Detection and Visualization of Human Sex Trafficking in Online Escort Advertisements

Catalina Vajiac

CMU-CS-26-113

May 2026

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Christos Faloutsos, Chair
Rayid Ghani
Adam Perer
Duen-Horng Chau (Georgia Institute of Technology)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored in part by the Pennsylvania Infrastructure Technology Alliance, a partnership of Carnegie Mellon, Lehigh University and the Commonwealth of Pennsylvania's Department of Community and Economic Development (DCED), in part by a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development, by the PNC Center for Financial Services Innovation, by Defensewerx under award number SOE00400, and by the National Science Foundation under award numbers DGE-1745016 and DGE-2140739. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Human Trafficking, Text Clustering, Data Mining, Data Visualization, Graph-based Visualization, Spatio-Temporal Visualization

To my parents, for their unwavering belief in me since the very beginning,
and to Jamie, who walked every mile of this journey with me.

Abstract

Human trafficking (HT) for forced sexual exploitation is incredibly pervasive, affecting an estimated 6.3 million people at any given time. The majority of victims are advertised online, mainly through online escort websites. Practitioners who want to help these victims, including criminologists, social workers, and law enforcement agencies, often manually scroll through these escort websites to try to find leads. This process is incredibly inefficient, as it requires lots of time, and ineffective, as the chance of finding HT while scrolling through ads at random is small. While a few industry tools exist to analyze ads, they use basic techniques such as connecting ads through metadata, and to our knowledge, don't analyze the ad text at all. Furthermore, these tools often have extremely limited functionality, requiring practitioners to export data out of one interface and into another to analyze a possible lead.

In recent years, some HT cases have been found, by chance, through a practitioner discovering that multiple ads have similar text, since many HT cases are part of organized crime groups. These cases are characterized by large numbers of similar looking ads in multiple locations and advertising multiple people, making it unlikely that only one individual posted all the content. These insights can be leveraged to automate lead generation using publicly data so that practitioners can help HT victims more quickly and effectively.

In this thesis, I propose to assist practitioners in identifying potential human trafficking cases by: (1) developing *scalable and explainable clustering algorithms* based on text and shared locations for finding large organized crime groups in escort ad data, and (2) creating *intuitive visualization techniques* for presenting the results of these to practitioners. These visualizations include an interface for quick label generation so that we can effectively evaluate our algorithms, as well as novel techniques for exploring connections in metadata throughout time. We also evaluate our algorithms to ensure they are fair across advertised race and ethnicity.

The share of individuals affected by human sex trafficking has only increased throughout time, and the amount of escort ad data has increased exponentially. Through the creation of better algorithms and effective visualizations, we hope to make the laborious task of lead generation much easier for practitioners, so that they can focus on case-building using private data and getting HT victims the help that they need.

Acknowledgments

I would like to thank my advisor, Christos Faloutsos, for his unwavering advice, support, and kindness throughout my PhD. His mentorship has been the cornerstone of my growth as a researcher. I am also deeply grateful to my committee members for their unique roles in my journey. I would like to thank Rayid Ghani for helping me better understand how to bridge research and "social good" and for leading the DSSG summer program in 2022, one of my most formative experiences at CMU. A special thanks to Polo Chau for taking me under his wing and guiding me through my first visualization project and for truly making me feel at home in the visualization community. I would also like to thank Adam Perer, not only for his support as a committee member, but for his Interactive Data Science class, which originally kickstarted my passion for visualization.

I am fortunate to have worked with a wonderful group of collaborators, including Andreas Olligschlaeger, Benedikt Boeking, Reihaneh Rabbany, Cara Jones, Rebecca Mackenzie, Pratheeksha Nair, and Aayushi Kulshrestha. I am also indebted to my DSSG colleagues, especially Abby Smith, Arun Frey, Joe Baumann, Kasun Amarasinghe, Lily Millán Nunez, and Alice Lai. I am grateful for my labmates, Saranya Vijayakumar, Minji Yoon, and Jeremy Lee. My appreciation also goes to the CSD staff who work tirelessly to keep everything running smoothly, including Deb Cavlovich, Jenn Landefeld, Matthew Stewart, and Charlotte Yano. I have also been fortunate to share my office at CMU with incredible women. My heartfelt thanks goes to Bailey Flanigan, Sara McAllister, and Ananya Joshi for their unwavering friendship and support, as well as Jennifer Brana, Rose Silver, Ray Ware, Naama Ben-David, and Katherine Kosaian.

Beyond CMU, I am grateful for the lifelong friendship from my closest friends, Joanna Stochitoiu and Diana Dimian. To my cousins, Carla and Tibi Sosea: thank you for being part of our 'pod' with my parents during the pandemic. Having three of us pursuing CS PhDs simultaneously during such an uncertain time was no small feat.

To my parents: your lifelong encouragement and unconditional support have been among the few constants in my life. You have always been my greatest champions, and I am profoundly grateful for everything you have given to help me reach this moment. Finally, to Jamie, the 'loaf' of my life: thank you not just for your constant support throughout my PhD, but also for moving to Pittsburgh to help me realize this dream.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> "While Jeffrey Epstein is dead and gone, there are many others around the world who are still committing the same kinds of crimes that he did.
> ...In America, where only 4% of law-enforcement agencies have personnel dedicated to exposing human trafficking, most victims must rely on their own wits, and on luck, to survive. I want to change that. I want not just to hold abusers accountable but also to challenge the ways that all too often our legal system protects those abusers."
>
> Virginia Roberts Giuffre
> *Nobody's Girl: A Memoir of Surviving Abuse and Fighting for Justice*

1.1 Human Sex Trafficking (HT) in the Modern World

Human trafficking is defined as the use of force, fraud, or coercion to obtain some type of labor or commercial sex act [82]. Human trafficking for forced sexual exploitation (hereby referred to as HT) is incredibly pervasive, affecting an estimated 6.3 million people at any given time [85], a 131% increase since 2017 [84]. Most victims of HT are advertised online, mainly through classified ad-style posts on escort websites, which at-will sex workers also use for advertising [103].

To mitigate this global issue, governments have tried a mix of generic and targeted approaches. While policy changes provide legal protections, services, and healthcare to both at-will sex workers and HT victims alike, targeted approaches like wellness checks or federal investigations are utilized once a case is manually identified. Due to the sheer volume of escort ads posted daily on over 20 websites worldwide, *identifying potential cases is incredibly difficult.* As a first attempt, practitioners would navigate to an escort site manually and comb advertisements for any suspicious signals [43], but this approach has some drawbacks:

(a) it's *time-consuming and impractical* to expect workers to find cases by manually reading each ad; time scrolling on various escort ad websites to try and find a lead;

(b) it's *ineffective for organized crime groups* who make up the vast majority of HT cases, since the suspicion often comes not from one individual advertisement, but instead by observing a group of highly similar posts advertising different people [18]; and

(c) it *ignores other relevant advertisers* on these websites, including **spammers** posting fake advertisements en masse or **massage parlors**, which may or may not involve HT.

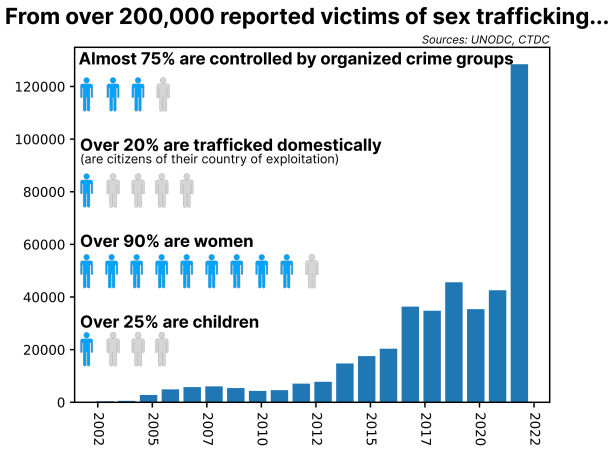


Figure 1.1: Global number of reported sex trafficking has been growing exponentially and tends to primarily affect those who identify as women. (Note: the true number of cases are vastly under-reported, so the true figures are likely much higher.)

1.2 Practical Challenges Informing Research Directions

Practitioners have long been interested in data-driven approaches to better identify possible leads of HT in escort ads since, theoretically, some of the aforementioned challenges could be addressed. At the start of this thesis, some *practical challenges* (hereby referred to as ***PC1: Data**— ***PC4: Interpret** throughout this thesis) have made it difficult for experts to directly use the state-of-the-art.

1.2.1 *PC1: Dirty Data breaks standard assumptions of online text content, affecting performance of off-the-shelf models.

While the capabilities of language models have exploded in recent years, there are some quirks of escort ad data that make it difficult to directly adopt these models to our purposes, especially due to the sheer amount of highly explicit content and writing style unique to escort ads. Since most models are not built or evaluated with this use case in mind, model performance suffers, even after fine-tuning [61]. Specifically, writing style quirks unique to escort ads, such as grammatical or spelling differences and secret encodings (i.e. emojis denoting certain services) can also affect performance.

Our Solution: Build models from scratch using insights from HT domain knowledge and *procure labels so we can evaluate on real-world data*, since we know performance in one domain may not generalize to ours.

1.2.2 *PC2: Few/Expensive Labels are difficult to procure for HT, severely limiting algorithm choices and evaluation.

Since HT is under-reported and difficult to prosecute, ground-truth labels are difficult to come by; the standard process involves domain experts labeling ads for their likelihood of TRAFFICKING. Practitioners previously tried manually labeling ads for a classification task (see Section 3.6.1.2) but found it too onerous and time-consuming to ever repeat. However, more up-to-date labels will be needed to continue evaluating future methods, especially as the behaviors of perpetrators change over time.

