Detection of Organized Activity in Online Escort Advertisements

Aayushi Kulshrestha

Computer Science

McGill University

Montreal, Quebec, Canada

April 20, 2021

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Masters of Science

Abstract

Human trafficking is an age old problem that continues to affect 25 million people worldwide. A majority of traffickers now leverage online media to advertise their victims, making it harder for law enforcement to reach offenders. Given the large volume of ads posted daily, how can we help expedite the crackdown on these organized crimes? Past studies suggest that most victims have minimal involvement in the content, rate, and images posted online. Often, ads originating from a single trafficker can be tracked down using the presence of strong similarities. Finding these cases manually is an arduous task that could take up weeks of valuable investigation time. Moreover, discerning whether the identified ad is advertising an independent sex worker or a trafficking victim is a very difficult task. Thus, we leverage the organized behavior of traffickers to look for groups of ads with striking similarities.

In this work, we present **TrafficLight** - an unsupervised, generalized approach that finds suspiciously connected cases of organized activity in escort ads and summarizes the evidence for faster investigation. TrafficLight provides easily interpretable **summarized results**, with an estimated *25x speedup* in the detection of such criminal groups. This is in contrast to most of the previous work done in this domain which either rely on matching known trafficking related keywords or training a language classifier, both of which are heavily dependent on past knowledge. We remove this dependency of obtaining a labeled data from experts as it is a costly and time consuming process. We present results on two real world datasets to validate our method. Using a previously labeled dataset, we show that top results from TrafficLight are highly likely to be trafficking. Moreover, TrafficLight is also able to find **novel cases** which can only be found by looking at connections between ads, but go unnoticed if looked at individually. When applied to a live dataset, results from TrafficLight correlate with the weak ground-truth available. With a further *meta-clustering*, that is, grouping clusters of ads, we detected several M.O.s (modus operandi = types of behaviors) - the most striking one was the 'smoke screen' behavior: several clusters of ads, some having different, but non-operational, phone numbers, probably aiming to frustrate the investigators with dead-end leads.

Sommaire

La traite des êtres humains est un problème ancien qui continue de toucher 25 millions de personnes dans le monde. La majorité des trafiquants utilisent désormais les médias en ligne pour faire la publicité de leurs victimes, ce qui rend plus difficile pour les forces de l'ordre d'atteindre les délinquants. Étant donné le grand nombre d'annonces publiées quotidiennement, comment pouvons-nous contribuer à accélérer la répression de ces crimes organisés? Des études antérieures suggèrent que la plupart des victimes sont peu impliquées dans le contenu, le tarif et les images mis en ligne. Souvent, les annonces provenant d'un seul trafiquant peuvent être retrouvées grâce à la présence de grandes similitudes. Trouver ces cas manuellement est une tâche ardue qui prendrait des semaines de précieux temps d'enquête. De plus, discerner si l'annonce identifiée fait la publicité d'un travailleur du sexe indépendant ou d'une victime de la traite humaine est une tâche très difficile. Ainsi, nous nous appuyons sur le comportement organisé des trafiquants pour rechercher des groupes d'annonces présentant des similitudes frappantes.

Dans ce travail, nous présentons **TrafficLight** - une approche non supervisée et généralisée qui permet de trouver des cas d'activités organisées suspectes dans

des annonces d'escortes et de résumer les preuves pour accélérer l'enquête. TrafficLight fournit des résultats résumés facilement interprétables, avec une accélération de la détection de ces groupes criminels estimée à 25 fois. Cela contraste avec la plupart des travaux antérieurs réalisés dans ce domaine qui reposent soit sur la mise en correspondance de mots-clés connus liés au trafic, soit sur la formation d'un classificateur linguistique, ces deux éléments dépendant fortement de connaissances antérieures. Nous éliminons cette dépendance de l'obtention de données étiquetées par des experts, car il s'agit d'un processus long et coûteux. Nous présentons les résultats sur deux ensembles de données réelles afin de valider notre méthode. En utilisant un ensemble de données préalablement étiquetées, nous montrons que les meilleurs résultats produit par la méthode sont très probablement des cas de trafic. En outre, TrafficLight est également capable de trouver des **nouveaux cas** qui ne peuvent être trouvés qu'en examinant les liens entre les annonces, mais passent inaperçus si on les examine individuellement. Lorsqu'ils sont appliqués à un ensemble de données en direct, les résultats de TrafficLight sont corrélés avec la faible vérité-terrain disponible. Avec un *meta-clustering* additionel, c'est-à-dire le regroupement de groupes d'annonces, nous avons détecté plusieurs M.O.s (modus operandi = types de comportements) - le plus frappant était le comportement "écran de fumée" : plusieurs groupes d'annonces, certaines ayant des numéros de téléphone différents, mais non opérationnels, visant probablement à frustrer les enquêteurs avec des pistes sans issue.

I dedicate this thesis to my family and friends for their

unconditional love and support.

I love you all dearly.

Acknowledgments

I would like to express my sincerest appreciation to my supervisor, Prof Reihaneh Rabbany for allowing me to be a part of the Complex Data lab and giving me the opportunity to work on impacting problems. I would like to thank her for her valuable guidance, utmost compassion and encouragement throughout the thesis journey. I would also like to thank the entire team that provided me support and were vital in creating a conducive work environment. The counsel provided by Prof Christos Faloutsos in the project and collaboration with his team - Catalina Vajiac, Jeremy Lee, Namyong Park helped me become a better researcher.

I would also like to offer a huge shout out to Sacha Levy and Yifei Li who put tremendous efforts for setting up a data pipeline for this project. The project would not have been possible without you. I would also like to thank Junhao Wang, the constant lab partner in whom I could find a collaborator, friend and motivator; and Kelin Pelrine for helping me out whenever stuck.

Last but not the least, I would like to make a special mention for my family who have always motivated me to aim high, and my friends who became my Montreal family and helped me remain sane during the gloomy winter days. Thank you for always being there through thick and thin.

Contents

List of Tables

List of Figures

List of Acronyms

ILO	International Labour Organization
SESTA	Stop Enabling Sex Trafficking
RCMP	Royal Canadian Mounted Police
TVPA	Trafficking Victims Protection Act
UCA	U.S., Canada and Australia
LCan100K	Canadian Live Dataset
HT	Human Trafficking
SVD	Singular Value Decomposition
LDA	Latent Dirichlet Allocation
t-SNE	t-distritbuted Stochastic Neighbor Embedding
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
M.O. Detection	Modus Operandi Detection
ARI	Adjusted Rand Score

Chapter 1

Introduction

1.1 Problem Definition

1.1.1 Scale of the Human Trafficking Problem

Human trafficking in all forms is pernicious to society and difficult to tackle. Technological advances aid traffickers by increasing their ability to exploit more victims transcending geographical boundaries. The International Labour Organization estimates that there are 24.9 million people trapped in forced labour, 55% of which are girls and women accounting for 99% of victims in the commercial sex industry [1]. Of the more than 23,500 endangered runaways reported to the National Center or Missing & Exploited Children in 2019, one in six were likely victims of child sex trafficking [2]. A majority of victims are advertised *online*, as suggested in [3]. In 2013, revenue from U.S. online prostitution advertising totaled $45 million, 82.3% of which was generated by Backpage.com, a classified escort advertisement website [4]. Over the years, the online presence of trafficking industry has grown to become a billion dollar industry [1].

Even though significant advances have been made in curbing human trafficking, the trafficking industry evolves to counteract these advances. On March 23, Craigslist voluntarily shut down its Personals section in response to the passage of SESTA (Stop Enabling Sex Trafficking Act) [5]. SESTA-FOSTA makes it illegal for websites to knowingly host or support sex trafficking activity and make it possible for victims sold on these websites to pursue legal action. On April 6, 2018, Backpage.com and its affiliated websites were seized by the FBI due to the arrest of a trafficking group using Backpage as its main medium [6]. This move aimed at eliminating online advertisements of escorts failed as new websites quickly spawned up to fill the gap [7]. Our analysis also confirms this, as shown in Figure 4.2, we observe a jump in the traffic of alternative websites crawled about five months after the Backpage shutdown. With the spread of traffic to smaller multiple websites now, it is much harder to find sex offenders as law enforcement has to go through multiple sources of data. It is then supplemented by the difficulty in labeling a potential act as trafficking or not.

1.1.2 Trafficking in Canada

Human trafficking is defined as "the recruitment, transportation, transfer, harbouring or receipt of persons, by means of the threat or use of force of other forms of coercion, of abduction, of fraud, of deception, of the abuse of power or of a position of vulnerability or of the giving or receiving of payments or benefits to achieve the consent of a person having control over another person, for the purpose of exploitation" [1]. Based on the definition, Canadian laws prohibit any kind of human trafficking, regardless of whether the victim is a Canadian citizen or an immigrant. According to the report published by [8], the extent of human trafficking is still unknown due to its clandestine nature. However, it is believed this industry has seen a rise in the past few decades due to the technological advancements that allow criminal networks and individuals to recruit and advertise victims remotely, particularly underage girls.

The latest report by Juristat on Trafficking in Canada indicates that between 2009 and 2019, 97% of human trafficking victims were female, majority of them under the age of 25 [9]. Cases till date suggest that human trafficking is more prevalent in Nova Scotia (with 2.1 victims for every 100,000 people) and Ontario (with 1.14 victims per 100,000 people) [10]. The Canadian government in collaboration with RCMP (Royal Canadian Mounted Police) as a part of its latest initiative to fight human trafficking launched a campaign "National Strategy To Combat Human Trafficking 2019-2024", with one of the focus areas being the need for technological advancements and research [11].

The RCMP also points towards the organized nature of trafficking. They found that most trafficking cases are linked to escort agencies or brothels and are extremely difficult to detect[8]. Many human trafficking suspects are linked to other organized criminal activities like credit card fraud, mortgage fraud and organized prostitution etc. The migration of victims have most often been linked to organized criminal networks. For example, Eastern European links have been involved with the migration of women from Soviet Union on account of employment in massage parlours and escort services in the Greater Toronto and Montreal area [8]. RCMP also reports evidence of Canadian women from Montreal and Niagara being transported to the United States for prostitution, indicating cross-country operations. In another report by RCMP on signals to identify trafficking, they recognize four major kinds of trafficking groups in Canada[12]:

- transnational organized crime groups
- smaller criminal groups that have decentralized operations
- small family criminal groups that control the end-to-end operation
- individuals working independently for personal gains

Looking at the Canadian market gives an insight into the infestation of problem in Canada as well as its organized nature. One of our primary focus is to gather and study Canadian escort data since too little is known about it. In the next few sections, we briefly explain the underlying concept of this thesis that exploits the organized/synchronized nature of trafficking. This is followed by laying out the contributions of this study and defining its scope.

1.1.3 Trafficking is an Organized Activity

Past studies have made evidential claims that human trafficking is an organized crime with an average pimp having control over 4 to 6 victims [4]. The organization and size of the operations vary: certain traffickers operate individually or at a small-scale and on ad hoc basis, some are organized in loose 'networks' operating autonomously, others belong to large and well-organized criminal operations with transnational links. In 2000, the Trafficking Victims Protection Act (TVPA) was passed in the United States. It highlighted the presence of organized crime related to human trafficking, stating: "Trafficking in persons is increasingly perpetrated by organized, sophisticated criminal enterprises. Such trafficking is the fastest growing source of profits for organized criminal enterprises worldwide". [13]

In an example convicted case in British Columbia [14], the trafficker solicited to illegally push 9 underage girls and 2 women into the trafficking industry. He took photographs, "set the rates to be charged and placed advertisements on various escort websites". Victims had different levels of involvement at different times up to having "no input into the wording used in the advertisements". The same finding was backed by [3] stating: "Eighteen percent of victims report that only they posted their own ads online, 56 percent of victims say only the controller posted the ads and 17 percent report that both they and their controller posted the ads." **Monitoring these online ads therefore is the key to combating human trafficking.** In a recent example, intelligence from the online marketing footprint aided in detection and takedown of an international sex trafficking enterprise operating in U.S., Canada, and Australia (UCA Convict) [15]. These ads exhibited significant similarities in the content of posting, with the common grammatical errors. The ads related to this UCA Convict activity are also present in one of our datasets, and TrafficLight detects them as highly suspicious organized activity (ranked 3rd).

This study will leverage the organized behavior of trafficking to spot these ads among a plethora of online escort ads. We present an algorithm that looks for interconnections between ads, detects dense clusters and highlights the common evidences. We also provide a follow-up characterization of these clusters to identify the different types of trafficking activities. No such analysis exists currently and can be helpful to law enforcement to filter out groups of malicious (spams, scams) as well as legal clusters (massage parlours, individual escorts etc.)

1.2 Thesis Organization

This thesis is organised as follows:

- **Chapter 2: Background and Literature Review**

  This chapter provides the background needed to understand the work presented in subsequent chapters. We start with giving an insight on how investigators currently spot human trafficking cases without the intervention of technology.

  This is followed up by a deep dive into existing research related to human trafficking domain and the problems they address. We then present an overview of existing machine learning models which bear resemblance to the work presented in this paper.

- **Chapter 3: TrafficLight**

  This chapter formally presents the research statement addressed in this paper. We present a detailed view of our proposed solution TrafficLight which looks at dense interconnections between ads to spot trafficking clusters. All the different modules of TrafficLight are discussed in detail as well.

This section also includes a discussion on the datasets used for this task - Trafficking10K and LCan100K. We highlight the issues related to datasets in this domain, namely the absence of ground truth since labeling such ads is a time consuming process and label inconsistency due to the difficulty of marking ads as trafficking.

- **Chapter 4: Findings and Observations**

  In this section, we present the analysis on the two formerly defined datasets using our method TrafficLight. We show results on Traffickig-10K dataset using the expert annotated labels to highlight label inconsistencies. We show three types of results on this dataset - Corroboration, Scooping and New Attack Discoveries.

  On the LCan100K dataset, we present some case studies of groups operating in Canada. Our analysis on the Canadian dataset LCan100K shows that there are different types of activities prevailing in the escorting domain. From individuals to local groups with only a handful of individuals to national and transnational chains with functions in many different cities and countries. Such a study is highly beneficial to the law enforcement since first, it helps them understand the nature of trafficking and, second, it becomes easier to filter out malicious groups, thus saving the valuable investigation time.

  In the final section of this chapter, we present a study of the different kinds of groups and indications on how to spot spams and scams.

