupré. Focussing Abductive Diagnosis. *AI Commun.*, 4(2/3):88–97, 1991.

[20] L. Console, M. L. Sapino, and D. T. Dupré. The Role of Abduction in Database View Updating. *J. Intell. Inf. Syst.*, 4(3):261–280, 1995.

[21] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 273–274. International World Wide Web Conferences Steering Committee, 2016.

[22] J. P. Dickerson, V. Kagan, and V. Subrahmanian. Using sentiment to detect bots on twitter: Are humans more opinionated than bots? In *Advances in Social Networks Analysis and Mining, 2014 IEEE/ACM International Conference on*, pages 620–627, 2014.

[23] S. do Lago Pereira and L. N. de Barros. Planning with Abduction: A Logical Framework to Explore Extensions to Classical Planning. In A. L. C. Bazzan and S. Labidi, editors, *SBIA*, volume 3171 of *Lecture Notes in Computer Science*, pages 62–72. Springer, 2004.

[24] T. Eiter and G. Gottlob. The complexity of logic-based abduction. *Journal of the ACM*, 42:3–42, 1995.

[25] P. Expert, T. S. Evans, V. D. Blondel, and R. Lambiotte. Uncovering space-independent communities in spatial networks. *Proceedings of the National Academy of Sciences*, 108(19):7663–7668, 2011.

[26] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[27] S. George, X. Wang, J. Lin, B. Qu, and J.-C. Liu. MECH: Algorithms and Tools for Automated Assessment of Potential Attack Locations. Technical report, Texas A & M University, College Station, 2015.

[28] W. Gorr and R. Harries. Introduction to crime forecasting. *International Journal of Forecasting*, 19(4):551–555, 2003.

[29] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM, 2010.

[30] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420–1443, 1978.

[31] A. Gupta, H. Lamba, and P. Kumaraguru. $1.00 per rt# bostonmarathon# prayforboston: Analyzing fake content on twitter. In *eCrime Researchers Summit (eCRS), 2013*, pages 1–12. IEEE, 2013.

[32] H. Han, W.-Y. Wang, and B.-H. Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, pages 878–887. Springer, 2005.

[33] J. C. Howell and E. Griffiths. *Gangs in America's communities*. Sage Publications, 2018.

[34] A. C. Kakas and P. Mancarella. Database Updates through Abduction. In D. McLeod, R. Sacks-Davis, and H.-J. Schek, editors, *VLDB*, pages 650–661. Morgan Kaufmann, 1990.

[35] C. Kang. Fake news onslaught targets pizzeria as nest of child-trafficking. *The New York Times*, 2016.

[36] M. Khader. *Combating Violent Extremism and Radicalization in the Digital Era*. IGI Global, 2016.

[37] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse. Identification of influential spreaders in complex networks. *Nature physics*, 6(11):888, 2010.

[38] S. Kleinberg and B. Mishra. The temporal logic of causal structures. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 303–312, 2009.

[39] A. Koutsioumpas. *Applications of Mathematics and Informatics in Science and Engineering*, chapter Abductive Reasoning in 2D Geospatial Problems, pages 333–347. Springer International Publishing, Cham, 2014.

[40] A. Krause, J. Leskovec, C. Guestrin, J. Vanbriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 2008.

[41] H. Liu and D. E. Brown. Criminal incident prediction using a point-pattern-based density model. *International journal of forecasting*, 19(4):603–622, 2003.

[42] J. M. McGloin, C. J. Sullivan, A. R. Piquero, and S. Bacon. Investigating the stability of co-offending and co-offenders among a sample of youthful offenders. *Criminology*, 46(1):155–188, 2008.

[43] E. McMenamin. Databasing the disappeared and deceased: a review of the resources available in missing and unidentified persons cases. 2008.

[44] C. Morselli. *Inside criminal networks*, volume 8. Springer, 2009.

[45] F. Morstatter, L. Wu, T. H. Nazer, K. M. Carley, and H. Liu. A new approach to bot detection: striking the balance between precision and recall. In *Advances in Social Networks Analysis and Mining, 2016 IEEE/ACM International Conference on*, pages 533–540, 2016.

[46] C. Overall and G. Day. The hammer gang: an exercise in spatial analyis of an armed robbery series using the probability grid method. *Crime Mapping Case Studies*, pages 55–62, 2008.

[47] M. Pagnucco. *The Role of Abductive Reasoning within the Process of Belief Revision*. PhD thesis, Basser Department of Computer Science, University of Sydney, 1996.

[48] A. V. Papachristos, A. A. Braga, and D. M. Hureau. Social networks and the risk of gunshot injury. *Journal of Urban Health*, 89(6):992–1003, 2012.

[49] A. V. Papachristos, D. M. Hureau, and A. A. Braga. The corner and the crew: the influence of geography and social networks on gang violence. *American sociological review*, 78(3):417–447, 2013.

[50] A. V. Papachristos, C. Wildeman, and E. Roberto. Tragic, but not random: The social contagion of nonfatal gunshot injuries. *Social Science & Medicine*, 125:139–150, 2015.

[51] D. Paulo, B. Fischl, T. Markow, M. Martin, and P. Shakarian. Social network intelligence analysis to combat street gang violence. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 1042–1049. IEEE, 2013.