Our Solution: First, design *unsupervised* algorithms so smaller datasets can be used for evaluation only (see Chapter 3, then design a custom data-labeling interface, carefully visualizing results to reduce the labeling burden, making manual labeling feasible again (see Chapter 4).

1.2.3 *PC3: Legal/Financial Limits affect possible model choices due to computational constraints and concerns about explainability

In a government or policing context, models can easily infer negative biases about race/ethnicity or other physical characteristics present in escort ads, which could lead to negative outcomes for some at-will sex workers. Since federal or international government agencies are often involved in HT cases (especially for organized crime groups), there is concern about "interpretability" from the expert's perspective, especially for methods used to *flag ads as suspicious*, as they could capitalize on racial biases that unfairly target some while leaving others behind.

These concerns are not just one-sided; tech companies that provide access to off-the-shelf models, understandably, are actively trying to limit the amount of sexually explicit content that is processed and/or generated by LLMs by implementing safeguards where the LLM refuses to generate certain content[1].

Our Solution: Be selective about which methods we use from an ethical and practical perspective. Use methods that meet the domain expert's criteria for "interpretability" and, as much as possible, check for racial biases (see: Chapter 6).

1.2.4 *PC4: Expert interpretability is difficult with clustering algorithm results, making it difficult for experts to incorporate models

Specificallly, when practitioners investigate *groups* of suspicious ads, they consider the timeline & volume of ads posted, patterns in the contact information used in the ads, spread and timeline of the geographic footprint, and the similarity of advertisements. They also sometimes consider the connections between some of these entities (i.e. ads that use the same phone number). All of this data can be overwhelming to parse, but is necessary to show a domain expert whether the goal is to label a suspicious cluster or recommend it for further intervention.

[1]Interestingly, we saw these safeguards become more comprehensive over time as we were testing out these models – we observed previously-working prompts that, a few months after their initial use, were no longer able to "trick" the LLM into generating sexually explicit content.

Our Solution: Provide experts with more than just algorithm results, using data visualization techniques to provide them with a summary of all the relevant information needed for them to make an informed decision on a possible case (see: Chapter 5).

1.3 Contributions and Thesis Organization

To manage these practical challenges, we must ensure that any improvements to the current process are **realistic** (i.e. respecting the constraints on data availability, compute power, and model selection) and **interpretable** (i.e. experts of all backgrounds can interpret and use the results). The contributions of this thesis have been organized under **Part 1: Algorithms** and **Part 2: Visualization**, where each chapter describes one or more projects towards a common research goal, as well as how those projects address *PC1 – *PC4.

Part 1: Algorithms for finding suspected HT in escort ads

Chapter 3: InfoShield and DeltaShield find microclusters of suspicious ads, outperforming the state-of-the-art all while addressing a more realistic setting.

- Addressing Challenges:
  - InfoShield is unsupervised (addr. *PC1: Data),
  - uses principled methods that are more easily interpretable (addr. *PC3: Legal), and
  - provides a summary template for each microcluster of similar ads (addr. *PC4: Interpret).
- Impact:
  - In the news: InfoShield highlighted on local Pittsburgh news channel WPXI, Carnegie Mellon University, and McGill University.
  - InfoShield is also currently being integrated by *Marinus Analytics*.

Part 2: Visualization for data labeling and interpretation

Chapter 5: TrafficVis provides a **10x** speedup in labeling time vs. manual labeling (addr. *PC3: Legal), the previous standard and was highly rated in expert feedback.

- Addressing Challenges:
  - By carefully visualizing results from InfoShield with other relevant data in an interactive dashboard (addr. *PC1: Data, *PC4: Interpret),
  - practitioners can finally see all the information available in the ads that can help them label the cluster (addr. *PC2: Labels) as HT, at-will, or other.
  - TrafficVis provides a **10x** speedup in labeling time vs. manual labeling (addr. *PC3: Legal), the previous standard and was highly rated in expert feedback.
- Impact:

- Best Poster Honorable Mention VIS 2021.
- Best Paper Honorable Mention VIS 2022.
- TrafficVis was used to curate a small labeled dataset that can be used for evaluation.

Chapter 5: TrafficBoard summarizes connections between pieces of evidence in a particular case to help practitioners kickstart their consideration or possible investigation of a potential case.

- Addressing Challenges:
  - After a formative study of 14 practitioners across 4 countries in North America and Europe (addr. *PC3: Legal),
  - TrafficBoard summarizes and redesigns standard graph-based layouts to help interpretability of HT evidence graphs (addr. *PC4: Interpret)
  - enabling users to more efficiently explore graph-based evidence data, which is error-prone (addr. *PC1: Data).
- Impact:
  - TrafficBoard is currently being integrated by *Marinus Analytics*.

A tabular representation of the thesis layout can be found below.

1 Introduction				
2 Background and Related Work				
Part I: Algorithms towards fighting HT	*PC1	*PC2	*PC3	*PC4
3 InfoShield: Detecting suspicious micro-clusters...	✓	✓	✓	✓
Part 2: Visualization for labeling and interpretation	*PC1	*PC2	*PC3	*PC4
4 TrafficVis: Interactive visualization for cluster labeling	✓	✓	✓	✓
5 TrafficBoard: Graph-based visualization...	✓		✓	✓
6 Discussion & Future Work				
7 Conclusion				

Table 1.1: Overview of the thesis.

Reproducibility. Each chapter has a link to the associated paper and public GitHub repository. We (and Marinus Analytics) keep the HT data private to protect the safety of victims and at-will workers, and additionally provide other public dataset alternatives for each project (i.e. Twitter data for InfoShield, synthetic data for TrafficVis).

Acknowledgements. This thesis is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE1745016 and DGE2140739. This work was also financed in part by a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development.

Chapter 2

Domain Information & Background

Important insights can be made by talking to investigators, survivors, criminologists, and other stakeholders that guide our algorithm and design decisions. We discuss the previous approaches they've taken to targeting HT online, their shortfalls, and how those approaches motivate our general approach.

2.1 Previous Attempts to Find HT

2.1.1 Practitioner's State of the Art: Manual Inspection

Previously, investigators would search for leads by navigating to an escort website and inspecting advertisements, one-by-one, for any possible signals of HT. Over time, practitioners have identified many "red flag" signals that associate an ad with HT risk by practitioners, such as "movement language" (i.e. "New in town") or client screening language (i.e. "Must be clean").

Although these signals are legitimately helpful to practitioners even today, they are not always reliable indicators of HT due to the adversarial nature of this problem. In particular, there are three main issues with relying on static signals alone.

1. **Signals change over time:** Traffickers constantly change their signals over time to avoid detection. For example, an ad containing movement language was found to be 70% less likely to be HT, despite movement language being commonly mentioned by practitioners as suspicious [78]. Due to the adversarial nature of catching HT, we cannot rely on these signals alone.
2. **Practitioners are at a disadvantage:** HT cases often take years to go through investigation and the legal system, during which time most evidence pertaining to that case will not be shared, even if that evidence points to new signals of HT. Even though new signals will eventually be shared with practitioners, those signals will be *years* old by that time, giving traffickers a huge advantage in avoiding detection.

3. **Legislation changes escort advertising patterns:** Signals for HT are largely affected by policy surrounding sex work and HT, which differ between countries and after large legislative changes within one country, such as *FOSTA-SESTA* passed in the United States in 2018.

Real-world example: FOSTA-SESTA

Since HT victims are posted alongside at-will sex workers on escort websites, the legality of sex work in a particular location shapes the escort ads from that region. For example, practitioners observe that at-will workers provide more details than traffickers in countries where sex work is decriminalized, making HT easier to spot than in places where sex work is illegal.

New legislation can also impact escort advertisements, such as the controversial *FOSTA-SESTA* (Allow States and Victims to Fight Online Sex Trafficking Act/Stop Enabling Sex Traffickers Act) passed in 2018 in the US, which held escort websites liable for HT content posted using their platforms. This led to the shutdown of the single-most popular escort website at the time, backpage.com, which investigators often used to catch traffickers.

While FOSTA-SESTA has been criticized for endangering at-will sex workers in the US, it also has made detection of HT much more difficult for investigators using those platforms, even prompting a letter from the US Department of Justice testifying that *FOSTA-SESTA* would make it increasingly difficult to investigate and prosecute trafficking cases [19]. Advertisements are now fragmented across many websites, each with their own posting rules and requirements, making it harder to connect ads coming from different websites, and unfortunately, much easier for traffickers to avoid detection.

2.1.2 Previous Tech-Based Solutions

Previous HT detection methods focused on advertisement-level classification rather than clustering [5, 40, 58, 100]. Many of these methods relied on specific indicators to mark ads as suspicious, such as keywords indicating underage victims. However, due to the adversarial nature of HT, predefined features will not stay relevant over time. Supervised methods used text and image data to predict the suspiciousness of an ad [100, 108] on a particular dataset, but supervised methods are not maintainable due to the number of labels required. Most problematically, these methods cannot find *groups* of organized activity, which is problematic for investigators – if a particular trafficker is being investigated, they need to discover all ads that are relevant to understand the scope of HT and quickly provide help to victims.

2.1.3 Major Insight from Accidentally-Found Case

One prosecuted case, which was found by chance through a benign misspelling, fundamentally changed how practitioners looked at escort advertisements. While manually looking on an escort site for suspicious signals, a practitioner noticed a benign but eye-catching misspelling

in an advertisement, similar to "hillo there" [1]. Many ads later, he noticed the same misspelling in another advertisement, then another. Upon a closer look, the practitioner realized that *the ads text was almost identical between these ads, even when advertising multiple different people.* This kickstarted an investigation leading to the successful prosecution of an international HT group including thousands of ads, all from a simple misspelling.