- **Chapter 5: Conclusion and Future Work** This chapter presents the conclusion on the research presented in this paper and our contributions in doing so. We also state the open problems and challenges that still need to be overcome to reach a viable long term solution for the law enforcement.

1.3 Scope of Thesis

This thesis solves three problems - find cases of organized activities in online escort advertisements, rank these groups in the order of suspiciousness and identify different types of modus operandi among the detected groups. We start by defining what human trafficking means and their link to organized crime groups. Detailed knowledge about the prevalence of the issue is provided, followed up by the current research work in this area. We discuss why there is an urgent need for unsupervised methods for detection of human trafficking cases due to various issues related to datasets. We then propose a solution TrafficLight comprising of 3 modules - *TLDetect*, *TLPresent* and MO Detection.

We test our solution on two real world datasets - Trafficking-10K and LCAN100K. Through our analysis we are able to show the inconsistencies in the labels provided by experts, highlighting their limitations on providing accurate labels. We also discuss some case studies observed in the Canadian dataset and provide a meta-analysis on the same.

1.4 Contributions

This paper presents TrafficLight, which combines and automates the first two steps and further ranks and summarizes the cases discovered to make the overall process more efficient.

The main contributions of this paper are three-fold:

- **Novel Lead Generation**: TrafficLight is an *unsupervised* solution which captures inherent patterns. Removing the *label dependency* makes the algorithm less costly as labelling is an inherently difficult and error prone process. Looking at connection between ads helps us do *group level detection*, which when looked at individually might have been missed.

- **MO Detection and Meta clusters**: Not only can we find and group suspicious ads, we can further group the groups ('meta clusters'), to find M.O.'s ('modus operandi' = behaviors). As we describe later several criminals (in many parts of Canada), follow the same M.O. of posting ads with non-functioning phone numbers (presumably, to confuse and flood law enforcement).

- **Productivity Boost**: TrafficLight provides *summarized results* which are sets/groups of ads that are related to a single case, with their connections highlighted within each case, and cases ranked against each other based on their suspiciousness. This achieves higher *interpretability*, while also making the process of detecting trafficking groups *efficient* by combining lead generation and case building into a single tool.

Chapter 2

Background and Literature Review

In this chapter, we discuss in detail the background work and past research related to this domain. We being by introducing how currently the process of investigation is carried out. This helps us better understand the requirements from an investigator's viewpoint. Next, we look at all the different research being carried out by the AI community related to identification of human trafficking. Even though this is a problem that has existed for centuries, its online presence is new. This shift in the nature of business gives the researchers access to a large amount of data that was not available before. The recent shift also means that the research on this subject is quite new too, and has a lot of open problems, some of which we will discuss in the next section. In the last section, we discuss all the past works related to the concepts used in this paper. It helps us gain better understanding of the available tools as well as our competitor baselines.

2.1 Current Practice by Investigators

The investigation of human trafficking cases in online escort markets comprises of three main steps, the first two of which our proposed TrafficLight focuses on, to automate and accelerate:

- *Step 1: Lead Generation*: find a suspicious ad by searching for red-flags such as using keywords indicative of underage activity, e.g. 'amber-alert', 'Lolita' [1], or following reported tips (e.g. phone number reported to a hot line).
- *Step 2: Case Building*: It takes more than just identification to convince law enforcement agencies to put in time for investigation of a case. In this step, the investigator collects related information and finds connected ads by searching through the evidence to identify whether the initial lead is part of an organized activity. Case building is a time intensive process which if done manually harbors the risk of missing out on evidences and is very time consuming.
- *Step 3: Verification & Target Identification*: If there is sufficient evidence to hint at organized crime, then secure law enforcement resources to investigate further into the case and eventually identify the person of interest.

These steps are currently performed manually for the most part, which requires considerable human effort and is extremely time-consuming. Fully solving a case, from the lead to the prosecution, might take several years. For example, one of the detected groups (discussed in Section 4.3.1) that resulted in shutdown of Backpage.com in 2018 was identified in 2015 but the final takedown happened in 2019.

[1] https://www.justice.gov/file/1050276/download (see item #50)

More often than not these cases go undiscovered due to the huge amount of time it takes to even spot a single case. Often, the preliminary investigation may not turn out to be fruitful due to many reasons: shortage of evidence, ads being hoax, fear among the victims etc. This further demotivates the law enforcement to spend time on detecting such cases. Given the law enforcement limited resources, the number of cases that can be investigated is also limited.

2.2 Analyzing Escort Ads

We review earlier human-trafficking detection efforts, grouped according to the methods they use. However, none of these methods match all the features of the proposed TrafficLight, as shown in Table 2.1.

2.2.1 Applied Software Solutions

. A few software solutions are currently available to help law enforcement tackle HT by monitoring online escort advertisements. Spotlight [16] is a victim identification tool that helps spot cases of child trafficking. Spotlight helps gather insights about a child victim by efficiently visualizing its evidences and attributes. This helps speed up the investigation and has helped law enforcement solve multiple cases. Marinus [17] is a company formed out of Carnegie Mellon Robotics lab that employ AI techniques to aid in human trafficking detection. They partner closely with law enforcement and private firms and reports their findings to them for further investigation. Their tool, TrafficJam, looks for identifiers like phone numbers, images to discover a trail of organized activities given a seed. These solutions show proofs of concept for collaborations with authorities, by offering interactive browsing interfaces and thus, enabling faster investigations.

2.2.2 Exploratory Analysis

. Most of the past research in human trafficking domain has been on discovering the human trafficking patterns or classification techniques to mark ads as trafficking. One of the first studies that recommends leveraging publicly available data to discern patterns of human-trafficking (HT) [18] presents a text engineering framework to look for ads which contain suspicious keywords. They introduce 3 approaches to classify ad as HT or not. The first two involves expert guidance in the form of commonly used keywords for trafficking and doing a keyword search or regex based search to mark ads as HT respectively. The third approach involves training a NLP model using ads with phone numbers known to be involved in suspicious activity as the positive labels. Their results show that NLP model achieves the highest AUC score, signifying that only expert identified keywords are not sufficient indicators.

They use the identified patterns to detect trafficking ads around the Super Bowl. The data shows increase in the count of ads posted daily in the 7-day period around the Super Bowl, suggesting high influx of out-of-town escort businesses. This study is later extended to find how local public events impact sex-worker advertising [19]. It looks at the behavior of escorting around a public event comparable in effect to Super bowl. Their analysis shows that contrary belief, some events do not contribute to increase in escorting ads, while others do. Their analysis also helped them discover other such public events that have an impact on the amount of activity which goes unnoticed. They provide a data driven approach to study this data in order to form better policies and resource allocation strategies.

2.2.3 Information Retrieval and Search

Given the informal nature of ads posted on escort websites, it is not a straight forward task to extract attributes like age, phone number, email address, location etc. These attributes come into play when exploring the relations between different ads. Many traffickers obfuscate these attributes making it harder for a script to automatically detect them, and hence remain under the radar. Many ads use domain appropriate slang words and symbols, for example using ‘roses’ as a symbol for hourly rates, using a mix of symbols and text for phone numbers etc. Hence, information extraction is a popular research area coupled with efficient storage and search to accomodate for the scale of the data.

The work in [20] describes an entity-centric knowledge graph approach to build a semantic search engine to assist analysts and investigative experts in the HT domain. The first component is the offline knowledge graph construction which takes a crawled web corpus as input and structures it into a semi-structured knowledge graph that is stored and indexed in a NoSQL fashion. The second component implements real time entity centric information retrieval by supporting searches on attributes like phone numbers and latent attributes like physical addresses. Another paper by the same authors [21] focuses on Information Extraction given an initial labeled set per attribute.

A closely related work [22] constructs a knowledge graph to provide a easy-to-use interface for entity based searches. Their tool called DIG uses Minhash/LSH algorithms to compute text similarity of ad content. They also use DeepSentiBank, a deep convolutional neural network to compute a hashcode of the images of the ad. Given an image, keyword or entity as a seed, DIG can successfully pull out the related ads from the knowledge base.

Another related work by the same group [23] looks at minimizing investigative efforts by creating a weighted network where the nodes are entity IDs and two nodes are connected iff they share a phone number or email address. They use a random walk based clustering to then discover latent cluster entities given a seed phone number. Similarly, [24] helps with search and case building by looking through shared attributes to search for related ads. It uses concepts from Active Search to inculcate expert feedback to automatically find the most important modalities for related trafficking ads.

A specific attribute that is extremely important to investigators is the geotag implicitly conveyed by the webpage. Escorts intentionally embed the location information in the content of the ad, while using a dummy location in the specific field to make it harder to obtain them automatically. Therefore, [25] presents a geotagging framework that integrates the strengths of semantic lexicons, extraction context, relational constraints and prior knowledge like Geonames dictionary to determine the probability of a candidate phrase being a geotag.

In an earlier study [26] present a tool, TrafficBot, that employs information retrieval, information integration and natural language technologies to build a data warehouse. They focus on two types of trafficking - first, transnational- people brought into the USA from other countries and secondly - domestic trafficking i.e. people within the USA. This paper also sheds some light on some features that could potentially help in identifying trafficking based on their study of transnational and domestic ads.

2.2.4 Advertisement Level Classifiers

Many previous works in this domain focus on classifiers to mark an ad as human trafficking or not. In particular, [27] presents a lightweight system combining expert's comments and semi-supervision coupled with state-of-the -art embedding methods. Their model requires a set of positively labeled ads to begin with. The model learns features based on these positive ads to assign a probability of being trafficking to each ad. At each iteration, few ads at the extremes of the probability distribution are picked and passed to the experts for annotation, thus increasing the set of labeled ads and retraining their model.

In an earlier work, [28] formulates the same task as an anomaly detection problem using hand engineered features. They look for features such as usage of third person language, words and phrases of interest like “sweet, candy fresh”, specific countries or ethnicities, advertisement of multiple victims, victim's age or weight. After identifying some initial set of positive and negative ads, they use Label Propogation and Label Spreading to mark an ad as trafficking. Their follow up work in [29] looks at the problem as an anomaly detection problem but uses a modification of SVMs to do so. They introduce an algorithm Laplacian SVM with regularization and pick ads that are farthest from the dividing plane as the anomalies.

The work in [30] provides a comprehensive study on features important for finding trafficking ads. Unlike other works, they also look at the likely bias caused due to using only the few expert annotations for model training. For example, the positive labels for groups of escort ads may come from a small number of law enforcement contacts that only provide cases for specific regions, whereas the negative labels sampled will follow the true location distribution. Also, experts usually use keywords to identify ads, which limit the types of trafficking activity a model will be able to pick up. They describe the trafficking detection problem in great detail and present an architectural overview of the the solution, followed by a bias mitigation plan.

The two most recent classification papers are deep learning based [31] [32]. Both train a neural network on the set of labelled ads, and does a classification task on the set of escort ads to mark ads as trafficking or not. [31] provides a deep multimodal model using language and vision modalities without using any explicitly defined expert keywords. This is also the first and only work to present the first rigorously annotated dataset for detection of human trafficking, called Trafficking-10K. This dataset includes 10,000 escort ads, with each ad scored from 0 to 6 on a spectrum of how likely trafficking they are. We provide extensive details on this dataset in Chapter 4 and use it as one of the primary datasets for analysis. [32] proposes an ordinal regression model with three components: a pre-trained Skip-gram model, a gated-feedback recurrent neural network and a multi-labeled logistic regression. They use a set of 168,337 ads from Backpage to train the the Skip-Gram model with negative sampling. The second component uses sequential modelling to get a sentence mapping from the word vectors in the skip-gram model. The third component is an ordinal regression module trained on Trafficking10K that ranks escort ads in the order of magnitude of trafficking likelihood. Their experiments show that their model outperforms [31] in the classification task.

2.2.5 Inferring Connections Between Ads

Though most of the work in this domain has been done as a supervised task to mark each ad as HT or not, there are a few works that model the problem differently. In many countries, individual escorting is a legal activity but pimping or forcing/managing another person is deemed illegal. One of the ways to identify such ads is to look at the similarity between ads for authorship attribution. Traffickers usually use similar templates to advertise multiple victims, or the attributes like phone numbers, email address or location may be the same.

A recent work, [33] finds tightly connected subgroups of nodes that have similar node-specific time series as well. They frame the problem as coupled-clustering over: 1) phone number co-occurrence network, 2) a set of phone number specific time series that capture ad content similarity (eg. per day average content similarity between ads using the same phone number).

An earlier work, [34] detects common authorship but is limited to two ads. Given two ads, it ascertains their probability of having been generated from the same source using patterns in the content. Similarly, [35] aims to find ads generating from the same source looking at the bitcoin transactions. It makes use of one of the only publicly available bitcoin transaction dataset. Their technique looks at synchronicity between bitcoin transactions and posting time of the ads. For a trafficker posting in multiple locations, the timestamp at which the ad appears across multiple locations is approximately the timestamp at which the transaction is propagated to the Bitcoin network.

The most recent and closely related work to ours, [36] uses an unsupervised text matching technique to find similar ads. Given an interconnected graph, they use HDBSCAN to find densely connected groups of ads. They address scalability by using a carefully designed hashing function to divide ads into buckets. The idea behind template matching is that if there is a high overlap in template signatures, there is an indication that the two templates might belong to the same organization.

2.3 General Related Works

2.3.1 Text Embedding

One of the main premise of this work is to look for similarity of ads based on strikingly similar phrases used in their content. To capture these, we look at some of the word and document embedding methods that can be applied. These embedding methods can be classified into two categories - generic factorization based embedding techniques applied on textual data (e.g. TFIDF representations) and specialized NLP based techniques.

Factorization Based Embedding

In this section, we cover the most commonly used factorization based embedding methods - SVD [37], NMF [38], PCA [39] and TSNE [40]. All these methods obtain a low rank approximation for the feature matrix, but use different methods to do so. While SVD and PCA obtain an embedding by extracting top k eigen vectors to represent each point as a combination of bases vectors, NMF calculates a non negative representation by optimizing the loss of information using a training dataset. NMF is suited well to situations where a negative value is hard to explain (eg. medical domain where the affect of a negative symptom is hard to comprehend). But NMF requires a training dataset and is not able to segregate the features into independent components. Thus, explaining results based on their correlation to the features is hard. NMF finds two non-negative matrices U and V, the product of which returns an approximation of the original matrix X. Depending on the initialization and optimization process and sampling of training data, the final embeddings may vary on different iterations.