[52] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[53] Y. Peng and J. Reggia. *Abductive inference models for diagnostic problem-solving*. Symbolic computation. Springer-Verlag, New York, 1990.

[54] Y. Peng and J. A. Reggia. Plausibility of Diagnostic Hypotheses: The Nature of Simplicity. In *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, August 11-15, 1986. Volume 1: Science.*, pages 140–147, 1986.

[55] R. R. Petersen and U. K. Wiil. Crimefighter investigator: A novel tool for criminal network investigation. In *Intelligence and Security Informatics Conference (EISIC), 2011 European*, pages 197–202. IEEE, 2011.

[56] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

[57] D. K. Rossmo. *Geographic profiling*. CRC press, 1999.

[58] A. Sadilek and H. Kautz. Location-based Reasoning About Complex Multi-agent Behavior. *J. Artif. Int. Res.*, 43(1):87–133, Jan. 2012.

[59] A. Sadilek and H. Kautz. Modeling success, failure, and intent of multi-agent activities under severe noise. pages 9–63, 2012.

[60] A. Sadilek and H. A. Kautz. Recognizing Multi-Agent Activities from GPS Data. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, 2010.

[61] P. Santos and M. Shanahan. Hypothesising Object Relations from Image Transitions. In F. van Harmelen, editor, *ECAI*, pages 292–296. IOS Press, 2002.

[62] J. Schroeder, J. Xu, and H. Chen. Crimelink explorer: Using domain knowledge to facilitate automated crime association analysis. In *International Conference on Intelligence and Security Informatics*, pages 168–180. Springer, 2003.

[63] S. B. Seidman. Network structure and minimum degree. *Social networks*, 5(3):269–287, 1983.

[64] E. Shaabani, A. Aleali, P. Shakarian, and J. Bertetto. Early identification of violent criminal gang members. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2079–2088. ACM, 2015.

[65] E. Shaabani, H. Alvari, P. Shakarian, and J. Snyder. Mist: Missing person intelligence synthesis toolkit. *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, 2016.

[66] E. Shaabani, R. Guo, and P. Shakarian. Detecting pathogenic social media accounts without content or network structure. In *Data Intelligence and Security (ICDIS), 2018 1st International Conference on*, pages 57–64. IEEE, 2018.

[67] E. Shaabani, A. Sadeghi-Mobarakeh, H. Alvari, and P. Shakarian. An end-to-end framework to identify pathogenic social media accounts on twitter. In *Data Intelligence and Security (ICDIS), 2019 2nd International Conference on*. IEEE, 2019.

[68] P. Shakarian, J. P. Dickerson, and V. S. Subrahmanian. Adversarial Geospatial Abduction Problems. *ACM TIST*, 3(2):34, 2012.

[69] P. Shakarian, M. Nagel, B. Schuetzle, and V. S. Subrahmanian. Abductive Inference for Combat: Using SCARE-S2 to Find High-Value Targets in Afghanistan. In D. G. Shapiro and M. P. J. Fromherz, editors, *IAAI*. AAAI, 2011.

[70] P. Shakarian, J. Salmento, W. Pulleyblank, and J. Bertetto. Reducing gang violence through network influence based targeting of social programs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1829–1836. ACM, 2014.

[71] P. Shakarian and V. Subrahmanian. Region-based Geospatial Abduction with Counter-IED Applications. In U. K. Wiil, editor, *Counterterrorism and Open Source Intelligence*. Springer, 2010.

[72] P. Shakarian and V. Subrahmanian. *Geospatial Abduction: Principles and Practice*. SpringerLink : Bücher. Springer New York, 2011.

[73] P. Shakarian, V. Subrahmanian, and M. L. Spaino. SCARE: A Case Study with Baghdad. In *Proceedings of the Third International Conference on Computational Cultural Dynamics*. AAAI, 2009.

[74] P. Shakarian, V. Subrahmanian, and M. L. Spaino. GAPs: Geospatial Abduction Problems. *ACM Transactions on Intelligent Systems and Technology*, 2010.

[75] M. Shanahan. Noise and the Common Sense Informatic Situation for a Mobile Robot, 1996.

[76] J. F. Stollsteimer. A Working Model for Plant Numbers and Locations. *Journal of Farm Economics*, 45(3):631–645, 1963.

[77] V. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, and F. Menczer. The darpa twitter bot challenge. *Computer*, 49(6):38–46, 2016.

[78] P. Suppes. *A probabilistic theory of causality*. North-Holland Publishing Company Amsterdam, 1970.

[79] M. A. Tayebi, M. Ester, U. Glässer, and P. L. Brantingham. Spatially embedded co-offence prediction using supervised learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1789–1798. ACM, 2014.

[80] M. A. Tayebi, M. Jamali, M. Ester, U. Glässer, and R. Frank. Crimewalker: a recommendation model for suspect investigation. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 173–180. ACM, 2011.

[81] K. Wang, Y. Xiao, and Z. Xiao. Detection of internet water army in social network. In *2014 International Conference on Computer, Communications and Information Technology (CCIT 2014)*. Atlantis Press, 2014.

[82] D. J. Watts and S. H. Strogatz. Collective dynamics of ?small-world?networks. *nature*, 393(6684):440, 1998.

[83] J. Webber and I. Robinson. *A programmatic introduction to neo4j*. Addison-Wesley Professional, 2018.

[84] X. Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2(3):4, 2006.

[85] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. 2002.