While the misspelling was eye-catching, it was a *side effect* of how traffickers generally operate. Globally, organized trafficking groups make up 74% of all HT cases and include 5–10 perpetrators and 10–12 victims, per average [104]. Additionally, traffickers write the advertisements in the majority of cases [86], resulting in a "template" ad that gets copied again and again with small perturbations describing the specific person, location, dates, etc. In this specific case, the traffickers' "template" just so happened to include a misspelling that propagated to thousands of ads, leading to their discovery, arrest, and successful prosecution.

Importantly, practitioners realized that some strong signals for HT cannot be found by looking at ads one-by-one, but instead by considering them *as a group*, motivating the design or adaptation of clustering algorithms, since advertisement grouping cannot be performed manually.

2.2 Participatory Design

Participatory design is a collaborative, user-centric approach where stakeholders, end-users, and algorithm designers collaborate throughout all stages of a project, from scoping a problem to the creation and validation of any proposed solution.

Our stakeholders include

- **Practitioners & Domain Experts:** includes the end users of our research, and more broadly, those with experience in finding HT online, including investigators, criminologists, and social workers.
- **HT Victims & Survivors:** includes the target population of our research (current victims). While we cannot directly consult HT victims, we instead speak with previous *survivors* of HT who not only help us understand the impact of our research, but also provide essential domain knowledge.

2.3 Minimum Description Length

The Minimum Description Length principle (MDL) [89] intuitively picks the best model among a set of models by balancing model complexity and descriptiveness through compression cost. More specifically, MDL considers the "cost" C of any input to be the number of bits required to fully (i.e. losslessly) describe or "write" that input. MDL states that the best model M for a dataset D minimizes the total cost $C(M, D)$ as described in Equation 2.1.

$$C(M, D) = C(M) + C(D|M) \tag{2.1}$$

[1] Investigators do not want to publicly disclose the actual phrase used, so "hillo there" is a placeholder.

The main insight is that MDL considers both the model cost $C(M)$, as well as the encoding of errors/deviations from the model $C(D|M)$, which can result in simpler but still high-performing models. Many methods don't consider model complexity, as it isn't an issue in all problem domains, but given *PC3: Legal and *PC4: Interpret, MDL is well-suited to our case.

2.4 Acronyms

Table 2.1 lists the acronyms used in this thesis, sorted in alphabetical order.

Acronym	Definition
ARI	Adjusted Rand Index
BP	Belief Propagation
ES	Early Stopping
MDL	Minimum Description Length
ML	Machine Learning
MSA	Multiple Sequence Alignment
POA	Partial Order Alignment

Table 2.1: Table of Acronyms

Part I

Algorithms towards fighting HT

Chapter 3

InfoShield: Detecting Suspicious Micro-Clusters in Escort Advertisements

This chapter is based on work published at *IEEE ICDE 2021* [PDF], *ACM TKDD 2023* [PDF].[a]

[a]These papers are joint work with Dr. Meng-Chil Lee, who performed Sections 3.3.2, and 3.4.2, included here for clarity.

3.1 Introduction

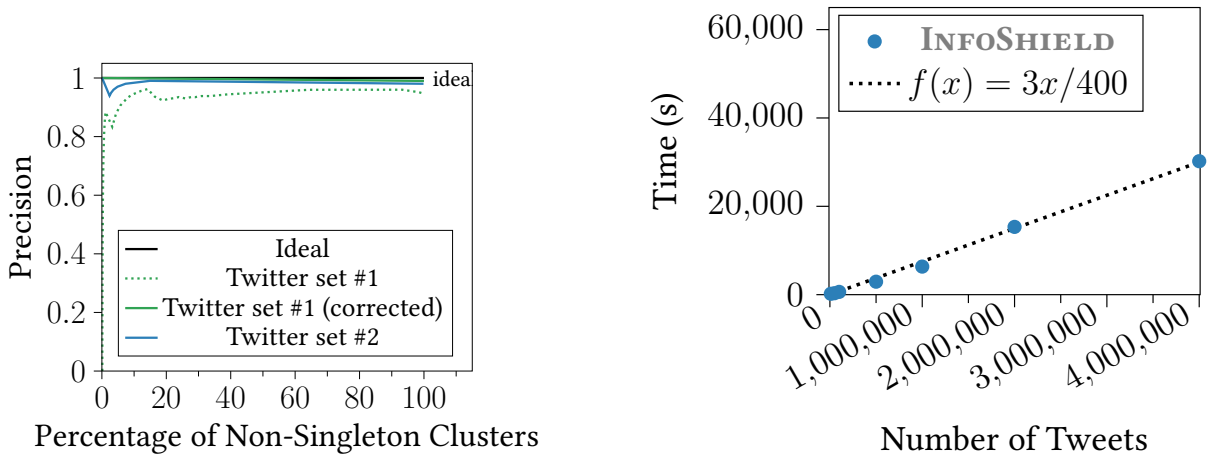
Given many advertisements, the majority of which do not belong to any cluster, how can we find small clusters of similarly phrased ads? While the driving application is human trafficking (HT) detection, finding similarly-phrased documents is a problem with numerous applications, such as search engines, plagiarism detection, mailing-address de-duplication, and more.

In this chapter, we describe InfoShield, a principled information-theory based method, then illustrate its efficacy not just towards its intended domain (HT detection), but also towards spambot detection in publicly-available Twitter data, so that others can see its results.

3.1.1 Application to the Human Trafficking Domain

As discussed in Chapter 2, our primary motivation is to find groups of related ads, in contrast to the ad-by-ad classification practitioners have been trying until now. Our goal is to find what we call *micro-clusters*, or small[1] groups of similarly phrased ads. Table 3.1 describes the expected M.O.s we expect to find in escort advertisements.

[1]Here, "small" is relative to the overall size of the data. While a particular micro-cluster might be "larger" than others, they will still be small relative to the overall number of ads, since most ads are not part of any organized behavior




```
Template #1:
   Hi gentlemen, Korea  super model just arrived...      *      specially selected...
-------------------------------------------------------------------------------------
1. Hi gentlemen, Korea  super model just arrived...Alma & Joan  specially selected...
2. Hi gentlemen, Korean super model just arrived...Paula & Miya specially selected...
3. Hi gentlemen, Korea  super model just arrived...Paula        specially selected...
4.    gentlemen, Korea  super model just arrived...Miya         specially selected...
```

Figure 3.1: *INFOSHIELD is effective on multiple domains:* (top left) precision@k on Twitter data is close to ideal, (top right) shows the scalability of INFOSHIELD over different data sizes, and (bottom) shows the interpretability of INFOSHIELD when applied to HT detection, finding *micro-clusters* of similar ads and visualizing slots (in red), i.e. portions of tweets that highly differ between otherwise duplicate documents.

INFOSHIELD helps practitioners save time by detecting micro-clusters of similar ads, grouping them, and summarizing the common parts, as shown in Figure 3.1, which depicts Twitter data[2].

M.O.	Description of advertised person
AT-WILL	an independent sex worker freely posting without coercion.
TRAFFICKING	an HT victim who is being forcibly posted, usually as part of a group.

Table 3.1: **Known M.O.s from experts**: Our goal is to separate these two behaviors. Ideally, we expect each micro-cluster to contain TRAFFICKING behavior and for AT-WILL behavior to not fall into any micro-cluster.

[2]We limit the number of escort ads shown for potential victim's safety.

3.1.2 Application to Twitter Bot Detection

Detection of organized activity also has a clear application to other CS research areas, such as spambot detection; given millions of tweets, most of which come from legitimate users, how can we find tweets that exhibit bot-like behavior? The simplest kind of bot behavior is spamming, i.e. posting tweets that are almost or exactly identical in text, to increase visibility. Bot detection has been well-studied, but the majority of algorithms use manually crafted features that are specific to certain platforms, for example, the number of retweets [31, 32]. Our goal is to find near-duplicates in any application, which includes social media platforms containing text, such as Twitter. This particular application benefits from a vast amount of publicly available data.

3.1.3 InfoShield: the Main Insights

Our first insight is to formalize the problem with information theory, leveraging the Minimum Description Length (MDL) principle to find good "templates" that serve as cluster representatives, automatically detecting which parts of the template differ for most ads, labeling them as "slots" (denoted in red in Figure 3.1 (bottom)). The resulting template can then be quickly shown to practitioners to inspect. InfoShield is *parameter-free*, since MDL can automatically pick the best choice of parameter values for any algorithm by choosing the combination with the shortest compression length. This is the InfoShield-fine part of our method.

The second insight is a novel pre-processing method, InfoShield-coarse, that improves scalability to be quasi-linear, by (a) eliminating single-copy documents/ads and (b) grouping the rest in coarse, but mainly homogeneous, clusters that can later be refined if needed.

The resulting algorithms, InfoShield and its time-evolving extension, DeltaShield, have a long list of desirable properties: They are

- **Practical**, being scalable and requiring no user-defined parameters thanks to the Minimum Description Length principle.
  - addr. *PC1: Dirty Data: no assumptions on text structure, just on similarity
  - addr. *PC2: Few/Expensive Labels: unsupervised algorithm
  - addr. *PC3: Legal/Financial Limits: scalable
- **Interpretable**, providing a clear visualization and summarization of the discovered micro-clusters
  - addr. *PC4: Expert interpretability: template is easy to interpret
- **Generalizable** and *domain independent* – we show results on two different application areas with organized text activity, Twitter spambot and HT detection.
  - addr. *PC1: Dirty Data: works on multiple languages (Spanish, English, Italian).
- **Incremental**, processing new batches of documents on-the-fly without recomputing on historical documents
  - addr. *PC3: Legal/Financial Limits: limited computational resources aren't used to recompute micro-clusters on historical advertisements with outdated information.