SVD, on the other hand is a low-rank approximation method that uses singular value decomposition to get the most important components. Using spectral decomposition, a matrix $X \in \mathbb{R}^{n \times m}$ can be defined as the product of three matrices: U, V and Σ.

$$X = U \Sigma V^T \tag{2.1}$$

where U and V are orthogonal matrices, that is,

$$UU^T = U^TU = I_{n\times n}$$

$$VV^T = V^TV = I_{m\times m}$$

and Σ is a diagonal matrix. The column entries in U are the row singular vectors arranged hierarchically in decreasing order of importance in terms of their ability to explain the variance in the columns of X. Similarly, the columns in V are column singular vectors again arranged in decreasing order of importance. Thus, U and V are known as the left singular and right singular matrices respectively. The entries in Σ are the singular values of X, which are non-negative and in decreasing order of magnitude, that is,

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_m \geq 0 \tag{2.2}$$

This essentially means that the first columns of U and V are more important than the second columns, which in turn are more important than the third columns and so on and so forth in capturing the information in matrix X. The relative importance of the values of U and values of V are given by the corresponding singular value. To obtain a low rank approximation to k dimensions, the top k singular values are retained along with the corresponding k column singular vectors in V. This captures the top k most informative components that capture the variance of features in X. Removing the last (n-k) means removing the relatively unimportant information to approximate the matrix X.

Where SVD removes the redundant information to return sparse matrix using full rank decomposition on X, Truncated SVD computes a factorization where the number of columns are equal to the specified truncation. Only the top k singular vectors are used along with the top k row and column singular vectors. Truncated SVD is faster than SVD since it does not require computing all components but only the top k. The approximation of X is the closes rank k approximation of X and is formulated as:

$$X \approx U_k \Sigma_k {V_k}^T \tag{2.3}$$

SVD is the basis for many regression problems as well as PCA (Principal Component Analysis). In PCA, eigen decomposition is performed on the covariance matrix of the original feature matrix. The covariance matrix contains the covariance values for each pair of features. This is done to capture the set of features that are highly correlated and thus, contain redundant information. This is similar to applying SVD decomposition on the original data matrix. This step is preceded by centering the data, i.e. columns means have been subtracted from each column.

TSNE is a more recent embedding technique that is majorly designed for exploration and data visualization. While the goal for PCA and SVD is to preserve variance, the goal for TSNE is to preserve the distances of neighboring points. TSNE is a non-linearity based dimensionality reduction and is good for visualization especially for data with manifolds. However, TSNE embeddings are not suited for regression or classification tasks since it distorts distances to be able to give a better visualization.

NLP Based Embedding

With the availability of data and GPU for faster training of deep models, NLP has made a lot of advances in the recent past. We look at the various state-of-the-art word and sentence level embedding methods.

Word Embeddings The most popular models for word embeddings are the Word2Vec[41], FastText[42] and Golve[43]. These models have different ways of looking at word embeddings. Word2Vec was the very first model that revolutionized the way to get word embeddings. It laid out two approaches - continuous bag of words approach that tries to predict a word given its context, and the skip-gram model which instead uses the target word to predict its context. Word2Vec places words with the same context close to each other.

Word2Vec[41] produced better results than any competing method, but had some drawbacks - the context for each word depends on the window size, does not capture the word co-occurrence probabilities, cannot handle Out Of Vocabulary words. These issues were solved by the future models - Glove and FastText. Glove stresses the need to use global frequency of occurrences of words instead of just within the context window. Glove uses dimensionality reduction coupled with optimizing the embeddings such that the dot product of two word vectors equals the logarithm of the number of times they occur in each other's neighbourhood.

FastText[42] is an extension of Word2Vec released by Facebook in 2016. Instead of using the exact word for training the skip-gram or CBOW[44] model, its breaks down the word into a ngram of characters. For example, the word 'characteristic' will be interpreted as ‘cha’, ‘har’, ‘ara’, ‘rac’ and so on. The same model as proposed by Word2Vec[41] is then trained using these tokens. The word embedding of a word is then a combination of these lower level embeddings. With Fastext, generalization to unseen words is possible since the subcharacters may have been seen before and it requires lesser training data than other models.

However, one issue that none of the above word embedding models are able to solve was the disambiguation. A word with the same spelling but different context would not be able to be distinguished. For example, these models do not understand the different between the ‘bank’ of river, and a ‘bank’ meaning financial institution. This problem was first solved by Elmo [45] that provides the contextualized word vector representation of a word. It uses a bi-directional LSTM whose input is a sequence of words within a window in the sentence. In contrast to the above models, it is thus able to maintain the context better, rather than all words within the context window being treated the same.

Sentence Embeddings Sentence level embeddings prove vital in document classification tasks or capturing the similarity of documents. The word vectors obtained from the above models can also be used to obtain sentence level embeddings by various techniques (averaging, summing, non linear transformation, etc.). However they would not fully capture the semantics of the entire sentence. One of the first works towards resolving this was the Doc2Vec model [46] which is an extension of Word2Vec to paragraphs. It introduced a paragraph vector along with the word vectors in the training step. The main intuition is that training to optimize the context probability would capture the latent semantics of the sentences, reflected in the learned paragraph representation. The paragraph vector is shared across all contexts derived from a single paragraph but not across different paragraphs. Elmo [45] discussed in the previous section, though creates word vectors, was primarily designed for sentence embedding. The deep bi-directional architecture of Elmo helps create word vectors of tokens as well as the sentence vector simultaneously.

A major breakthrough in text embedding happened with the introduction of transformer based models. One of the models that garnered significant attention was the BERT [47]. This was followed by a number of different models with small tweaks to create different versions of BERT. BERT is a bidirectional transformer model. In its training, instead of context words, it uses two training strategies: a) masking 15% of the words in the corpus, and training model to predict them; b) predicting the next sentence in the sequence. BERT achieves exceptional performance but is computationally expensive. The original paper[47] trains various versions of BERT - BERT-Large with 24 layers, 1024 hidden units and 340M parameters. A relatively efficient version is the BERT-Base with 12 layers and 768 hidden units with 110M parameters. BERT-Tiny [48] is another such model which achieves comparable performance with significantly less parameters (4 layers, 312 hidden units and 14.5M parameters).

2.3.2 Finding Dense Regions

We look at two different line of research under this section. One uses euclidean embedding and clusters points based on their distance. Another line of work looks at this problem as dense subgraph identification where a graph is formed with datapoints as nodes, and edges are introduced based on the pairwise similarities or shared attributes.

Dense Block Detection

Dense block detection is a common approach to spot anomalous or suspicious nodes by looking at the connections in the graph [49–52]. For example, in a twitter network, where users buy fake followers or create bots, spotting a dense network of connections will help find malicious users. CopyCatch[49] is one such method that uses local clustering method to find spammers on Facebook. They look at the social graph between users and pages and the timestamp of edge between them corresponding to the time at which the user liked a page. They infer nodes with lockstep behaviour from this graph to find illegitimate page likes by spammers using co-clustering.

Similar to CopyCatch, CatchSync[50] looks at the global patterns to find suspicious nodes. It looks at two tell-tale signs: synchronized behaviour i.e. nodes showing similar pattern of activity, and rare behavior i.e. groups of nodes with strikingly different behavior from the rest of the graph. This picks on relatively suspicious nodes based on the relative density of data points. CrossSpot[53] is an extension to CatchSync that not only finds suspicious blocks in a graph, but also ranks them in order of their suspiciousness. They define a set of axioms that any metric of suspiciousness should satisfy and provides experimental results on one of the metrics from this class. [52] looks at the eigenspokes that is the SVD components that show show good separability and finds clusters by grouping together points at the end of the eigenspokes into a single cluster.

FRAUDAR [51] adapts the properties defined in the axioms put forth by [53] to extend to their setting where nodes and edges can be weighted. This is one of the few works that extends to weighted graph. They also consider the graph to be bipartite in nature with each edge signifying a follower relationship or product review, and poses the problem of finding fake reviews or fake followers in such a graph as dense block detection. FRAUDAR looks at the individual suspiciousness of nodes, aggregating them to find the suspiciousness of the group, making it biased towards large clusters.

Density Based Clustering

We look at three of the most widely used density based clustering algorithms - DBSCAN [54], OPTICS[55] and HDBSCAN[56]. DBSCAN is the most primitive of the three. It uses a specified threshold of distance to separate dense clusters from sparser noise. It uses a metric called ‘search distance’, which is the largest distance within which it looks for ‘minimum number of points’ for them to be recognized as a cluster. Due to a fixed search distance, it can happen that two dense clusters are merged together to form one large cluster. DBSCAN is the fastest of all methods, but gives good results only if there is a very clear value of search distance to be used.

OPTICS solves the issues in DBSCAN by also considering ‘core distance’ along with the ‘search distance’. Core distance is defined as the distance of a point to its k^{th} nearest point, where k is a hyper-parameter meaning minimum number of points per cluster. OPTICS treats search distance as the maximum distance within which points will be compared to points within the core distance. The distance between two points is known as the reachability distance. OPTICS creates a reachability plot using these distances, the valleys in this reachability plot form a cluster. It allows a lot of flexibility in fine tuning the clusters but is very computationally intensive.

HDBSCAN uses the least user input out of all three methods. It is also known as self-adjust clustering since it uses a range of distances to separate clusters of varying densities from sparser noise. It is an extension of DBSCAN that creates a hierarchy of clusters obtained using different thresholds of search distance. Flat clustering is then obtained by getting clusters that are the most stable i.e. the cluster does not break into two or more separate clusters each having more than 'minimum number of points' with decreasing search distance. HDBSCAN is the most data driven approach for density based clustering and is a good choice when expected density of points in unknown or clusters are expected to have varying densities.

2.4 Qualitative Comparison

In this section we lay out the 5 desired features required for a successful human trafficking detection tool, and provide a qualitative contrast of different methods (See table 2.1 for a quick overview). The first three methods in the table - Fraudar[51], CrossSpot[53] and SpokEn[52] are dense block detection techniques while HTDN[31] and Template Matching[36] are HT related most recent works.

- Label Independent : This feature is utmost important since obtaining labels is very difficult and requires the involvement of experts who have limited time to offer. Here, we present an unsupervised solution, a feature that is missing for instance in HTDN [31] or other classifier based solutions.

- Interpretable : The sensitivity of the nature of the applied area makes this a vital requirement. Any group that is passed onto law enforcement for investigation should be convincing and easy to explain. Unlike previous works, we put an emphasis on making the results interpretable by highlighting the connections between ads.

- Ranked Output : Around 100,000 ads are added daily to these online platforms. Due to the enormity of the data it is impossible for law enforcement to look at all the probable cases of trafficking. Hence, we introduce a ranking component to pick the topmost suspicious clusters. Though there are previous works that use a suspiciousness score to extract dense blocks [51], [53], [52], none of them have been applied to this domain before. Our experiments show that our suspiciousness metric outperforms the metrics defined in these works.

- Group Detection : Traffickers generally have control over more than one victim and operate as a group. Thus, looking at the similarities between ads to find ads originating from the same source can help us find trafficking. It also reduces the effort on case building and holds more investigative value than investigating a single lead. However, most of the past works perform analysis at ad level.

- Summarization : This step contributes to the efficiency of investigation. By summarizing the common connections between ads in a group, we make the

	Generic Methods			HT Methods		
	FRAUDAR[51]	*CrossSpot[53]*	*SpokeEn[52]*	*HTDN[31]*	*Template Matching[36]*	***TrafficLight***
Label Independent	✓	✓	✓		✓	✔
Interpretable	✓	✓	✓		✓	✔
Ranked Output	✓	✓	✓			✔
Group Detection	✓	✓			✓	✔
Summarization					✓	✔

Table 2.1: TrafficLight *addresses all the challenges:* competitors miss one or more.

process of sifting through ads and reaching a conclusion much faster. This feature is also missing in most of the previous works as can be seen in Table 2.1.

Chapter 3

Method

As discussed in Chapter 1, a large part of trafficking happens as an organized crime with the involvement of a single controlling person or organization, thus exhibiting lockstep behavior. More often than not, victims have limited control or knowledge over the content of their ads. Such ads coming from a single source tend to share strong connections, such as using the same phrases and misspellings, the same rates, and even using the same phone number to advertise different victims. Our proposed solution TrafficLight leverages this organized characteristic of posting to detect ads exhibiting **lockstep** behaviour, see Figure 3.1 for an abstract illustration.