Reproducibility: Our code is open-sourced at https://github.com/catvajiac/InfoShield-Incremental. The HT dataset is available to researchers after NDA (email Dr. Cara Jones cara@marinusanalytics.com). The Twitter datasets are publicly available (see [30]).

3.2 Background and Related Work

There is a lot of work on HT detection, document clustering, and multiple sequence alignment, and we group it in the following sub-sections.

3.2.1 Human Trafficking Detection

Some previous works try to classify whether or not a particular advertisement is suspected of HT [4, 41, 58, 101]. For instance, HTDN [101] proposes a supervised deep multimodal model trained on 10K manually labeled ads. Unfortunately, due to the adversarial nature of escort advertisements, these predefined or learned features don't stay relevant over time. These labeled ads are also expensive to obtain (requiring the precious time of domain experts) and are error-prone, as will be discussed in Section 3.6. Moreover, inspecting ads individually, we might overlook ads that are part of an organized activity but do not stand out on their own. Therefore, unsupervised algorithms that find connections between ads [75, 87, 88] and *groups* of organized activity are preferred in this domain [67]. In particular, Template Matching [67] proposes the first anti HT method to our knowledge to perform clustering. However, the interpretability of clusters is limited, and the algorithm isn't scalable.

3.2.2 Social Media Bot Detection

Most efforts in detecting bots in social media platforms are formulated as supervised classification based on features from users and the content they post [96, 111]. Fewer works look for anomalies or fraud in networks, rather than in text, for instance [95]. A notable method, Botometer [32], formerly called BotOrNot, is an online service that provides a score of likelihood that a particular user is a bot. Since it is the only state-of-the-art method with public access to the implementation, we will use it as a baseline for our experiments in Section 3.6. [30] gives a more comprehensive overview of Twitter bot detection methods, and also provides the dataset we will use in Section 3.6.1.1. Very few works focus on detecting *organized activity* - groups working together to mislead people about who they are and what they are doing, which is a rising issue [46]. ND-Sync [45] finds a related but different type of behavior, i.e. "retweet spam", where groups of multiple users exhibit organized behavior by consistently upvoting a particular user's tweets.

3.2.3 Document Embedding and Clustering

Much work has been done to represent documents in a machine-understandable format. The most widely-used approaches to represent documents include bag of words [50] and term frequency-inverse document frequency (tf-idf) [56]. These methods are commonly used for

plagiarism detection [20, 39, 57, 70], which is a similar setting to near-duplicate detection. However, none of these methods do visualization or ranking, and some assumptions do not work in our case, i.e. [20] assumes documents consist of multiple lines, which is not the case for tweets or the majority of escort advertisements.

Unsupervised word vector models such as Word2Vec [74], Doc2Vec [63], and FastText [17] assume that words occurring in the same context tend to have similar meaning, with much success. However, these methods require large amounts of time and data to train. Even when trained using large datasets from Twitter data and the HT domain, we find that these generic embedding methods do not perform as well, as shown in Section 3.6.

BERT [34] is another successful language model, but through experiments on the Trafficking10k dataset, we find it does not perform well on escort ad text[61], due to the sheer number of misspellings, shortenings, and specific escort keywords not found in normal text. Instead, we take the approach of developing a lighter-weight solution that naturally handles the small amount of labeled data.

Large language models (LLMs) such as ChatGPT and Gemini are being applied to a variety of problems. Our application to the HT domain has so far been limited; many models have failsafes against explicit content found in escort ads. While we used early versions of ChatGPT to generate a synthetic version of HT data [77], at the time of writing this thesis, ChatTGPT will no longer generate ads using the same prompts as before.

Given any document embedding, we can choose from many clustering algorithms. Density-based clustering techniques are most relevant to finding small dense text clusters, such as DBSCAN [42], HDBSCAN [72], OPTICS [6], or k-means [35]. These are all powerful methods, but none of them do slot-detection. We compare InfoShield to HDBSCAN as part of our curated baseline for HT detection, see Section 3.6 for more details.

In Table 3.2, we give several question-marks for clustering methods because some of the methods are scalable (k-means), while others are almost quadratic; some methods are parameter-free (G-means), but most are not.

Finding pairs of nearby points (or intersecting rectangles) is an old problem, under the name of "spatial joins" [21, 69]. However, these methods are best for low-dimensional spaces, since they use the R-tree [48] spatial access method.

3.2.4 Multiple-Sequence Alignment

Multiple-Sequence Alignment (MSA) is a well-studied area with an application to biology, for comparing DNA sequences. The Barton-Sternberg algorithm [8] is an early profile-based approach which aligns sequences by updating a profile sequence iteratively. However, profile-based approaches generate ambiguity among sequences. To solve this, [64] uses partial order graphs instead of profile sequences, which enables a base in dynamic programming to have multiple predecessors and successors.

Nature Language Processing (NLP) is another area benefiting from MSA. [9] applies MSA to learn the patterns of given word sequences by word lattices and rewrite the sentences. [97] focuses on aligning sentences by syntactic features to create the description for a particular fact. However, most of these methods highly rely on parameter tuning and English syntactical rules, assuming that all sentences are grammatically correct. This assumption does not hold for

data on any social network or for escort advertisements, where misspellings and grammatical errors are common. Thus, these methods are not generalizable.

3.2.5 Minimum Description Length

The Minimum Description Length principle (MDL) [89] assumes that the best model $M \in \mathbb{M}$ for data D minimizes $C(M) + C(D|M)$, where $C(x)$ is defined as the cost, i.e. number of bits, needed to describe x losslessly. The main insight is that it penalizes both the model cost $C(M)$, as well as the encoding of errors/deviations from the model $C(D|M)$ - while several other methods ignore the model complexity.

MDL has been extremely successful in several data mining tasks [47], including decision trees (SLIQ [73]), graph mining (CrossAssociations [26]), time series segmentation and mining (AutoPlait [71]), string similarity [59], and many more applications. It formalizes the very intuitive "Occam's razor" idea: the simplest explanation for a phenomenon or dataset is the best explanation.

While all of the above methods have provided unique and interesting contributions, none have all of the same features as InfoShield. Table 3.2 contrasts InfoShield against the state of the art competitors. The algorithms in Sections 3.3 and 3.4 appeared in the conference version of this work [65], while the journal version added the incremental algorithm, DeltaShield (Section 3.5) and additional experiments (Section 3.6).

3.3 Proposed Method - Theory

3.3.1 Intuition and Theory

Our problem is split in the following parts: given N documents, where we suspect that there are microclusters of organized activity:

1. Theory: how do we measure the goodness of a set of clusters, and
2. Algorithms: how do we quickly find clusters that describe patterns in the data concisely (InfoShield-coarse– Section 3.4.1) and then how do we refine these clusters (InfoShield-fine– Section 3.4.2).

To explain our MDL-based method, we introduce an example set of documents (in this case, escort advertisements) in Figure 3.2a.What would be the ideal way to summarize these documents? Intuitively, we see that documents 1–4 and 5–7 should fall into their own clusters, and document 8 is unlike any other in the corpus. This desired output is shown in Figure 3.2b(b).

One part of our proposed InfoShield is to use *templates*, which consist of constant strings and variable strings, called slots. We depict slots with '*', following the Unix convention. We also allow the usual string-editing operations (insertions, deletions, and substitutions). Looking at Documents 1–4 in Figure 3.2a, a human (and our InfoShield) would produce the template:

"**Hi gentlemen, Korea super model just arrived... * specially selected...**"

as shown in Figure 3.2b.

Property \ Method	Clustering [6, 35, 42, 72]	Barzilay and Lee [9]	Shen et al. [97]	HDTN [101]	Template Matching [67]	InfoShield	DeltaShield
Practical- Scalable	?			✓	✓	✔	✔
Practical- Effective	?	?	?	?	✓	✔	✔
Practical- Ranked output	?			?	✓	✔	✔
Parameter-free	?					✔	✔
Principled						✔	✔
Interpretable	?	✓	✓		✓	✔	✔
Slot Detection		✓				✔	✔
Generalizable	✓					✔	✔
Incremental							✔

Table 3.2: ***InfoShield*** *and* ***DeltaShield*** *satisfy our goals*, while competitors miss one or more of the features. "?" means that it does not always satisfy the condition or that it is unclear from the original paper.

```
Documents
-----------------------------------------------------------------------------------------
1. Hi gentlemen, Korea super model just arrived...Alma & Joan specially selected...
2. Hi gentlemen, Korean super model just arrived...Paula & Miya specially selected...
3. Hi gentlemen, Korea super model just arrived...Paula specially selected...
4. Gentlemen, Korea super model just arrived...Miya specially selected...
5. Outcall only...Anne available, call 123.456.7890
6. Outcall only...Bella available, call 234.567.8900
7. Outcall only...Alexa available, call 345.678.9000
8. Fully independent, contact me at 987-654-3210
```

(a) **Example Input:** An example set of documents, some of which are clearly related to each other. Note: documents 1–4 are sanitized versions of actual escort advertisements.

```
Template #1:
   Hi gentlemen, Korea  super model just arrived...      *      specially selected...
-----------------------------------------------------------------------------------------
1. Hi gentlemen, Korea  super model just arrived...Alma & Joan  specially selected...
2. Hi gentlemen, Korean super model just arrived...Paula & Miya specially selected...
3. Hi gentlemen, Korea  super model just arrived...Paula        specially selected...
4.    gentlemen, Korea  super model just arrived...Miya         specially selected...

Template #2:
   Outcall only...      *     available, call       *
-----------------------------------------------------------
5. Outcall only...Anne  available, call         123.456.7890
6. Outcall only...Bella available, call or text 234.567.8901
7. Outcall only...Alexa available, call         345.678.9012
```

(b) **Desired Output:** documents organized into *microclusters*, with each microcluster containing a *template* describing the ads it contains.

Figure 3.2: Illustrative Example: (a) example corpus of advertisements, (b) ads grouped into two micro-clusters and visualized using a "template", with any individual ad deviations highlighted.

Now let's consider documents 5–8. Intuitively, we observe that documents 5–7 should be in a second micro-cluster. The expected output for InfoShield should now be two templates T_1 and T_2, with T_1 representing Doc #1-4, T_2 representing Doc #5-7, and Doc #8 does not belong to any template, as shown in Figure 3.2b.

In more detail, but still informal, InfoShield should achieve lossless compression, with the cost being as follows:

1. *Model complexity* $C(M)$: the cost to encode the t templates we discover. In our working example, this would be the coding cost (roughly, the number of characters, below), for
   T_1: "Hi gentlemen, Korea super model just arrived... * specially selected...
   T_2: "I made 30K working * - call * or visit *"
2. *Data compression* $C(D|M)$: the cost to encode slot-values, insertions, and deletions, for each of the documents, with respect to its best template (or just the listing of the words in the document, if no template matches). Thus, for each document, we must store (a) the tokens in slots, (b) position and token for insertions, (c) position for deletions, (d) position and token for substitutions, and (e) the template-id that best matches the document. Table 3.3 shows the information we include in $C(D|M)$ for our running example.

Doc	Temp.	Slots	Ins.	Del.	Sub.
#1	T_1	{"Alma and Joan"}			
#2	T_1	{"Paula and Miya"}			
#3	T_1	{"Paula"}			3: "Korean"
#4	T_1	{"Miya"}		1	
#5	T_2	{"Anne", "123.456.7890"}			
#6	T_2	{"Bella", "234.567.8901"}			
#7	T_2	{"Alexa", "345.678.9012"}			
#8	*N/A*	"Fully independent, contact me at 987-654-3210"			

Table 3.3: Example encoding for $C(D|M)$

Notice that Docs #1-4 are compressed with much fewer characters when we use template T_1, since they have so many phrases in common.

The coding cost is roughly proportional to the number of characters we need to describe (1) and (2) above. More formally,

Definition 1 (Total encoding cost)**.** *The total coding cost for a set of* n *documents with* t *templates is given by*

$$C = C(M) + C(D|M) \tag{3.1}$$

In Section 3.3.2, we explain the exact cost for N documents and t templates more precisely. Then, in Section 3.4, we propose algorithms on how to *discover* such a good set of templates.

We want to highlight that the separation of the cost function in Equation 3.1 from the algorithms makes InfoShield extensible: we can use any and every optimization algorithm we want. The ones we propose in Section 3.4 are carefully thought-out, and give meaningful

results, but any other set of algorithms is fine to include – we can pick the solution with the best coding cost.

Furthermore, InfoShield is parameter-free: any optimization algorithm minimizing total cost does not need user-defined parameters — we can try as many parameter values as we want, and pick the solution with the lowest cost.

3.3.2 Data Compression and Summarization

In this subsection, we give the details of the encoding cost in InfoShield. Table 3.4 provides symbols and definitions relevant to the encoding.

Symbols	Definitions
N	Total number of documents in D
t	Total number of templates
V	Number of words in vocabulary
T_i	i-th template
l_i	Length of template T_i
s_i	Number of slots in T_i
$\hat{l}_d$	Alignment length of data d
$w_{d,j}$	Number of words in j-th slot in aligned data d
e_d	Number of unmatched words in aligned data d
u_d	Number of substituted/inserted words in aligned data d
$\langle n \rangle$	$\approx 2 \lg n + 1$: universal code length for a non-negative integer
$\lg(L)$	$= \log_2(L)$: code length for integer i ($1 \leq i \leq L$)

Table 3.4: Symbols and definitions for InfoShield-fine

3.3.2.1 Template Encoding

We use the notation $\langle n \rangle$ for the coding cost of integer n, using the universal code length [90], that is $\langle n \rangle = \log^* n \approx 2 \times \lg n + 1$.

We also assume that we have V vocabulary words total and that each is encoded as an index, requiring $\lceil \lg V \rceil$ bits. For a length-l document, we need $\langle l \rangle$ bits to encode the number of words and $\lg V$ for each word, resulting the total cost $\langle l \rangle + l * \lg V$.

Definition 2 (Model encoding cost)**.** *The coding cost for t templates is given by*

$$C(M) = \langle t \rangle + \sum_{i=1}^{t} \langle l_i \rangle + l_i \lg V + (1 + s_i) \lg l_i \tag{3.2}$$

Let's describe every term of the above definition:

- $\langle t \rangle$ - universal coding, for the number of templates T
- For each template T_i, we need:
  - $\langle l_i \rangle$ to encode the number of words in the i-th template

- $\lg V$ for each word in T_i
- $\lg l_i$ for the number of slots s_i in the template, and
- $\lg l_i$ for the location of each slot.

Arithmetic Example 1. *The encoding cost for a single template T_i with 10 tokens and 2 slots is:*

$$\langle 10 \rangle + 10 \lg V + 3 \lg 10$$

3.3.2.2 Alignment Encoding

Given a template and a document that it describes, what is the best way to encode the document? The intuition is to encode insertions, deletions, and substitutions in the template, and the tokens in slots. For the templates, we need only encode the word-location of a mismatch, its type, and, for insertion/substitution, we encode the relevant word.

Definition 3 (Data encoding cost)**.** *The coding cost for N documents encoded with t templates is given by*

$$\begin{aligned} C(D|M) = N &+ l_d \times \lg V \\ &+ \sum_{i=1}^{t} \sum_{d \in D_i} (\lg t + \left\langle \hat{l}_d \right\rangle + \hat{l}_d \\ &+ e_d \lg \hat{l}_d + u_d \lg V + \sum_{j=1}^{s_i} \mathcal{S}(w_{d,j})), \end{aligned} \tag{3.3}$$

where D_i denotes the data encoded by template T_i. We describe this definition in more detail: Let D_U denotes the documents that do not match any template. The encoding cost for data $d \in D_U$ which is not encoded by template is simply computed by $l_d \times \lg V$. For the rest, the reasoning is as follows: Given a template T_i and a document $d \in D_i$, the alignment coding cost is:

- 1 bit for template flag yes/no
- $\lg t$ for template-id (if the flag is 'yes'):
- $\left\langle \hat{l}_d \right\rangle$ for length of the alignment
- 1 bit for each word in alignment if matched/unmatched
- $\lg \hat{l}_d$ for the location of each unmatched word
- $\lceil \lg 3 \rceil = 2$ bits for operation type of each unmatched word (insertion/deletion/substitution)
- $\lg V$ for word index in vocabulary if insertion/substitution
- $\mathcal{S}(w_{d,j})$ for the number of words $w_{d,j}$ in j-th slot:

$$\mathcal{S}(w_{d,j}) = 1 + \begin{cases} \langle w_{d,j} \rangle + w_{d,j} \lg V & \text{, if } w_{d,j} > 0 \\ 0 & \text{, otherwise} \end{cases} \tag{3.4}$$

- repeat, for all other editing operations

Arithmetic Example 2. *The alignment encoding cost of Doc #4 by template T_1 (see Figure 3.2b, is the following:*

$$\begin{aligned} &\lg 2 + \langle 14 \rangle + 14 \\ &\quad + 3 \lg 14 + 2 \lg V + 2 \times (1 + \langle 1 \rangle + 1 \lg V) \end{aligned} \tag{3.5}$$

3.3.2.3 Overall Encoding

Notice that we ignored the cost of encoding the vocabulary, since it would be the same for all sets of templates, and roughly the number of bytes to spell out all the vocabulary words, separated by a word-delimiter, such as a newline character. More accurately, this would be: $\langle V \rangle + V \times (l+1) \times 8$ where l is the average word length, 8 bits per character, and 1 bit for the delimiter between words.

3.4 Proposed Method - Algorithms

How can we find templates that minimize our cost function in a scalable way? While the intuition described in Section 3.3 is correct, finding such templates is an expensive operation, being quadratic in the worst case. Thus, we first create reasonable clusters of related documents in a scalable way, using InfoShield-coarse, then work to find templates within each cluster using InfoShield-fine. If the average cluster size remains small, in comparison to N, then we process N documents in sub-quadratic time.

3.4.1 InfoShield-coarse

How do we quickly create coarse-grained clusters of documents with high text similarity? We start with document embedding, then perform clustering.

3.4.1.1 Document Embeddings

How do we generate a meaningful document embedding? We wish to capture similarity between documents that contain similar phrasing, but may have small variations (insertions, deletions, misspellings, etc). To this end, we first calculate the tf-idf weights for each phrase (n-gram)-document pair in the corpus. When calculating tf-idf, we consider phrases up to n-grams, with $n = 5$. [3]

Then, for each document, we extract the top phrases with the highest tf-idf scores. By using tf-idf and limiting the number of phrases used, we only keep the most important phrases in the document that are unique to only a few advertisements, while ignoring commonly-used phrases. By making the number of phrases selected a function of input size, we reduce the risk of our results being heavily impacted by document length. Since some documents have a maximum length (i.e. tweets) but many do not, this helps to prevent InfoShield-coarse from being domain-specific.

3.4.1.2 Clustering

Now, how do we quickly create meaningful candidate clusters? We construct a bi-partite graph of documents and phrases. For any document i and phrase j, we construct an edge i, j if j is a top phrase in i. A pictoral example of this can be found in Figure 3.3. Once all documents are

[3] Phrase length has little impact on results past $n = 5$: see Section 3.6.6.

processed sequentially, we consider all connected components in G to be our coarse-grained clusters.