TrafficLight draws on ideas from dense block and anomaly detection techniques to look for a suspiciously coherent set of ads. Given the limited time and resources of the downstream human investigator, TrafficLight uses a ranking measure to only get the ads most likely to be involved in trafficking so that the investigator's time is used judiciously. Moreover, TrafficLight reduces the time required to spend on each case by collecting and summarizing the evidence. It is

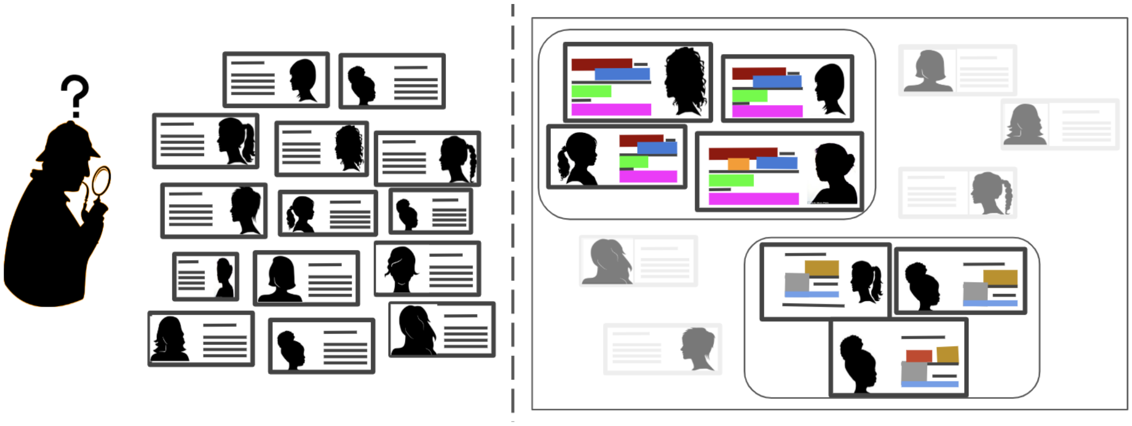
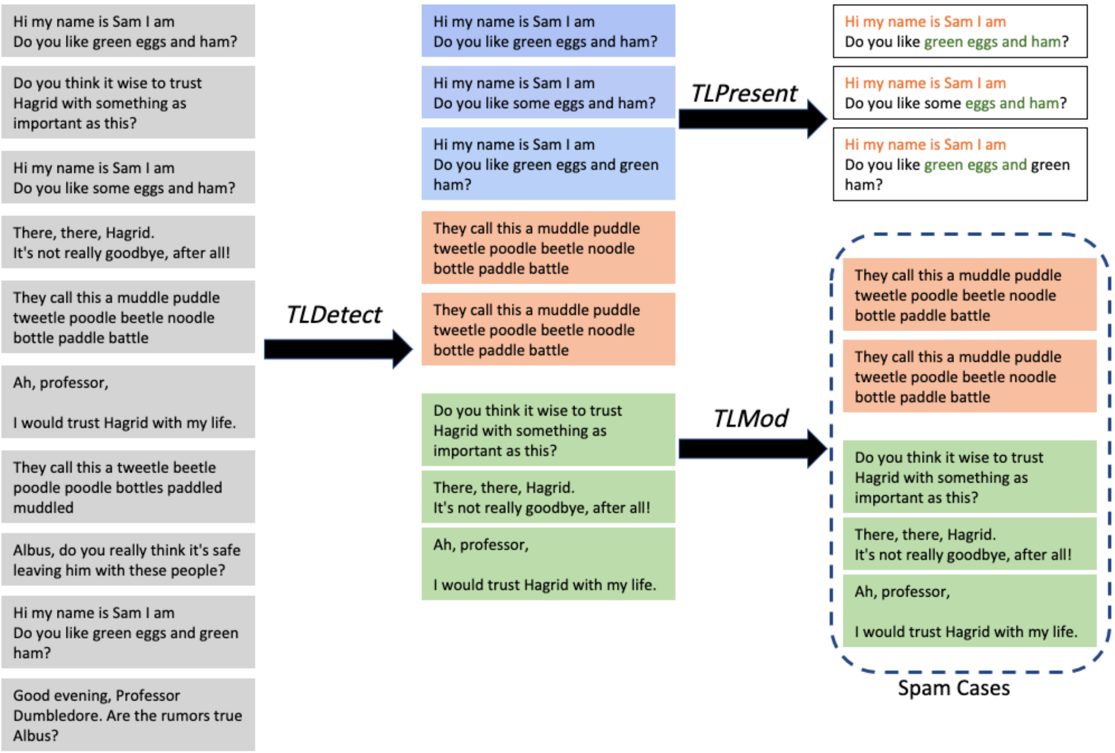


Figure 3.1: TRAFFICLIGHT *in action*: spots organized activities and highlights striking evidence for easier investigation.

imperative to state here that we do not intend to completely replace the manual investigation but provide a solution to make the job of investigators much easier. Our proposed solution does the hard job of finding needle in the haystack, i.e. find related groups of ads within hundreds of thousands of ads and provide the most relevant ones for further investigation.

This is followed by a study on the characteristic attributes of these clusters, which could be of vital importance but is unexplored at the moment. Due to the freedom of online posting and the curse/boon of anonymity, these websites have not only attracted traffickers but other type of organizations like massage parlours, escorting services etc. This has also lead to exploitation of these platforms by malicious agents like scammers and spammers. Unfortunately, there is very little knowledge on the different types of activities that the escorting websites are currently being used for. Through this paper, we wish to study some of these characteristics to differentiate between different organized groups to have a better insight into the domain. We



Hi my name is Sam I am
Do you like green eggs and ham?

Do you think it wise to trust Hagrid with something as important as this?

Hi my name is Sam I am
Do you like some eggs and ham?

There, there, Hagrid.
It's not really goodbye, after all!

They call this a muddle puddle tweetle poodle beetle noodle bottle paddle battle

Ah, professor,

I would trust Hagrid with my life.

They call this a tweetle beetle poodle poodle bottles paddled muddled

Albus, do you really think it's safe leaving him with these people?

Hi my name is Sam I am
Do you like green eggs and green ham?

Good evening, Professor Dumbledore. Are the rumors true Albus?

Figure 3.2: *Overview of the* TrafficLight *method: TLDetect* spots micro-clusters; *TLPresent* highlights within-cluster similarities; *TLMod* groups into meta-clusters, revealing potential M.O.s (modus operandi).

also look at the characteristics of spams and scams and provide a way to filter them out so as to save valuable investigation time.

In this chapter we provide the technical details of our solution TrafficLight and define in depth the various modules. We start by formally defining our problem in Section 3.1, and describe the modules of TrafficLight in the subsequent sections.

3.1 Problem Statement

Goal: Find organized activity with strikingly similar features.

Input: A set of n escort advertisements each represented as a text document.

Questions: (a) How to identify micro-clusters forming dense blocks and summarize the similarities between ads, for *lead generation and case building*?

(b) How to design a metric to rank discovered micro-clusters in order of highest potential of being trafficking, to achieve *Productivity Boost* by the ranked output?

(c) How to further group clusters into meta-clusters to identify different types of MO('Modus Operandi', i.e. behaviors)?

TrafficLight characterizes typical posting behaviors, spots anomalous & suspicious ones, and visualizes patterns within the collection to aid law enforcement.

We model our proposed solution as a combination of three modules:

- ***TLDetect*** - Finds micro-clusters of highly related, unlabeled advertisements using dense block detection.

- ***TLPresent*** - Picks the topmost suspicious ads for further investigation and visualizes common evidence to make it easier for an investigator to reach a conclusion.

- **TLMod** - Characterizes the detected clusters to find different types of activities.

In the next sections, we provide details on the individual modules.

3.2 Finding Organized Activities: *TLDetect*

Main Intuition. Traffickers usually have multiple victims and advertise them similarly. How can we find these similarities in advertisements? The first component *TLDetect* creates a feature space from the ads and looks for tiny dense sub-regions.

3.2.1 Embedding Ads

How do we generate a meaningful embedding for advertisements? Organized traffickers use templates with small variations (insertions, deletions, misspellings) to write advertisements, which can be identified by the prominence of noticeable features. The first step of *TLDetect* is feature reduction using Truncated SVD on the tf-idf weights of n-grams to capture the importance of a word in the document.

More precisely, let $A_{i,j}$ denote the tf-idf score of n-gram j in ad i; $A \in \mathbb{R}^{p \times q}$ represents the feature matrix extracted on the set of ads p, q being the total number of n-grams in the corpus. $A_c = SVD(A) \in \mathbb{R}^{p \times k}$ gives the embeddings of the ads in a k-dimensional space where $k << q$. The lengths of the n-grams and k are hyper-parameters of the model. In our experiments we use all 2-grams to 5-grams and $k = 20$. In Section 4.2.4, we provide an evaluation of our method on a synthetically generated dataset to show that TrafficLight works better compared to baselines. Our analysis showed that the results were insensitive to the choice of n and k (see Section 4.2.1 for justification of choices).

3.2.2 Detecting Micro-clusters

We identify dense regions in the embedded space to find ads that share a high percentage of features. We start by using an off-the-shelf clustering method[1] to detect candidate clusters on the latent representations A_c obtained in Section 3.2.1. HDBSCAN is suited to our purpose due to the following reasons:

- Finds dense regions alongwith marking noisy elements (data points that do not form a part of any dense cluster)
- It finds clusters of varying densities, by adjusting the density threshold
- Does not require hyper-parameter tuning, the only hyper-parameter is the minimum number of points in a cluster

Though we recommend using HDBSCAN, the pipeline is modular to fit any other density based clustering algorithm in its place.

Most clustering algorithms have a quadratic complexity and are difficult to scale to millions of data points. To make the clustering scalable, we first filter ads that do not form a distinguishable cluster. A lot of advertisements in the escort ads crawled from escort websites will be non-relevant ads posted by individuals. The embedding of such ads will not have a significant magnitude along any of the components. Thus, to refine the dataset, we filter out ads very close to the origin (0, 0) in the embedding space. We want to re-emphasize that through this work our focus is on discovering groups of organized activities with high level of suspicion. We are not interested in

[1] We recommend using HDBCSAN based on our experiments.

finding all the groups, hence it is safe to filter out the insignificant points in order to make the pipeline scalable.

Since escort agents tend to re-post ads to increase traffic, we also remove all duplicate ads with very high similarity. This is done using cosine similarity between the count vectors of the ads and de-duplicating ones with a high similarity score.

Trafficking clusters are usually tiny relative to the size of the data. Due to the dynamic density threshold used by HDBSCAN, sometimes the clusters obtained may not be perfect. That is, they might suffer from one of the following limitations: 1) some candidate clusters are a combination of two closely situated dense clusters 2) a group of ads may be divided into two or more clusters due to dense connections within some parts of the ads. Thus, we adopt a cluster refinement process to adapt it to this application by: 1) Dissolving clusters that are non-homogeneous, and 2) Merging clusters that share a high inter-cluster agreement score.

Dissolving non-homogeneous clusters. Occasionally, a cluster consists of two or more partially overlapping sub-clusters. To remedy this, we propose to measure the coherence of a group of ads, as the ratio of the first two eigen values of the ad-term matrix A_c: Given a set of clusters $C = \{c_1, c_2 \dots c_l\}$, the coherence for a cluster is defined by the first (λ_1) and second (λ_2) eigen values of the slice of the matrix A_c corresponding to the ads within the cluster.

$$coherence(c_i) = \lambda_1 / \lambda_0 \tag{3.1}$$

If the ratio is small, it means that the cluster is coherent (low rank); otherwise, we want to break it into sub-clusters. Thus, we mark all ads in clusters with

$coherence > \delta$ [2] as noise and re-cluster them, along with the ads marked as noise in the first round. This two-step process helps us detect tiny clusters, which are then added to the coherent clusters from the first step, before being passed through a final merging process (Algorithm 3).

Merging very similar clusters. How can we measure cluster similarity and determine if any pairs of clusters should be merged?

Since we detect micro-clusters, this step ensures that different micro-clusters having small dissimilarities are grouped together. We use a clique merging technique by constructing a graph with all the clusters as nodes. An edge is added between any two nodes where cosine similarity of the cluster centroid exceeds a threshold value. The cluster centroids are calculated by taking the average of the feature vectors of all ads belonging to a cluster. All the ads in the clusters forming a clique are then merged together to be in a single cluster. See Algorithm 3 for details.

In all our experiment we use $\epsilon = 0.1$, $\alpha = 0.8$, and $\beta = 0.8$, which are our recommendation for the default values of these parameters.

The pseudo-code for TLDetect is provided below. The function db_clustering is a shorthand for density based clustering, HDBSCAN in our implementation. It returns the cluster assignments and the ads not included in any cluster are marked as noise.

[2] we choose $\delta = 0.8$ since we only need to preserve the highly coherent clusters

```
Input: $P$ = set of ads

$A \in \mathbb{R}^{p \times q}$ = Tf-Idf matrix of ads × n-grams ;
$X$ = SVD($A$, n_components=k);
$O = [0]^k$;
$P_0 = \{p \in P$ if $euclidean(X[p], O) > \epsilon\}$ {# retain points not on the center};
$X_0 = X[P_0]$;
$C_1, N_1$ = db_clustering($X_0$) {# get clusters and noises};
$C_s, N_s$ = dissolve_nonhomogenous_clusters($C_1$);
$P_n = N_1 \cup N_s$;
$X_n = X_0[P_n]$;
$C_2, N_2$ = db_clustering($X_n$) {# re-cluster remaining noise};
$C = C_s \cup C_2$;
$C_m$ = merge_clusters(C, X)
```

Algorithm 1: *TLDetect*

```
**Input:** $C$ = set of candidate clusters

$N_s = \{\}$;
$C_s = \{\}$;
**for** $c$ *in* $C$ **do**
    $coherence = \lambda_1/\lambda_0$;
    **if** $coherence \geq \alpha$ **then**
        $N_s = N_s \cup c$ {# dissolve cluster into noise};
    **end**
    **else**
        $C_s \xleftarrow{+} c$ {# retain cluster as a result}
    **end**
**end**
return $C_s$, $N_s$
```

Algorithm 2: Dissolve Non-homogeneous Clusters

```
Input: $X_{n\times k}$ = Feature Matrix and $C = \{c_1, c_2, ..., c_m\}$ = Set of $m$ clusters

$M_{m\times k}$ = centroids(C) {# compute the center of each cluster};
$A_{m\times m} = 0^{m\times m}$ {# initialize the cluster adjacency matrix} ;
$sim_M$ = cosine_similarity($M$){# compute all pairwise centroids similarities}
$\forall_{(i,j)\in m}$ $A(i,j) = 1$ if $sim_M(i,j) \geq \beta$ {# connect similar clusters}
$Q$ = find_cliques(A)
$C \leftarrow \forall_{q\in Q}$ merge_clusters( $\{c \in q\}$)
 return $C$
```

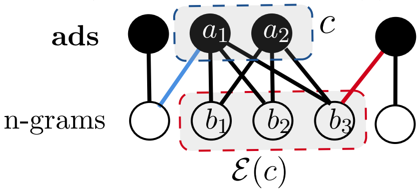
Algorithm 3: Merge Clusters

3.3 Presenting Detected Cases: *TLPresent*

How do we minimize the time investigators spend on finding and building a case? The goal of *TLPresent* is to present highly suspicious cases in an easily interpretable way. We first choose the micro-clusters that are most likely to be involved in organized trafficking, improving the quality of results shown to the human experts. Then, we visualize common features among ads in the micro-cluster to make evidence collection easier for law enforcement.

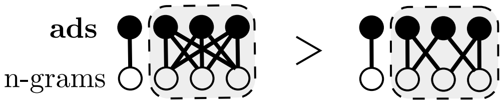
3.3.1 Ranking Suspicious Activities

How do we create a metric that, given a micro-cluster of ads, determines its suspiciousness? Let us first consider a sub-graph connecting ads in a given cluster $c = \{a_1, a_2, \dots\}$ to the n-grams used at least in two of those ads – shared pieces of evidence to their connection. We denote these shared n-grams by set $\mathcal{E}(c) = \{b_1, b_2, \dots\}$. More formally $\mathcal{E}(c) = \{b | \exists_{i,j} a_i \in c \wedge a_j \in c \wedge a_i \neq a_j \wedge b \in a_i \wedge b \in a_j\}$.