Figure 3.3: *InfoShield-coarse in action:* processing each document sequentially, we extract the top phrases according to tf-idf, then analyze the connected components as coarse-grained clusters.

In the case that these clusters end up too large (due to an "unimportant" phrase that combined documents that ideally should not be combined), we rely on InfoShield-fine to refine these clusters and split them if necessary. This is why InfoShield-coarse is very permissive, only requiring ads to share one important phrase to be connected.

Algorithm 3.1 shows more formally how to construct a document graph using InfoShield-coarse.

Algorithm 3.1: InfoShield-coarse

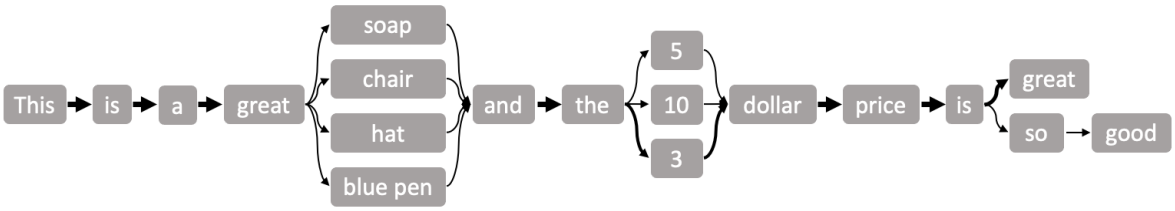
Data: N documents
Result: candidate clusters generated from N
initialize empty document-phrase graph $G = (V_1, V_2, E)$;
forall *documents* d **do**
  forall *phrase* p *in FindTfidfPhrase*(d) **do**
    $E \leftarrow E \cup (d, p)$;
  end
end
clusters $\longleftarrow$ FindConnectedComponents(G) ;

3.4.2 InfoShield-fine

Once we have coarse-grained clusters, how do we find templates and visualize the resulting clusters? Given data D containing multiple documents, split into coarse-grained clusters, the goal is to automatically find a template set M containing zero or more templates. Each template is expected to encode at least two documents. Within each coarse-grained cluster, the first task is to generate non-singleton candidate sets of documents and find potential templates. Next, we search for the best consensus document, i.e. the document that most represents the cluster, and detect possible slots by optimizing our cost function in Equation 3.3. We continue finding templates until we've processed all documents in a coarse-grained cluster, then move to the next cluster. We divide our algorithm into three major steps as follows:

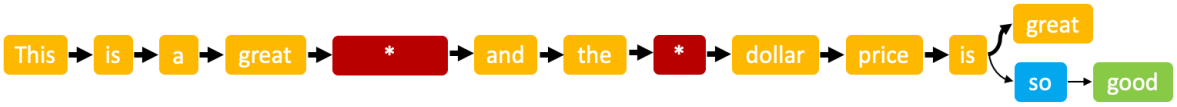
1. **Candidate Alignment:** Identify the candidate set for a template and align all the documents in the set, using multiple sequence alignment (MSA).
2. **Consensus Search:** Search for the best consensus document in the alignment.
3. **Slot Detection:** Detect slots in the consensus document to generate a template.

Let's take the first template from Table 3.2b as an example. We show a visual representation of what each step does in Figure 3.4.



(a) Step 1: Candidate Alignment

(b) Step 2: Consensus Search



(c) Step 3: Slot Detection

Figure 3.4: *Example pipeline of InfoShield-fine:* Here we show the output after each step of InfoShield-fine.

To compute the MSA, we carefully choose to use Partial Order Alignment (POA) [64] as our alignment method for its effectiveness and efficiency. It is worth noting that InfoShield-fine can co-work with any off-the-shelf MSA approaches.

3.4.2.1 Candidate Alignment

Given data D from one cluster generated by InfoShield-coarse, containing multiple documents at iteration i, the candidate set for the template needs to be identified first. We first align all the documents $d \in D$ with the first document d_1 individually and then compute the cost $C(d|d_1)$ and $C(d)$ for every $d \in D$; if $C(d|d_1)$ is smaller than $C(d)$, meaning that d and d_1 have high similarity and can possibly be encoded by the same template, we add d into the set D_i containing all similar documents found in iteration i. Finally, we generate the alignment A_i by the POA method with all documents in D_i.

3.4.2.2 Consensus Search

After generating alignment A_i, how do we decide which tokens are part of the template, and which are insertions/deletions/substitutions? Keeping too many words in the template causes more unmatched operations (insertion/deletion/substitution); while keeping too few words hurts interpretability.

To solve this problem automatically, we turn it into an optimization problem by MDL. Function $Sel(A_i, h)$ is used to select the sub-alignment from the POA graph, where we only keep edges between words that occur more than h times in A_i. We aim to search for the best threshold h_i^* to generate the consensus of alignment with the lowest cost. The optimization problem can then be formed as follows:

$$h_i^* = \min_h C(D_i|Sel(A_i, h)) \tag{3.6}$$

Although our cost function is not convex, the optimization problem is only 1-dimensional, being relatively easy to solve. Hence we employ the Dichotomous Search algorithm [29] as our optimization method, where it returns the optimal solutions in most cases. The optimization algorithm is shown in Algorithm 3.2, where we iteratively shrink the search space to half. The consensus document $T_i^{'}$ only contains one sequence and no slot.

3.4.2.3 Slot Detection

Once we have a template, how do we find slots? Slots contain parts of documents which we expect to differ, either in length or content, in the same location of each document. Slots inherently differ from unmatched words; instead of storing the location of each unmatched word per document as we would for unmatched words, we only store the location once, as part of the template.

Algorithm 3.3 shows how we do slot detection. We first recognize the operation types of words by each alignment $a \in A_i$, which are either insertions or substitutions. We identify which words each potential slot p contains in the given consensus document $T_i^{'}$. With this information, the computation of total cost with or without the slot p can easily be done. We only keep slots that decrease the total cost and store them in T_i.

Algorithm 3.2: *Consensus-Search*

Data: An alignment A_i and a candidate set D_i
Result: A consensus document $T_i^{'}$
Initialize $h_L = 0, h_R = |D_i| - 1$;
while $h_L < h_R$ **do**
  $h_M \leftarrow (h_L + h_R)/2$;
  if $C(D_i|Sel(A_i, h_M - 1)) \leq C(D_i|Sel(A_i, h_M + 1))$ **then**
    $h_R \leftarrow h_M - 1$;
  else
    $h_L \leftarrow h_M + 1$;
  end
end
$T_i^{'} \leftarrow Sel(A_i, h_M)$;
Return $T_i^{'}$;

Algorithm 3.3: *Slot-Detection*

Data: A consensus document $T_i^{'}$, an alignment A_i, and a candidate set D_i
Result: A template graph T_i with slot(s)
Initialize P as a dictionary, $T_i = T_i^{'}$;
for $a \in A_i$ **do**
  $x = 0$;
  for $j = 1, .., l_a$ **do**
    if a_j *is an insert or substitution word* **then**
      $P[x] \leftarrow P[x] + 1$;
    else
      /* a_j is a matched or deleted word */
      $x \leftarrow x + 1$;
    end
  end
end
for $p \in P$ **do**
  if $C(D_i|T_i^{'}(p.slot \leftarrow True)) < C(D_i|T_i^{'})$ **then**
    $T_i \leftarrow T_i(p.slot \leftarrow True)$;
  end
end
Return T_i;

3.4.2.4 Relative Length

To study the quality of compression by InfoShield-fine, we use relative length:

$$\text{Relative Length} = \frac{\text{Cost after compression}}{\text{Cost before compression}} \tag{3.7}$$

When relative length is close to 1, it means that the quality of compression is low; when it is close to lower bound, it means that the quality of compression is high, and the compressed documents are near-duplicate. For that reason, we derive the lower bound encoding cost of a cluster to study whether it is close to near-duplicate or not.

Lemma 1. *The lower bound encoding cost of a cluster by InfoShield-fine is*

$$\frac{t}{n} + \frac{1}{\lg V} \tag{3.8}$$

where t denotes the number of templates in the cluster, n denotes the number of documents in the cluster, and V denotes the number of words in vocabulary.

Proof. The encoding cost of n documents without template is $nl \lg V$. By Equation 3.2, we know that the encoding cost of t templates is $\langle t \rangle + t(\langle l \rangle + l \lg V + \lg l)$; and by Equation 3.3, we know that the encoding cost for each document with no unmatched words is $(1 + \langle l \rangle + l)$. We can then derive:

$$\begin{aligned} &\frac{\langle t \rangle + t(\langle l \rangle + l \lg V + \lg l) + n(1 + \langle l \rangle + l)}{nl \lg V} \\ &\approx \frac{t \lg V + nl}{n \lg V} \approx \frac{t}{n} + \frac{1}{\lg V} \end{aligned} \tag{3.9}$$

where l is a small constant value that is negligible. So the total encoding cost for n near-duplicate documents by t templates is approximately $\frac{t}{n} + \frac{1}{\lg V}$. □

3.4.2.5 Overall Algorithm

The overall algorithm of InfoShield-fine is shown in Algorithm 3.4. Given data D containing multiple documents from one cluster by InfoShield-coarse, we first initialize the template set $\mathcal{T}$ and the number of detected template i. At iteration i, we initialize alignment by the first document $d_0 \in D$. We compare with all other documents $d \in D$ to identify whether they should be encoded by the same template. After generating the alignment A_i and the data D_i that it encodes, we search for the best consensus sequence $T_i^{'}$ by optimizing the cost function. Then we detect the slots on the consensus sequence $T_i^{'}$ to generate template T_i. We include the T_i into our template set $\mathcal{T}$, and compute the total cost for both templates and data encoded by templates. If the total cost decreases by including T_i, we include it into $\mathcal{T}$ and update the total cost; otherwise, we treat D_i as noise. We run InfoShield-fine on every cluster generated by InfoShield-coarse, thus our final model M is $\mathcal{T}_1 \cup \mathcal{T}_2 \cup \cdots \cup \mathcal{T}_m$, where m is the number of coarse clusters. It is worth noting again that InfoShield-fine is parameter-free, needing no human-defined parameters and optimizing for each template automatically.

Algorithm 3.4: InfoShield-fine

Data: Data D consisting of multiple documents
Result: A template set $\mathcal{T}$
Initialize $\mathcal{T}, c^* = C(D), i = 1$;
while $|D| > 0$ **do**
  Initialize $A_i = d_1$ by the first document in D;
  Initialize candidate set D_i;
  for $d \in D[2:]$ **do**
    if $C(d|d_1) < C(d)$ **then**
      $D_i \leftarrow D_i \cup \{d\}$;
      $A_i \leftarrow MSA(A_i, d)$;
    end
  end
  $T_i^{'} \leftarrow ConsensusSearch(A_i, D_i)$;
  $T_i \leftarrow SlotDetection(T_i^{'}, A_i, D_i)$;
  $c \leftarrow C(\mathcal{T} \cup \{T_i\}) + C(D|\mathcal{T} \cup \{T_i\})$;
  if $c < c^*$ **then**
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{T_i\}$;
    $c^* \leftarrow c, i \leftarrow i + 1$;
  else
    Treat D_i as noise(s);
  end
  $D \leftarrow D \backslash D_i$
end
Return $\mathcal{T}$;

3.4.3 Complexity Analysis

Lemma 2. *InfoShield is quasi-linear on the input size, taking time*

$$O(Ncl) + O(k_{max}Nlog(N)l^2) \tag{3.10}$$

where N is the number of documents, l is the (maximum) length of a document, m is the number of coarse clusters, c is the maximum number of non-duplicate documents in a cluster, and k_{max} is the maximum number of templates in a coarse-grained cluster.

Proof. We analyze the runtimes of InfoShield-coarse and InfoShield-fine separately. For InfoShield-coarse, we iterate through N documents, picking the top 10% of phrases in N, and adding edges between these documents and phrases. Thus the runtime of InfoShield-coarse is $O(Nl)$, where l is the average length of the documents.

In InfoShield-fine, there are a total of k iterations, where k is the maximum number of templates generated from the given data. With the help of vectorization, MSA can be done in $O(l^2)$. For each iteration, *Consensus-Search* requires $O(\log S' \times S' l^2)$ time, where S' is the average number of documents being aligned in each template; and *Slot-Detection* requires $S' l^2$ time. The time complexity of *Candidate-Alignment* in each iteration is $O(Sl^2)$, where $S \geq S'$ is the average number of documents in the each cluster. Thus the time complexity of InfoShield-fine is $O(\sum_{i=1}^{m} k_i S_i \log(S_i) l^2)$, which is upper-bounded by $O(k_{max} N \log(N) l^2)$, and where where m is the number of coarse clusters generated by InfoShield-coarse, k_{max} is the maximum number of templates generated by a cluster.

In total, the algorithm takes time $O(Nl) + O(k_{max}Nlog(N)l^2)$ time.

In practice, $k_{max} \leq 2$ in the Twitter datasets. Furthermore, the value of c will be quite low, since Twitter spambots post many duplicate tweets, which will make the runtime fast. Empirical evidence of this can be found in Figure 3.5, where we see that InfoShield-coarse scales linearly with input size. For the use cases presented in this chapter, i.e. escort advertisements and tweets, we also note that l is bounded (280 for Tweets). □

3.5 Proposed Method - Incremental

In order to do HT detection in practice, we must develop an algorithm that can process documents incrementally. Domain experts have hundreds of millions of ads and keep crawling additional ones each day. If we have already grouped historical ads into t clusters, we want to process a batch of newly-crawled documents without recomputing on historical documents. Here we'll discuss the necessary modifications to InfoShield and present DeltaShield, with relevant experiments in Section 3.6.

3.5.1 DeltaShield-coarse

How can we modify InfoShield-coarse to be conducive to an online setting? We consider a setting where batches of documents come in during an aggregated time period, i.e. daily or weekly. Most of the algorithm can be adopted with minimal changes; since InfoShield-coarse incrementally adds to the document-term graph, we can process an entire batch, send

the results to InfoShield-fine, and then continue processing documents as they arrive. The biggest challenge we have is in computing the tf-idf score of n-grams in a given document before seeing the entire corpus. To this end, we approximate the tf-idf score by computing the idf only on *the documents seen so far*, rather than the entire corpus. This approach is advantageous for two reasons: (1) as we process more and more documents, the approximate tf-idf score of a given n-gram will approach its actual tf-idf score, as verified empirically in Section 3.6.7, and (2) for the HT application, domain experts have a lot of historical, in-actionable data that can be processed first to improve the approximate tf-idf score.

3.5.2 DeltaShield-fine

In an online setting, we still need to generate new templates if needed, so Algorithm 3.4 will be performed in every batch. Moreover, a preprocessing step and an updating step must be included to keep DeltaShield efficient and effective.

3.5.2.1 Preprocess

We propose Algorithm 3.5 as a preprocessing step before trying to generate a new template in Algorithm 3.4. If we were able to process all documents at once, the intuitive solution is to go through all the documents and generate templates. However, in an online setting, we often see documents from any one template span over multiple batches. To this end, the preprocessing step tries to encode an incoming document by all existing templates in its coarse cluster and select the template with the lowest encoding cost. If the cost by the selected template is lower than the encoding cost of the document itself, we consider that the document belongs to that template.

Unfortunately, the time complexity of examining all the existing templates is $O(k_{max}l^2)$. If a coarse cluster has a large number of templates, we will incur a large overhead. To address this, we adopt an early-stopping (ES) mechanism in Algorithm 3.6. Instead of sequentially investigating all templates in a coarse cluster, we order the templates by the lengths of intersection between unigrams in the incoming document and each template. Then, we select the first template that lowers the encoding cost of the document.

Lemma 3. *The time complexity of naive preprocessing step is*

$$O(k_{max}l^2) \tag{3.11}$$

but can be reduced by ES mechanism to

$$O(l^2 + k_{max}l) \tag{3.12}$$

where l is the (maximum) length of a document, and k_{max} is the maximum number of templates in a coarse-grained cluster.

Proof. To compute the cost after compression, we calculate the alignment, which takes $O(l^2)$. To search for the template with lowest encoding length, we examine $O(k_{max})$ templates. In total, the naive preprocessing step takes $O(k_{max}l^2)$ time to find the template with lowest encoding cost.

Next we analyze the ES mechanism. The time complexity of computing the lengths of intersection for one template is $O(l)$. It takes $O(k_{max}l)$ to compute all the lengths for all templates in the coarse cluster. The total time complexity is then reduced to $cl^2 + k_{max}l$, where c denotes the number of templates examined. However, c is a small number in most cases (close to 1), which is negligible, so the final time complexity is $l^2 + k_{max}l$. □

Later in Section 3.6.7.2, we will demonstrate that the ES mechanism largely improves the efficiency while achieving comparable effectiveness.

Algorithm 3.5: *Preprocess-Naive*

Data: An incoming document d_1, and a template set $\mathcal{T}$
Result: An updated template set $\mathcal{T}$ or False if no appropriate template
/* Examine all the templates */
$i^* = \arg\min_{T_i \in \mathcal{T}} C(d_1|T_i)$;
if $C(d_1|T_{i^*}) < C(d_1)$ **then**
    $D_{i^*} \leftarrow D_{i^*} \cup d_i$;
    $T_{i^*}.added \leftarrow True$;
    /* Find appropriate template -> Continue with the next incoming document */
    Return $\mathcal{T}$;
end
/* No appropriate template -> Used as an initial document to generate the new template */
Return False;

3.5.2.2 Template Update

Once the new documents are added into a template, its representation will be slightly changed. It is also important to update the template to represent new documents. Hence, we perform an updating step, Algorithm 3.7, right after Algorithm 3.4. It is worth noting that we only update templates that now represent any new documents. Furthermore, since changes in templates tend to be gradual over time, it is not necessary to process the updating step in every batch. We can set either a threshold (e.g. two hundred more documents) or an interval (e.g. one month) to trigger this step in order to improve the efficiency. We will demonstrate the trade-off between effectiveness and scalability using interval and threshold setting in Section 3.6.7.2.

3.6 Experiments

We begin with a description of datasets, metrics used, and the experimental setup. Our goal is to evaluate InfoShield by answering the following questions.

Q1. *Practical*: How fast is InfoShield, and how well does InfoShield work?
Q2. *Interpretable*: How well does InfoShield visualize clusters? Are there any interesting results with respect to the relative length metric?