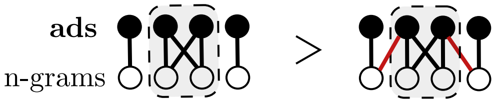


Given this evidence graph, we first define the following set of axioms which a proper metric of organized behavior should satisfy. Then we introduce *TLScore* which satisfies these axioms and can be used to rank micro-clusters based on their suspiciousness.

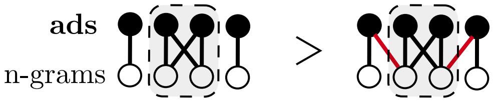
- *Coherence*: the cluster in which more ads have more of the shared terms should be ranked higher, e.g.



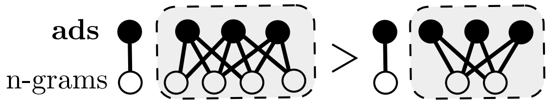
- *Exclusivity*: the cluster in which more of the terms used in the ads are only shared amongst them should be ranked higher, e.g.



- *Rarity*: the cluster in which more of the terms used in the ads are rarely used outside of the cluster should be ranked higher, e.g.



- *Support*: the cluster with the more terms shared between the ads in that cluster should rank higher, e.g.



To satisfy these axioms, TLScore first assigns weights to the edges using the smooth *tf-idf* transformer, to capture the rarity of evidences i.e.

$$w(a,b) = f(a,b) \times \frac{log(1+n)}{1+d(b)} + 1, \tag{3.2}$$

where $f(a,b)$ denotes how many times ad a has the n-gram b, $d(b)$ denotes how many ads in total have the n-gram b, and n is the total number of ads in the corpus. Given these weights, TLScore measures the ratio of the weights to the shared n-grams to account for exclusivity. More specifically:

$$TLScore(c) = \frac{\sum_{a\in c,\, b\in\mathcal{E}(c)} w(a,b)}{\sum_{a\in c,\, b\in a} w(a,b)} \times log(\frac{\sum_{b\in\mathcal{E}(c)} d(c,b)}{|\mathcal{E}(c)|}) \tag{3.3}$$

where $d(c,b)$ denotes how many ads in c have the n-gram b.

TLScore is the only measure that fully satisfies these axioms compared to alternative generic metrics for cluster quality in graphs as well as Fraudar[51] which is specific for detecting organized activity. This can be verified by the examples above and is summarized in Table 3.1, where ‘p’ means a partial pass only when we use the smooth tf-idf transformer, to capture the rarity, otherwise the original metric fails to satisfy that. Table 3.2 reports the difference in the measurements of the left hand side and right hand side examples in each axiom (should always be positive to satisfy the inequality). A good metric should have all the values as positive.

We use TLScore to select the top-most suspicious clusters to provide to law enforcement. We provide as many micro-clusters to law enforcement as they are willing to investigate.

Method Axioms	Density	Conductance	Modularity	Fraudar	**TLScore**
Coherence	✓			✓	✔
Exclusivity		✓	✓		✔
Rarity	p	✓	✓	p	✔
Support			p	✓	✔

Table 3.1: *TLScore satisfies all axioms:* other measures miss at least one of them.

Method Axioms	Density	Conductance	Modularity	Fraudar	**TLScore**
Coherence	0.21	0.0	-0.33	0.32	0.41
Exclusivity	0.0	0.38	0.12	0.0	0.23
Rarity	0.29	0.19	0.17	0.29	0.16
Support	0.0	-0.02	0.02	0.87	0.63

Table 3.2: *Positive is a must - only TLScore obeys all axioms:* each cell corresponds to the difference δ between the values of left hand side and right hand side examples in each axiom. The red highlights $\delta < 0$ i.e. where the axioms fail.

3.3.2 Explaining Detected Clusters

How do we determine the top features that describe a suspicious micro-cluster and visualize them in a useful way for law enforcement? The goal is to show ads with visual aids for evidence so that they stand out, and an investigator does not have to go through the entire content of an ad to make his/her decision. Using concepts from Human-Computer Interaction, we design an output using color coding and highlighting to make the evidence visually snappy.

We use topic modeling to get the phrases for describing a cluster. Long common phrases between ads are highly suspicious of organized activity, so we unify any phrases that overlap or appear close to each other to generate variable-length n-grams. We highlight similar phrases using the same color across all ads within the cluster. Any phrase which is not shared but is in the topmost phrases returned by topic modeling is also highlighted since we do not want to misguide the investigators by pushing down distinctive features among grouped ads.

Lemma 1 (Estimated Speedup by TrafficLight)**.** TrafficLight *speeds up the process of identifying a potential case of trafficking by at least* $25\times$*.*

Proof. The Psychology of Computer-Human Interaction [57] models the human mind as a combination of three systems - perceptual system(p), motor system(m) and cognitive system(c), each parameterized by a cycle time (τ), decay rate (δ) and a fixed memory (μ). We assume infinite memory for each system, and the parallel processing capacity of C chunks for the cognitive system. Without loss of generality, we assume each ad to consist of P phrases and a collection of 10,000 ads. The time to process all ads then can be calculated as:

- Time taken to process each phrase, $T_{ph} = \tau_p + \tau_c + \tau_m$
- Time taken to process N phrases, T_P= $P \times \tau_p + \frac{P}{C} \times \tau_c + P \times \tau_m$
- Time taken to process P phrases for 10,000 ads = $10000 \times T_P$

TLPresent further ranks and highlights the evidences, reducing the number of phrases per ad. We are also interested in only the topmost suspicious clusters, considerably reducing the final pool of ads and clusters to be looked at. Taking the k topmost suspicious clusters with an average size of z and l, the total time reduces to:

- $T_{TL} = \frac{l}{n} \times (k * z) \times T$

From our experiments, $z \approx 12, l \approx 0.5 \times n$. Assuming the investigator looks at the top k=100 cases, we get:

- $T/T_{TL} = 25$

We can see that TrafficLight speeds up the process of identifying a potential case of trafficking by *25x*. Furthermore, the time saved is proportional to the size of the dataset, with *minimum 10x increase in saved time corresponding to a 10x increase in the size of data.* □

Note that this analysis assumes infinite human memory storage, so our estimate is only the lower bound of the amount of time saved, which, in practice, would be much higher.

3.4 Characterizing Detected Cases: *TLMod*

In this section, we present a novel contribution on characterizing different types of activities to obtain meta-clusters. This analysis is not in place currently, automatically or manually. We believe that such analysis will help us and law enforcement better understand the nature of trafficking ads, while removing activities that obfuscate the online space, e.g. spam/scam activity. *TLMod* also provides an added advantage of enabling all the micro-clusters in the meta-cluster to be labeled together, e.g. as spam/trafficking/massage parlor chains, thus making the process more time efficient.

Given a micro-cluster, how can we identify various cluster level attributes that can be used to find different activity patterns? e.g. local v.s. national, lockstep v.s. organic. *TLMod* incorporates the temporal and spatial meta-data available with these ad postings, as well as the occurrences of hard-identifiers, i.e. email addresses and phone numbers which are further verified for validity using the Twilio platform [58]. More specifically, *TLMod* defines the features in Table 3.3 to characteristically distinguish micro-clusters. Given the features extracted for each micro-cluster, *TLMod* then applies singular value decomposition followed by KMeans to obtain k' meta-clusters [3]. In the experiments section, we discuss two significant types of meta-clusters/M.O.s discovered by this approach while explaining the difference between them in terms of their attributes.

[3] $k' = 8$ since it achieved the maximum silhouette score over the range (2, 10)

Feature	Description
Cluster Size	Number (#) of ads
Phone Count	# of unique phone numbers
Email Count	# of unique Email Ids
Location Count	# of distinct locations
Location Radii	Max dist. between any two locations
Date Entropy	Entropy of date of posting
Email Entropy	Entropy of occurrence of emails
Phone Entropy	Entropy of occurrence of phone numbers
Invalid Phone Ratio	Ratio of valid to total phone numbers
Posting Timeline	# of days a cluster is active, i.e. difference in the posting dates of the first and latest ad

Table 3.3: *MO Detection: TLMod* extracts features per each micro-cluster for higher level characterization of activity types.

Chapter 4

Findings and Observations

In this chapter, we focus on the experimental evaluations and results using TrafficLight. The analysis was done on two real world datasets - one with ads mostly from the USA that had been posted on Backpage.com and the other from different escort sites in Canada. On both the datasets, we provide quantitative and qualitative analysis using weak ground truth labels and case studies respectively. Through this section, we shall obtain an understanding of the labeling inconsistencies in the labels provided by the experts. This again highlights the need for unsupervised models that would not rely on manual labels. Due to the unavailability of ground truth labels for the exact clustering task, we provide two solution: 1) curate a synthetic dataset to provide quantitative results and justify design choices, 2) use hard-identifiers or classification labels as a proxy for the ground truth. We shall discuss this more in the next sections.

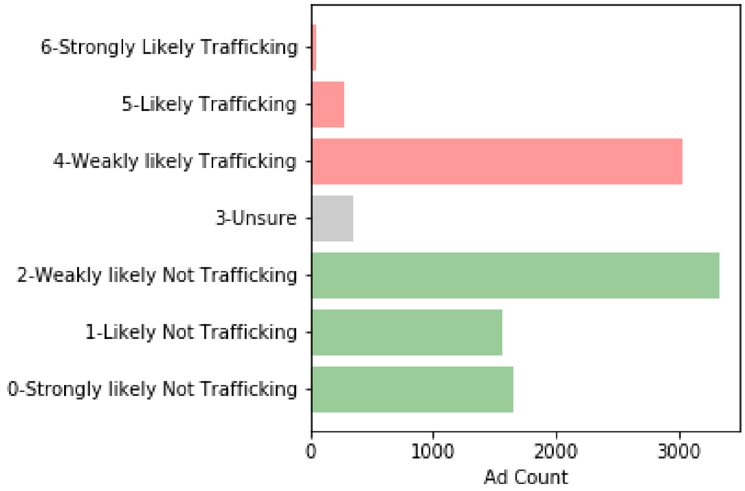


Figure 4.1: Distribution of Trafficking labels in Trafficking-10K dataset

4.1 Data Description

We will start by providing some details about the datasets we analysed followed by results on them.

4.1.1 Trafficking-10K

This dataset released by [31] contains 10,365 ads sampled from Backpage.com. Each ad has been annotated by expert investigators with inter-annotator agreement of 80% to get a label for likelihood of an ad being trafficking, *0 - Strongly Likely Not Trafficking, 1 - Likely Not Trafficking, 2- Weakly Likely Not Trafficking, 3 - Unsure, 4 - Weakly Likely Trafficking, 5 - Likely Trafficking, 6 - Strongly Likely Trafficking.* We use these labels to validate some of the clusters found by TrafficLight as a measure of the accuracy of our method. The distribution of the labels in the dataset can be seen in Figure 4.1

Trafficking-10k is the only labeled HT dataset available given the difficulty of labelling such data. There is a inter-annotator agreement of 80% reported for this data between the three expert annotators [31]. Our analysis suggests that only 40% of duplicate ads have the exact same label. We further discuss the labeling inconsistencies in our experiments and highlight how TrafficLight could be potentially used as a labelling facilitator since clustered, highly similar, ads should have similar labels. The data is also sampled, so many ads marked as trafficking might not necessarily be grouped into organized clusters due to connected HT ads missing from the sampled dataset.

4.1.2 Canadian Live Escort Ads

This is a collection of 100,000 unique live ads we crawled from two escort services websites (https://max80.com, https://escortalligator.com) posted across 49 different cities in Canada. All ads collected were published between January 2016 and September 2019. Figure 4.2 shows the spatial and temporal distribution of ads in this dataset.

The crawled websites are hubs for online escort services in Canada, serving more than one million ads over the past four years. A preliminary look into the volume and spread of escort advertisements across Canada motivates us to investigate this data further. The content collected for each ad includes the ad's url, title, description, posting date, as well as the escort's age, phone number and location. We treat phone numbers as hard identifiers, meaning that ads using the same phone number are related and hence should be clustered together. In total, there are 13825 unique

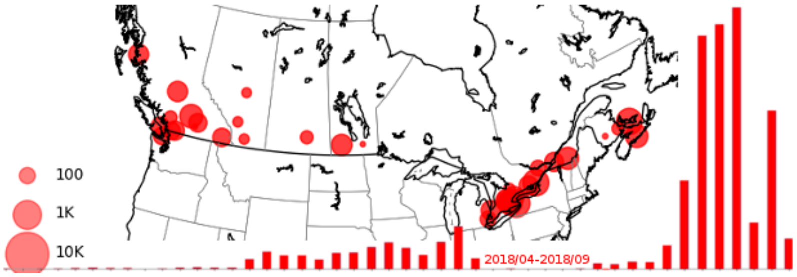


Figure 4.2: *Pervasiveness of Issue:* Distribution of escort ads in Canada shows a wide-spread presence of online escorting, esp. in major cities. Temporal distribution given in bar chart shows a clear rise in traffic in these alternative sites after the Backpage shut down in April 2018.

phone numbers in the dataset. By identifying groups in this dataset, we firstly ascertain that TrafficLight is robust to changes in the pattern of posting. We further present two case studies identified by TrafficLight, which leads us to the discovery of 'Smoke Screens'. Using *TLMod* we study the different attributes of ads within a cluster to find meta-clusters with similar properties, thus providing more insight into the different types of activities in the trafficking domain.

4.1.3 Synthetic Dataset

The two real-world datasets described in the previous section are not annotated and do not have ground truth labels suited to our task. Hence, for comparison to the baselines and provide a quantitative study of our method, we curate a synthetic generation process with densely injected blocks. We simulate characteristics similar to trafficking clusters obtained in the past as well as inputs by experts to create the synthetic data.

We start with sampling a set of 1800 dissimilar ads from *LCan100K*, all of which have a pairwise cosine similarity < 0.5. Of these ads, we randomly sample an ad and insert a dense block with various levels of perturbations. The degree of perturbation (p), number of injected blocks (k) and the size of each cluster (s_i) are parameters to the generator. Each injected block form one cluster of the ground-truth clustering that should be recovered by TrafficLight. The psuedocode for the generation process is given in Algorithm 4. For analysis we generate a dataset with 50 blocks and randomly sampled perturbation between 10% to 50% for each injected ad to create a synthetic data of 3200 ads. We also create a set of different datasets, each with increasing degrees of perturbation to test the robustness of our algorithm to changes in perturbation level.

```
Input: $S$ : seed set of dissimilar ads
  $C$ : corpus of words {# all the words in our corpus}
  $k$ : number of dense blocks
  $p_{low}$ : minimum level of perturbation
  $p_{high}$ : maximum level of perturbation
  injected ← 0
while $injected < k$ do
    $s \sim S$ {# randomly sample a seed ad from $S$}
    $n \sim \mathcal{U}(10, 50)$ {# sample size of the block from a uniform dist.}
    block_count ← 0
    while $block\_count < n$ do
        $p_i \sim \mathcal{U}(p_{low}, p_{high})$ {# sample level of perturbation for this injection}
        $l = p_i \times len(s)$ {# number of words to be replaced}
        inserted ← 0
        while $inserted < l$ do
            $w \sim C$ {# Sample a word from the corpus}
            $i \sim \mathcal{U}(0, len(s_i))$ {# Choose an index of the word to be replaced}
            $s[i] = w$
            inserted ← inserted + 1
        end
        block_count ← block_count + 1
    end
    injected ← injected + 1
end
```

Algorithm 4: Synthetic Data Generation

4.2 Results on Synthetic Dataset

4.2.1 Choice of N-grams

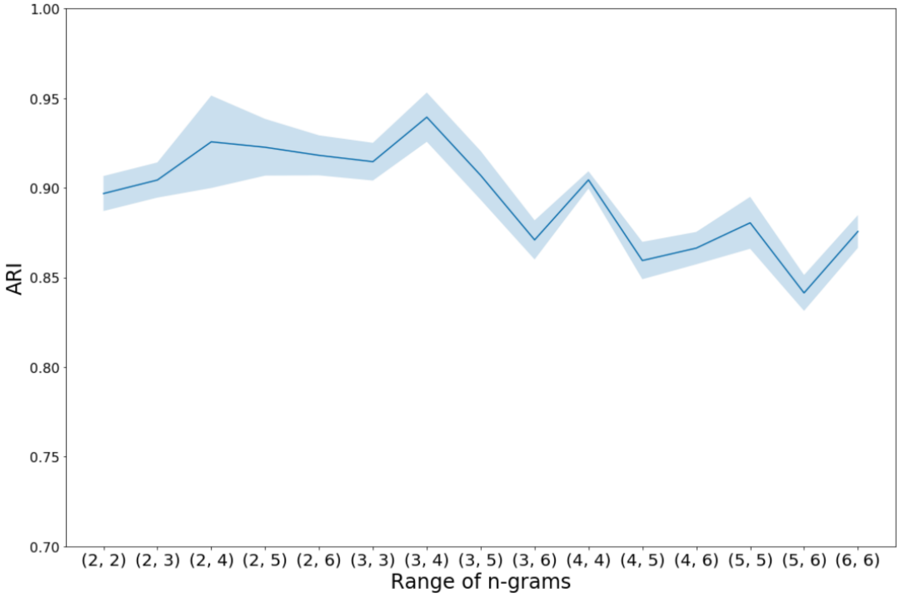


Figure 4.3: *Choice of n-grams:Bigrams and trigrams are the most important, the performance on the synthetic dataset increases slightly with increasing value of n-grams when bigrams are included.*

We design our solution using combination of bigrams to 5-grams. This was a carefully designed choice to obtain the best results as can be seen from Figure 4.3, where the performance is maximized when bigrams and trigrams are considered. The maximum size of 5-grams also matches with the recommended window size of NLP models like Word2Vec, Fasttext etc., some of which we will be using as our baselines.

4.2.2 Choice of k in SVD

For the choice of dimensions of embedding space, we conducted an experiment with various choices of k. As can be seen from Figure 4.4, the best performance is acheived at k =20 and k = 50. Since there is no significant difference in the ARI scores at these two values, we made the choice to go ahead with k = 20.

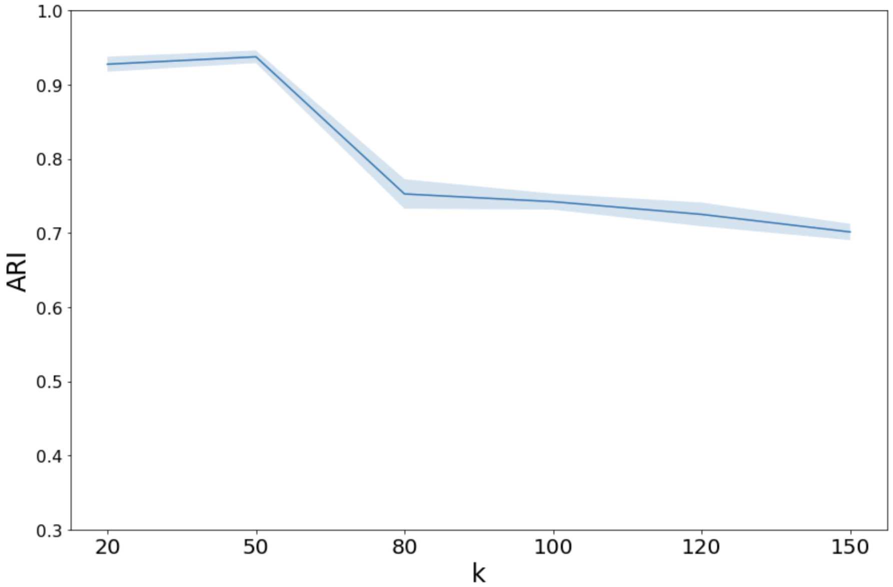


Figure 4.4: *Choice of k: Model performs equally well at k=20 and k=50 after which the performance decreases significantly. This evaluation too was done on the synthetic dataset.*

4.2.3 Ablation Study

In this section, we study the importance of each module of *TLDetect*. We justify our design choices of removing noise and splitting and merging of clusters for getting the final clusters by doing a component based analysis of the model's performance.

Components:	Embedding	Clustering	Noise Removal	Split & Merge	ARI
Variation 1 :	✓	✓			0.88 ± 0.002
Variation 2:	✓	✓	✓		0.88 ± 0.001
Variation 3:	✓	✓		✓	0.95 ± 0.002
Variation 4:	✓	✓	✓	✓	0.96 ± 0.01

Table 4.1: *Ablation Study:* Effect of different components on the quality of results as observed on the synthetic dataset

Here, we have four versions of the model as can be seen in Table 4.1. Most clustering algorithms along with HDBSCAN are quadratic in implementation. Thus to scale to larger datasets, we introduced a noise removal step by filtering out the points closer to origin (0,0). This is based on the understanding that these points do not have a significant magnitude along any of the components and hence, would not be a part of any significant cluster. As can be seen from Table 4.1, the removal of data preserves the performance of the model, thus proving that all significant clusters still remain in the unfiltered data.

In section 3.2.2, we outline a variation of HDBSCAN to obtain better clustering results. This is implemented in the model variation 3, which shows a boost in the results as compared to Variation 1. Finally, bringing all the components together into one single pipeline (Variation 4) gives the best results.

4.2.4 Comparison to Baselines

Unfortunately, it is *impossible to compare against earlier published methods*: either the task is different, or their code/data are not available. Most previous works focus on supervised, ad-level classification, e.g. HDTN [31], where the results are not clustered. The only model that produces group level results, TemplateMatching[36], reports a comparable performance on a similar setting but on a different dataset. Moreover, TrafficLight also provides summarization, meta-clustering and cluster characterization which are all novel to this work - see Table 2.1 for a conceptual comparison of TrafficLight with the current contenders and see Section 4.4 for the cluster characterization results and discussions.

Instead we compare against three potential baselines of *our own design*, based on state-of-the-art sentence and document embedding methods, i.e. Word2Vec [41], FastText [42], Doc2Vec[46] and BERT [47]. We train these NLP models using the 1 million ads crawled from Canadian escort websites. Table 4.2 summarizes the results of TrafficLight and these baselines. We use the same clustering method in all the baselines, i.e. HDBSCAN[56]. We ranked results of different baselines using the same metric, TLScore, when a ranked output is needed in the Trafficking-10K results as explained in 3.3.1. Please note that these experiments are not a general comparison with these generic text embedding methods. The results confirm that for the specific task of clustering text originated from the same template and in this particular domain, TrafficLight outperforms these alternatives. We discuss the performance of TrafficLight on the three datasets and the exact experimental settings used in the corresponding sections below.

Dataset	Synthetic	Live-*LCan100K*	Historic-*TF10k*	
Ground-truth	injected	phone numbers	expert tagging	
Metric	ARI	ARI	Avg. TS @ 20	Precision @ 20
TrafficLight (ours)	**0.96 ± 0.01**	**0.58**	**4.12**	**62.5**
Word2vec baseline	0.84 ± 0.03	0.34	1.21	17.6
FastText baseline	0.85 ± 0.02	0.37	3.22	57.5
Doc2Vec baseline	0.93 ± 0.01	0.32	2.03	25.3
BERT baseline	0.96 ± 0.05	0.32	2.2	43.4

Table 4.2: TrafficLight *performs better:* TrafficLight outperforms potential baselines on all three datasets.

The first column in Table 4.2 reports the cluster agreement between the results of different methods with the ground-truth injected clusters. We report the Adjusted Rand Index (ARI) [59] scores as a robust metric for cluster agreement[1], ignoring any ads marked as noise by the clustering algorithm. Please note that removing the noise means that the main objective is to find good quality clusters rather than recovering all the clusters. We can see that the performance of TrafficLight is almost perfect and on par with the last baseline, while significantly outperforming the other two. This validates that the clusters returned by TrafficLight are

[1] Similar comparison is obtained by NMI, the other commonly used metric, which is not chosen as the main metric to report given its known bias [60].

of good quality and correspond well with the injected clusters. We also observe that TrafficLight tends to find bigger clusters compared to the baselines with higher average text similarity. The average TLScore of the clusters obtained from TrafficLight is also higher than the baselines.

	Avg TS				Precision			
	@20	@30	@40	@50	@20	@30	@40	@50
TrafficLight (ours)	**4.12**	**2.60**	**2.53**	**2.34**	**0.62**	**0.56**	**0.44**	**0.47**
Word2Vec Baseline	1.21	1.7	1.7	1.5	0.17	0.27	0.26	0.21
Fasttext Baseline	3.22	2.2	2.1	2.3	0.57	0.44	0.39	0.43
Doc2Vec Baseline	2.03	1.6	1.6	1.7	0.25	0.24	0.27	0.27
BERT Baseline	2.2	2.3	2.2	2.1	0.43	0.45	**0.44**	0.39

Table 4.3: *Sensitivity to choice of k:* TrafficLight performs better than other baselines at varying degrees of threshold of the k-most suspicious clusters ranked using TLScore. Trafficking-10K was the only dataset on which this sensitivity analysis could be done since it is the only dataset with some labels.

4.3 Results on Historic Dataset

Here, we validate TrafficLight against the Trafficking-10K dataset, the only available labeled data for this task. Each ad is labeled on a scale of 0-6, signifying how likely the ad is to be HT. Since the original labels are for ad-level classification, and there is no grouping available, we can not use them directly for clustering validation. Instead, we look at the top clusters ranked by TrafficLight as the most suspicious and report the average trafficking scores of the ads in those clusters (Avg. TS), which are available as the ground-truth label. We also binarize the scores to HT (4-6) or not HT (0-3) and report the Precision in the top clusters. We can see in Table 4.2 that top 20 clusters ranked by TrafficLight as highly suspicious are significantly more likely to involve trafficking ads compared to the same number of clusters ranked highest by the baselines. We also compare the results at varying choices of k for picking the topmost k suspicious clusters and compare our results to the baselines (see Table 4.3). Though the precision reduces, reasons of which can be attributed to missing groups or labeling inconsistencies as seen in further sections, TrafficLight works best in capturing groups that are potentially trafficking.

We further investigate the cluster results by TrafficLight and discover corroboration, scooping, and new attack discoveries. More specifically, the cluster corresponding to the only verified convicted group of trafficking in this data is ranked high in our results and many of our clusters corroborate the ground-truth labels. However, TrafficLight also finds groups of ads that have been mislabeled due to human error, as discussed below.

4.3.1 Corroboration

In 2018, one of the companies working towards the detection of human trafficking helped with the takedown of a trafficking group with most victims of Asian origin. This group, referred to as UCA Convict in this paper (Figure 4.5), was identified due to a rare grammatical error in all the ads.

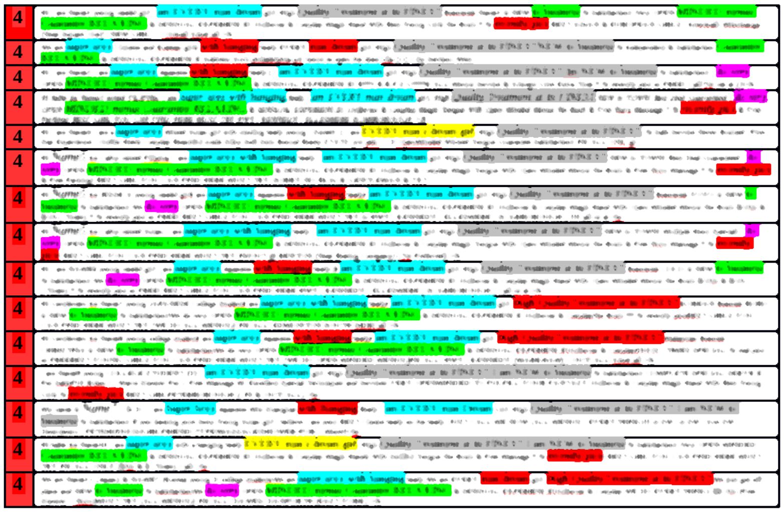


Figure 4.5: TRAFFICLIGHT *corroborates:* the third most suspicious cluster in our results includes 15 distinct ads and 11 different victims, most from Asian ethnic background. This UCA Convict case is the only verified case we are aware of in our Trafficking-10K dataset.

The results generated by TRAFFICLIGHT using Trafficking-10K spot this trafficking group, which were represented by a collection of 15 ads advertising 11 different victims. In general, it takes an average of 2 years for a case to go through the cycle of detection to investigation. With TRAFFICLIGHT, this can be reduced considerably.

4.3.2 Scooping Power

Imagine a human trying to find groups among a set of 10,000 documents. It is safe to say that it would take hours to finish the job alone. They would also make mistakes; especially given a limited time frame. Furthermore, some ads are difficult to discern when looked at individually. We find this to be the case when comparing our results

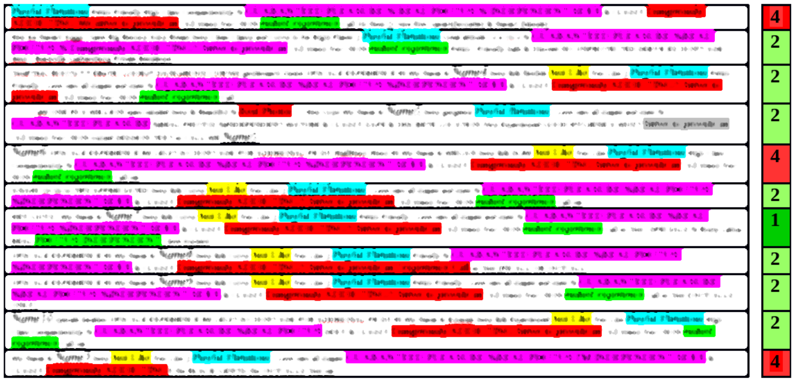


Figure 4.6: *Novel Lead Generation by* TRAFFICLIGHT*:* Domain experts verified this case to be one of the "Scooping" found by *TLDetect* in the Trafficking-10K dataset. This case was mislabeled by human annotators since they were looking at the ads individually.

to the labels in the Trafficking-10K dataset. TRAFFICLIGHT not only corroborates the findings of the human expert, but is able to find other organized groups, some of which were later termed as 'potential trafficking groups' by the domain experts i.e. the initial data annotators. We also look at clusters with a mid-range average label score ($\in [2.0, 3.5]$), to capture groups of ads that were similar enough to show the same authorship but were allocated different labels. Notice the ads in Figure 4.6, which have many common features but advertise different people. However, due to manual labeling, there are inconsistencies in the labels allocated to these ads (8 have been marked as 'not trafficking').

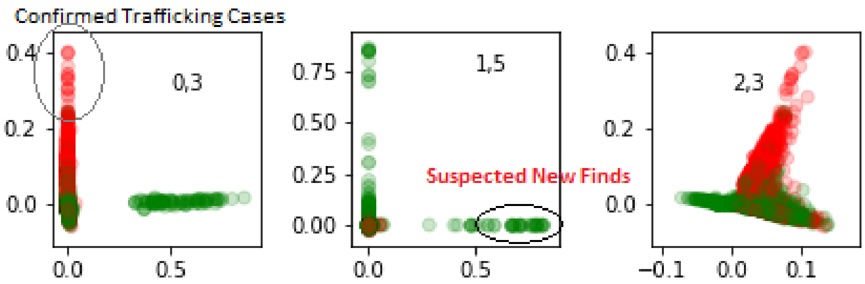


Figure 4.7: TrafficLight *spots new leads:* Red points are ads in the Trafficking-10K dataset that have been marked as trafficking by the annotators, while greens are non-trafficking marked ads. We see the presence of groups among the non-trafficking ads. These could be mislabeling, some of which were found and presented in Section 4.3.2.

4.3.3 New Attack Discovery

Observe the spokes in Figure 4.7. The collection of points near the end of spokes along the axes indicates the presence of communities. We find such groups among trafficking ads (marked in red) as well as non-trafficking labeled ads (marked in green). As per our analysis, there might be some undetected organized groups among the non-trafficking ads that could be helplines, massage parlors, or some mislabeled trafficking clusters.

Figure 4.8 is one of the instances of such a group detected by TrafficLight where all the ads have been marked as 'non-trafficking'. However, experts (original data annotators) confirmed that looking at these ads together with their connections highlighted provides a strong signal that this might have been a case of trafficking. Such cases that miss the eye when zoomed in on ad-level would either never have

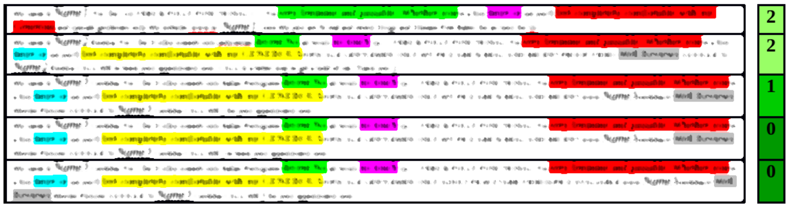


Figure 4.8: *Novel Lead Generation by* TrafficLight*:* Domain experts verified this case to be one of the ”New Attack Discoveries” found by *TLDetect* in the Trafficking-10K dataset. This case was mislabeled by human annotators since they were looking at the ads individually.

been picked or taken too much time to be identified without TrafficLight.

4.4 Results on Live Data

The results on the Canadian dataset reported in Table 4.2 show the quality of clusters found by the methods when the phone numbers[2] are masked from the input and used as the ground-truth. More specifically, ads that share the same phone number are grouped as the ground-truth clusters, i.e. each phone number gives a cluster.

In most cases, a smart criminal will use different phone numbers for different victims, and changes the numbers frequently to avoid detection. It is also common for regular users of these platforms to change their number frequently. In fact, most of the clusters we find include more than one phone number, even though they clearly belong to the same activity. We can see that the clusters returned by TrafficLight show a meaningful correlation with these weak labels, which is significantly higher than the baselines. We want to emphasize that this agreement is strong given the ground-truth is weak and doesn't correspond to the true results but only correlates with it.

In the next section, we discuss some example cases discovered by TrafficLight, followed by a discussion on smoke screens, a spamming method used by traffickers to perplex investigators.

[2] when detected with our extractor

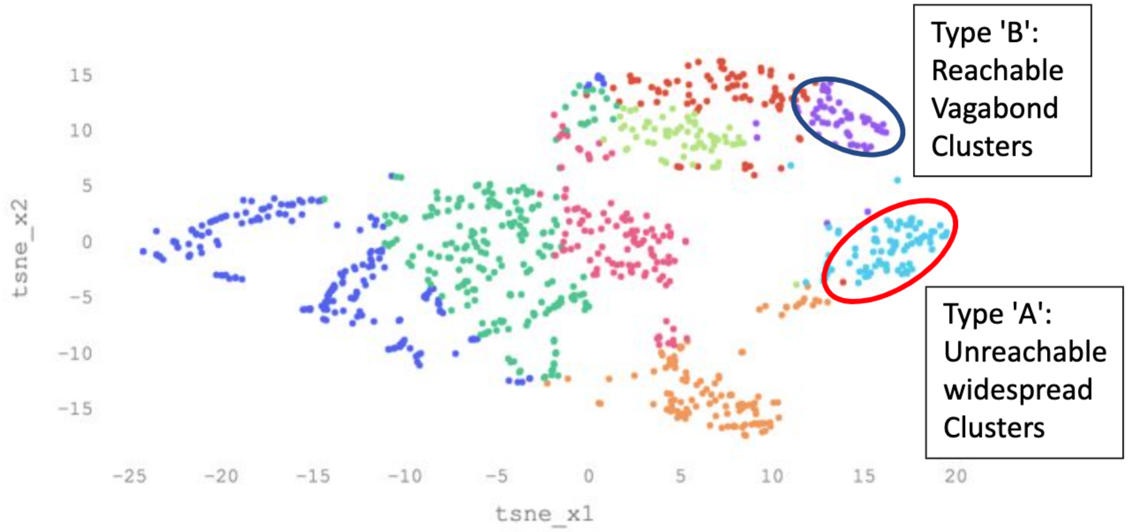


Figure 4.9: *MO Detection and Meta-Clusters:* Characterizing clusters from TrafficLight: Figure shows TSNE embeddings of all clusters and the resulting meta-clusters obtained from *LCan100K*, two of which correspond to the two case studies discussed Section 4.4.

4.4.1 Case Study 1: *Unreachable Widespread Clusters*

We find two groups, one advertising 5 different people from Indian/Asian origin, and the other a group of Latino females. We notice that both these groups have ads spread through the entirety of Canada and use many different phone numbers (see Fig 4.11). On further investigation into the two cases with the help of domain experts, we discover that these are probably hoax ads injected into the system to confuse law enforcement and hamper investigation. We further verify that most of the ads in these groups use fake/unreachable phone numbers, using the Twilio platform [58].

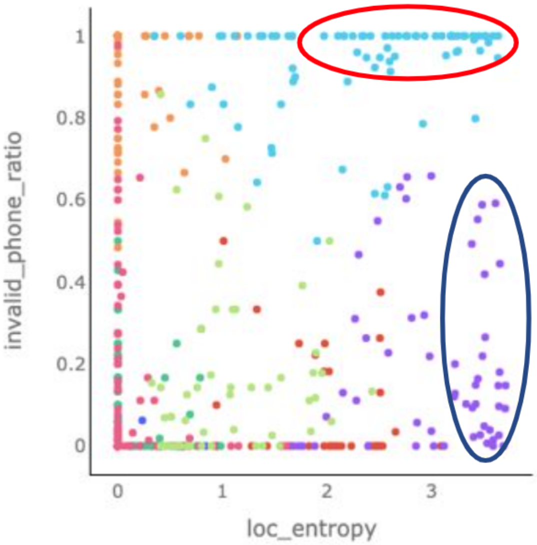


Figure 4.10: *MO Detection and Meta-Clusters:* Characterizing clusters from TrafficLight: This plot explains the distinction between the two groups highlighted in Figure 4.9, where there is a clear demarcation on the location entropy vs invalid phone ratio attributes

4.4.2 Case Study 2: *Reachable Vagabond Clusters*

We spot another widespread cluster having activities across Canada. This cluster has a burst of activities spread over 2 days (May 15, 2019 and May 21, 2019) followed by a few activities afterwards. 12 different individuals are advertised over more than 300 ads with the usage of 33 different valid phone numbers. Manual investigation into this cluster suggests that a single image is associated to many of these ads, raising doubts on the credibility of the posted ads.

These two case studies are part of a larger pattern, and through these we gain insight into a major roadblock for detection of human trafficking groups, namely 'Smoke Screens'.

4.4.3 MO Detection - ‘Smoke Screens’

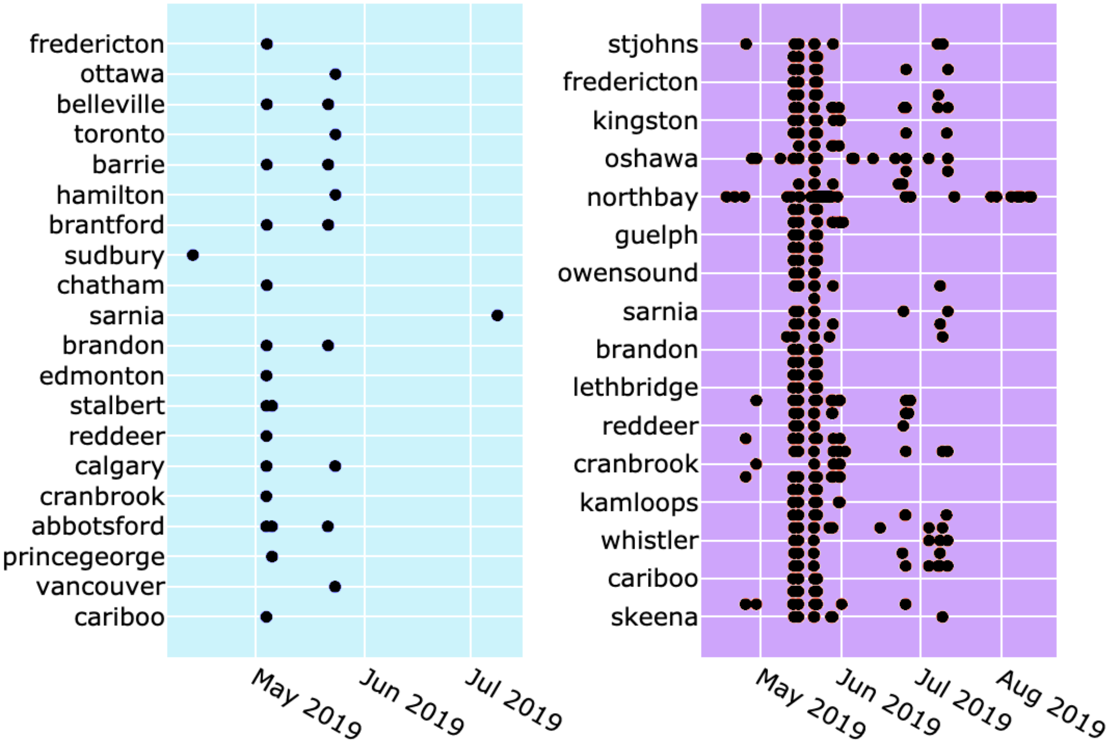


Figure 4.11: Activity plot for confirmed spam Latino group from Case study 1 (left), and activity plot for Case Study 2 from Reachable Vagabond Clusters (right).

We notice the presence of a significant number of spam ads of varied nature on these online platforms. These spam ads obfuscate the online space making it difficult to detect legitimate trafficking groups. In this section, we highlight two of the meta-clusters obtained through *TLMod*- Type ‘A’ (Unreachable Widespread Clusters) and Type ‘B’ (Reachable Vagabond Clusters) corresponding to the two case studies in the previous section; their names attributed to invalid phone ratio and location entropy as shown in Figure 4.9 and 4.10. Both the case studies from

Type ‘A’ and Type ‘B’ were later confirmed to be spams by the domain experts. Furthermore, as seen in Figure 4.11, the two studies in Type ‘A’ and Type ‘B’ both have a synchronized pattern of posting within their group. For case study 1, most ads are posted around two dates in May 2019 and Jun 2019, suggesting the involvement of a single author or script. Similar pattern can be observed in case study 2 as well, where multiple ads in different locations are posted on the same date.

This higher level characterization not only helps us understand the different types of activities prevalent in the online domain, but can also save immense investigation time. In the example above, we can safely ignore all the clusters under Type ‘A’, thus significantly reducing the number of clusters to be looked at.

Chapter 5

Conclusions and Future Work

The problem of detecting human trafficking is crucial as it affects hundreds of thousands of individuals, including children. In this paper, our focus is primarily to provide a solution that can help investigators swift through the relevant data. Due to the large number of advertisements being posted daily, it has become physically difficult for a human to look at these ads, find those showing trafficking signs and do case building. Due to these reasons, many cases go unidentified or take too long to be detected and solved.

Through our work, we present a solution to detect these trafficking cases, rank them in the order of suspiciousness, highlight the common evidences and further characterize the found clusters into meta-clusters. We use two real world datasets to validate our method as well as a synthetically generated dataset to provide a proof of concept with strong labels. As we saw in the reported experiments, our proposed method, TrafficLight, performs well on all three datasets. This is one of the first works to solve the problem of lead generation as well as case building in the same pipeline. Our study on meta-clusters is also novel to this work as it has not been studied before, to the best of our knowledge.

The contributions of our work are summarized below:

- Novel Lead Generation : TrafficLight is label-independent and unsupervised, which prevents the scope of error in human labeling. As we saw in Section 4.3, human annotations are often erroneous, hence using supervised training method using such labels will yield incorrect results. Using a label independent technique further helped us uncover previously unobserved cases of organized activities, thus unearthing new behavior.

- M.O. Detection : We present a novel method of understanding the different types of activities in online escort ads. We present two case studies attributing to two different kinds of characterizations. With a confirmed case, we also highlight the presence of 'smoke screens' possibility as an adversarial technique to confuse law enforcement.

- Productivity Boost : TrafficLight picks the topmost suspicious clusters and provides summarized results by highlighting the connection between similar ads. Grouping together similar ads accompanied by highlighting of evidences makes it easier for investigators to spot a trafficking cluster. This makes the results more interpretable and accelerates investigation by an estimated 25x.

5.1 Future Work

A good tool for detecting human trafficking requires a lot of different components to work in tandem to give us the best results. There is a lot of technological advancement required in the field that can aid the investigators and save them a lot of manual work. Our solution addressed the issue of being able to spot a needle in the haystack by looking for densely connected sets of ads. However, there is still a lot of open problems to be solved.

One of the most promising future works would be an extension of the meta clustering approach for characterizing the type of clusters. There is also a lot of work done on the entity extraction techniques for escort advertisements. Often traffickers obfuscate important information like phone numbers, rates, etc in order to avoid detection. Such attributes are vital to drawing connections between ads. The phone numbers can act as weak labels and can act as must-have links to improve the performance of clustering. It would also be interesting to study trends on the cross-country posting or influence of geography or country on the ads posted online. Looking at sources of ads present in various countries, it would be also interesting to uncover transnational criminal groups and compare activities across and within different nations.

Bibliography

[1] ILO, *Forced labour, modern slavery and human trafficking*, International Labour Organization: https://www.ilo.org/global/topics/forced-labour/lang--en/index.htm, 2016.

[2] NCMEC, *Child sex trafficking*, NCMEC: http://www.missingkids.com/theissues/trafficking, 2019.

[3] Thorn, *Survivor survey r5*, http://www.thorn.org/wp-content/uploads/2015/02/Survivor_Survey_r5.pdf, 2015.

[4] A. Wagner, *Human trafficking & online prostitution advertising*, Wagner House: https://wagner.house.gov/Human%20Trafficking%20%26%20Online%20Prostitution%20Advertising, 2019.

[5] M. Kennedy, *Craigslist shuts down personals section after congress passes bill on trafficking*, 2018. [Online]. Available: https://www.npr.org/sections/thetwo-way/2018/03/23/596460672/craigslist-shuts-down-personals-section-after-congress-passes-bill-on-traffickin.

[6] L. L. Sarah N. Lynch, *Backpage shutdown*, https://www.reuters.com/article/us-usa-backpage-justice/sex-ads-website-backpage-shut-down-by-u-s-authorities-idUSKCN1HD2QP, 2018.

[7] R. Tarinelli, *Online sex ads rebound, months after shutdown of backpage*, https://nationalpost.com/pmn/news-pmn/online-sex-ads-rebound-months-after-shutdown-of-backpage, 2018.

[8] R. C. Intelligence. (2019). Human trafficking in canada, [Online]. Available: http://www.kristenfrenchcacn.org/wp-content/uploads/2019/08/Human-Trafficking-In-Canada-2010-Unclassified.pdf.

[9] A. Cotter. (2020). Trafficking in persons in canada, [Online]. Available: https://www150.statcan.gc.ca/n1/pub/85-002-x/2020001/article/00006-eng.htm.

[10] M. Lopez-Martinez. (). Sex trafficking still a prevalent issue across canada, advocates and police say, [Online]. Available: https://www.ctvnews.ca/canada/sex-trafficking-still-a-prevalent-issue-across-canada-advocates-and-police-say-1.4820944.

[11] M. The Honourable Ralph Goodale P.C. (). National strategy to combat human trafficking 2019-2024, [Online]. Available: https://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/2019-ntnl-strtgy-hmnn-trffc/index-en.aspx.

[12] RCMP. (2020). Recognizing human trafficking victims, [Online]. Available: https://www.rcmp-grc.gc.ca/en/human-trafficking/recognizing-human-trafficking-victims.

[13] 106th United States Congress, "Victims of trafficking and violence protection act of 2000," *Public Law*, 2000.

[14] B. Court, *In the supreme court of british columbia*, https://www.bccourts.ca/jdb-txt/SC/14/17/2014BCSC1727.htm, 2014.

[15] D. o. O. Department of Justice U.S. Attorney's Office, *Nationwide sting operation targets illegal asian brothels, six indicted for racketeering*, https://www.justice.gov/usao-or/pr/nationwide-sting-operation-targets-illegal-asian-brothels-six-indicted-racketeering.

[16] [Online]. Available: https://centerforimprovinginvestigations.org/human-trafficking-investigations/.

[17] [Online]. Available: https://www.marinusanalytics.com/.

[18] A. Dubrawski, K. Miller, M. Barnes, B. Boecking, and E. Kennedy, "Leveraging publicly available data to discern patterns of human-trafficking activity," *Journal of Human Trafficking*, vol. 1, no. 1, pp. 65–85, 2015.

[19] K. Miller, E. Kennedy, and A. Dubrawski, "Do public events affect sex trafficking activity?" 2016.

[20] M. Kejriwal and P. Szekely, "Knowledge graphs for social good: An entity-centric search engine for the human trafficking domain," *IEEE Transactions on Big Data*, 2017.

[21] ——, "Information extraction in illicit web domains," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 997–1006.

[22] P. Szekely, C. A. Knoblock, J. Slepicka, A. Philpot, A. Singh, C. Yin, D. Kapoor, P. Natarajan, D. Marcu, K. Knight, *et al.*, "Building and using a knowledge graph to combat human trafficking," in *International Semantic Web Conference*, Springer, 2015, pp. 205–221.

[23] M. Kejriwal and P. Szekely, "An investigative search engine for the human trafficking domain," in *International Semantic Web Conference*, Springer, 2017, pp. 247–262.

[24] R. Rabbany, D. Bayani, and A. Dubrawski, "Active search of connections for case building and combating human trafficking," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*, 2018.

[25] R. Kapoor, M. Kejriwal, and P. Szekely, "Using contexts and constraints for improved geotagging of human trafficking webpages," in *Proceedings of the Fourth International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data*, ACM, 2017, p. 3.

[26] H. Wang, A. Philpot, E. Hovy, and M Latonero, "Data mining and integration to combat child trafficking," *Retrieved from Carnegie Mellon University, School of Computer Science website: http://www. cs. cmu. edu/˜ hovy/papers/12dgo-trafficking. pdf*, 2014.

[27] M. Kejriwal, J. Ding, R. Shao, A. Kumar, and P. A. Szekely, *Flagit: A system for minimally supervised human trafficking indicator mining*, 2017. arXiv: 1712.03086. [Online]. Available: http://arxiv.org/abs/1712.03086.

[28] H. Alvari, P. Shakarian, and J. Snyder, "A non-parametric learning approach to identify online human trafficking," English (US), in *IEEE International Conference on Intelligence and Security Informatics: Cybersecurity and Big Data, ISI 2016*, United States: Institute of Electrical and Electronics Engineers Inc., Nov. 2016, pp. 133–138. DOI: 10.1109/ISI.2016.7745456.

[29] H. Alvari, P. Shakarian, and J. E. K. Snyder, "Semi-supervised learning for detecting human trafficking," *Security Informatics*, vol. 6, no. 1, p. 1, 2017. DOI: 10.1186/s13388-017-0029-8. [Online]. Available: https://doi.org/10.1186/s13388-017-0029-8.

[30] K. Hundman, T. Gowda, M. Kejriwal, and B. Boecking, "Always lurking: Understanding and mitigating bias in online human trafficking detection," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, ACM, 2018, pp. 137–143.

[31] E. Tong, A. Zadeh, C. Jones, and L. Morency, "Combating human trafficking with multimodal deep models," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, July 30 - August 4, Volume 1: Long Papers*, R. Barzilay and M. Kan, Eds., Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1547–1556, ISBN: 978-1-945626-75-3. DOI: 10.18653/v1/P17-1142. [Online]. Available: https://doi.org/10.18653/v1/P17-1142.

[32] L. Wang, E. Laber, Y. Saanchi, and S. Caltagirone, “Sex trafficking detection with ordinal regression neural networks,” *arXiv preprint arXiv:1908.05434*, 2019.

[33] Y. Liu, L. Zhu, P. Szekely, A. Galstyan, and D. Koutra, “Coupled clustering of time-series and networks,” 2019.

[34] C. Nagpal, K. Miller, B. Boecking, and A. Dubrawski, “An entity resolution approach to isolate instances of human trafficking online,” *CoRR*, vol. abs/1509.06659, 2015.

[35] R. S. Portnoff, D. Y. Huang, P. Doerfler, S. Afroz, and D. McCoy, “Backpage and bitcoin: Uncovering human traffickers,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[36] L. Li, O. Simek, A. Lai, M. P. Daggett, C. K. Dagli, and C. Jones, “Detection and characterization of human trafficking networks using unsupervised scalable text template matching,” in *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, N. Abe, H. Liu, C. Pu, X. Hu, N. Ahmed, M. Qiao, Y. Song, D. Kossmann, B. Liu, K. Lee, J. Tang, J. He, and J. S. Saltz, Eds., Seattle, WA: IEEE, 2018, pp. 3111–3120, ISBN: 978-1-5386-5035-6. DOI: 10.1109/BigData.2018.8622189. [Online]. Available: https://doi.org/10.1109/BigData.2018.8622189.

[37] G. H. Golub and C. Reinsch, “Singular value decomposition and least squares solutions,” in *Linear Algebra*, Springer, 1971, pp. 134–151.

[38] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[39] I. T. Jolliffe, “Principal components in regression analysis,” in *Principal component analysis*, Springer, 1986, pp. 129–155.

[40] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[41] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *NIPS’13 Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, IEEE, Lake Tahoe, USA: ACM, 2013, pp. 3111–3119.

[42] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, *Fasttext.zip: Compressing text classification models*, 2016. arXiv: 1612.03651. [Online]. Available: http://arxiv.org/abs/1612.03651.

[43] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162.

[44] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.

[45] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” *arXiv preprint arXiv:1802.05365*, 2018.

[46] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *International conference on machine learning*, 2014, pp. 1188–1196.

[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[48] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, “Tinybert: Distilling bert for natural language understanding,” *arXiv preprint arXiv:1909.10351*, 2019.

[49] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos, “Copycatch: Stopping group attacks by spotting lockstep behavior in social networks,” in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 119–130.

[50] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, “Catchsync: Catching synchronized behavior in large directed graphs,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 941–950.

[51] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, “Fraudar: Bounding graph fraud in the face of camouflage,” in *Proceedings of the 22Nd*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD '16, San Francisco, California, USA: ACM, 2016, pp. 895–904, ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939747. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939747.

[52] B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos, "Eigenspokes: Surprising patterns and scalable community chipping in large graphs," in *Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part II*, M. J. Zaki, J. X. Yu, B. Ravindran, and V. Pudi, Eds., ser. Lecture Notes in Computer Science, vol. 6119, Hyderabad, India: Springer, 2010, pp. 435–448, ISBN: 978-3-642-13671-9. DOI: 10.1007/978-3-642-13672-6_42. [Online]. Available: https://doi.org/10.1007/978-3-642-13672-6_42.

[53] M. Jiang, A. Beutel, P. Cui, B. Hooi, S. Yang, and C. Faloutsos, "A general suspiciousness metric for dense blocks in multimodal data," in *2015 IEEE International Conference on Data Mining, ICDM 2015, November 14-17, 2015*, C. C. Aggarwal, Z. Zhou, A. Tuzhilin, H. Xiong, and X. Wu, Eds., Atlantic City, NJ, USA: IEEE Computer Society, 2015, pp. 781–786, ISBN: 978-1-4673-9504-5. DOI: 10.1109/ICDM.2015.61. [Online]. Available: https://doi.org/10.1109/ICDM.2015.61.

[54] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, 1996, pp. 226–231.

[55] M. Ankerst, M. M. Breunig, H. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," in *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA*, A. Delis, C. Faloutsos, and S. Ghandeharizadeh, Eds., ACM Press, 1999, pp. 49–60, ISBN: 1-58113-084-8. DOI: 10.1145/304182.304187. [Online]. Available: https://doi.org/10.1145/304182.304187.

[56] L. McInnes and J. Healy, "Accelerated hierarchical density based clustering," in *2017 IEEE International Conference on Data Mining Workshops, ICDM Workshops 2017, November 18-21, 2017*, R. Gottumukkala, X. Ning, G. Dong, V. Raghavan, S. Aluru, G. Karypis, L. Miele, and X. Wu, Eds., New Orleans, LA, USA: IEEE Computer Society, 2017, pp. 33–42, ISBN: 978-1-5386-3800-2. DOI: 10.1109/ICDMW.2017.12. [Online]. Available: https://doi.org/10.1109/ICDMW.2017.12.

[57] S. Card, T. P. Moran, and A. Newell, *The model human processor- an engineering model of human performance*, 1986.

[58] NA. (). Twilio, [Online]. Available: https://www.twilio.com/.

[59] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[60] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Is a correction for chance necessary?" In *Proceedings of*

the 26th annual international conference on machine learning, 2009, pp. 1073–1080.