Algorithm 3.6: *Preprocess-ES*

Data: An incoming document d_1, and a template set $\mathcal{T}$
Result: An updated template set $\mathcal{T}$ or False if no appropriate template
$I_i \leftarrow |Intersection(d_1, T_i)|$ for all $T_i \in \mathcal{T}$;
$\mathcal{T}^* \leftarrow \mathcal{T}$ ordered by I;
for $T_{i^*} \in \mathcal{T}^*$ **do**
    /* Early stopping */
    if $C(d_1|T_{i^*}) < C(d_1)$ **then**
        $D_{i^*} \leftarrow D_{i^*} \cup d_i$;
        $T_{i^*}.added \leftarrow True$;
        /* Find appropriate template -> Continue with the next incoming document */
        Return $\mathcal{T}$;
    end
end
/* No appropriate template -> Used as an initial document to generate the new template */
Return False;

Algorithm 3.7: *Template-Update*

Data: A template set $\mathcal{T}$ and data D
Result: An updated template set $\mathcal{T}$
for $T_i \in \mathcal{T}$ **do**
    if $T_i.added$ **then**
        $A_i \leftarrow MSA(A_i, D_i)$;
        $T_i^{'} \leftarrow ConsensusSearch(A_i, D_i)$;
        $T_i^* \leftarrow SlotDetection(T_i^{'}, A_i, D_i)$;
        if $C(D_i|T_i^*) < C(D_i|T_i)$ **then**
            $T_i \leftarrow T_i^*$;
        end
    end
end
Return $\mathcal{T}$;

Q3. *Robust*: How much does InfoShield-coarse change as we consider longer n-grams?
Q4. *Incremental*: How does DeltaShield compare with **InfoShield** in terms of efficiency and effectiveness?

Then, we report advantages and observations about **InfoShield**.

3.6.1 Datasets Used

3.6.1.1 Twitter SpamBot Data

We use data from [30]. This data includes the tweet text and user id. The data is split into the following categories:

Dataset	Accounts	Tweets
genuine accounts	3,474	8,377,522
social spambots #1	991	1,610,176
social spambots #3	464	1,418,626
Test set #1 (spambots #1)	1,982	4,061,598
Test set #2 (spambots #3)	928	2,628,181

Table 3.5: Statistics for Twitter bot data

To create each test set, [30] sampled all tweets from 50% genuine accounts, and 50% from either social spambots #1 or social spambots #3. We use the provided test sets, which focus on social spambots only, so we can easily compare results to the best performing methods in [30].

This data not only contains binary labels as to whether particular tweets were posted from bots or legitimate users, but also inherent clusters; i.e. user ids that correspond to legitimate users or bots.

We expect **InfoShield** to cluster most tweets from bots in clusters, ideally in one cluster per bot, and to have few clusters with legitimate users in them. With this intuition, we can create ground truth cluster labels in Twitter data as follows: (1) all legitimate users get labeled -1, since we assume their tweets are different enough that they shouldn't be clustered together; (2) all bots get labeled with their user id.

3.6.1.2 HT Data - Trafficking10k Dataset

The Trafficking 10k dataset is created in [101], where expert annotators manually labeled 10,265 ads from 0-6. 0 represents "Not Trafficking", 3 represents "Unsure", and 6 represents "Trafficking".There are 6,551 ads labeled as not HT, 354 labeled as "Unsure", and 3,360 labeled as HT.

Since the likelihood of an ad being HT is subjective, labeling is a difficult task. In fact, our analysis shows that 40% of exact duplicate ads (without any preprocessing) had label disagreement – i.e. multiple labels for the same exact text. Ads that are exact duplicates account for 12% of the dataset. We expect this labeling issue to occur for near duplicates as well. Therefore, we argue that looking at ads individually, whether manually or algorithmically, is a non-ideal way to find or to label HT cases.

Despite the noisy labels, this is the only HT dataset to our knowledge with labeled data by human investigators. Thus, we use this dataset in our experiments, while being aware that noisy labels may impact results.

This data does not have ground truth clusters. However, to create binary labels, we can call scores 0-3 as not HT, and 4-6 as HT.

3.6.1.3 HT Data – Cluster Trafficking

Cluster Trafficking is a new dataset provided by Marinus Analytics. This data contains cluster labels, provided by domain experts, for a mix of AT-WILL and TRAFFICKING ads, as well as for a new-found behavior that we will call SPAM. The true purpose of SPAM is unknown, but it is distracting to practitioners as they look for true HT cases.

Definition 4. *SPAM: script-generated, eye-catching advertisements with fake contact information, flooding the escort websites with near-identical posts.*

We are given ads from 6 manually-found spam clusters as well as ads from identified HT cases around the US. Cluster Trafficking consists of 157,258 ads, with 6,283 spam ads, 50,985 HT ads, and 99,990 normal ads.

3.6.2 Baselines

The state-of-the-art method for HT detection, HTDN [101], is not open-sourced and does not cluster, but we compare its results to INFOSHIELD on the same Trafficking10k dataset. We also develop three clustering baselines using three standard text embedding methods Word2Vec [74], FastText[17], and Doc2Vec [63]. Each embedding method was trained using 1 million escort advertisements. Then, the resulting ad embeddings are clustered using HDBSCAN [72] with a minimum cluster size of 3[4]. We refer to these three baselines as Word2Vec-cl, Doc2Vec-cl, and FastText-cl.

There are more methods for spambot detection on Twitter, though the majority are still supervised; we compare to three supervised methods [2, 32, 109] and one unsupervised method [31]. These methods all use Twitter-specific features that our domain-independent INFOSHIELD does not use, such as number of mentions, favorites, retweets, posting time, etc. The unsupervised method also does require parameter tuning; a manually-set threshold discerns spambots from legitimate users, and must be re-calibrated for each dataset. Despite these challenges, INFOSHIELD provides comparable results to these baselines.

3.6.3 Metrics

For Twitter data, we have both binary labels and ground truth cluster labels. To compare binary labels, we can report precision, recall, and F1 score. For cluster labels, we use Adjusted Rand Index (ARI) [52]. Note that we do not care about all of these metrics equally; for example, since there are too many HT cases for practitioners to reasonably investigate, we care *less* about recall

[4]Many standard embedding methods, clustering methods, and parameter configurations were tried, but these empirically gave the best results.

and *more* about the precision of InfoShield, since higher precision implies that each flagged case would be more likely to actually be TRAFFICKING.

We calculate precision, recall and F1 by marking all documents that ended up in templates to be of the class of interest.

3.6.4 Q1 – Practical

How scalable is InfoShield? By using InfoShield-coarse to create coarse-grained clusters, and using the more expensive InfoShield-fine on smaller input sizes, we save time. We design an experiment on Twitter data by sampling Tweets the same way [30] did to create the test sets, and report the average runtime for each dataset out of five trials. The result is shown in Figure 3.5. Error bars were too small to be visible, so they were omitted.

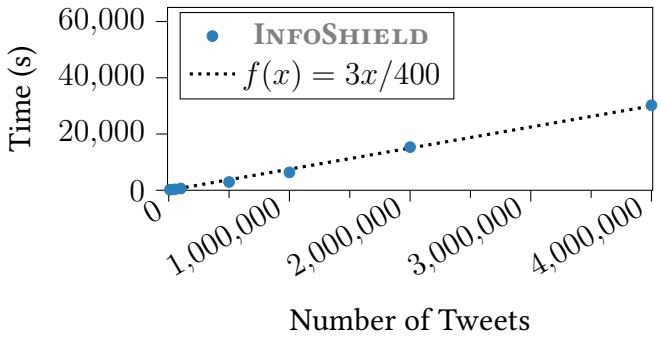


Figure 3.5: *InfoShield is scalable:* Linear on the input size; ≈8hours for 4M tweets, on a stock laptop.

How effective is InfoShield? We run InfoShield, as well as our developed baselines on both the Twitter data and Trafficking10k datasets. We report our results in Table 3.6, comparing against the two highest performing methods from [30].

On Twitter data, InfoShield always performs within ten points of the top contender, despite using no features specific to Twitter such as retweets, favorites, or posting times.

For HT data, we see that InfoShield reports the highest precision; this is crucial since we want to avoid giving false positives to law enforcement at all costs. Law enforcement would rather know that they receive a real HT case (precision) than for all HT cases to be returned (recall) since they likely won't have time to pursue all cases. False positives cause law enforcement to lose trust in the algorithm and abandon it, as happened with previous applied solutions.

3.6.5 Q2 – Interpretable

How well does InfoShield visually interpret the clusters and templates we find? We show a few results of templates for Twitter data, and a censored version for the HT data, with discussion.

Twitter Data								
Dataset	Test Set #1				Test Set #2			
Metric	ARI	Prec.	Rec.	F1	ARI	Prec.	Rec.	F1
InfoShield	**83.2**	*93.0*	*91.2*	*92.1*	**75.7**	*96.7*	*88.9*	**92.6**
Cresci [31]	n/a	**98.2**	**97.2**	**97.7**	n/a	**100**	85.8	92.3
BotOrNot [32]	n/a	47.1	20.8	28.9	n/a	63.5	**95.0**	76.1
Yang [109]	n/a	56.3	17.0	26.1	n/a	72.7	40.9	52.4
Ahmed [2]	n/a	*94.5*	*94.4*	*94.4*	n/a	*91.3*	*93.5*	*92.3*

Human Trafficking Data							
Dataset	Trafficking10k			Cluster Trafficking			
Metric	Prec.	Rec.	F1	Prec.	Rec.	F1	ARI
InfoShield	**84.8**	50.7	*63.5*	**85.4**	**99.8**	**92.0**	**43.1**
Word2Vec-cl	19.4	10.7	13.8	71.7	*99.5*	*83.1*	9.6
Doc2Vec-cl	25.6	10.9	15.3	74.2	*98.8*	*84.7*	16.2
FastText-cl	28.4	22.4	25.1	69.6	*99.6*	81.9	6.8
HTDN [101]	71.4	**62.2**	**66.5**	—	—	—	n/a

Table 3.6: InfoShield performs well: Notice that InfoShield beats or approaches the best *domain-specific* method in both settings. Bold shows the best score, underline shows methods within 10 points of the best. Methods in red are supervised, while InfoShield is unsupervised.

3.6.5.1 Twitter Data

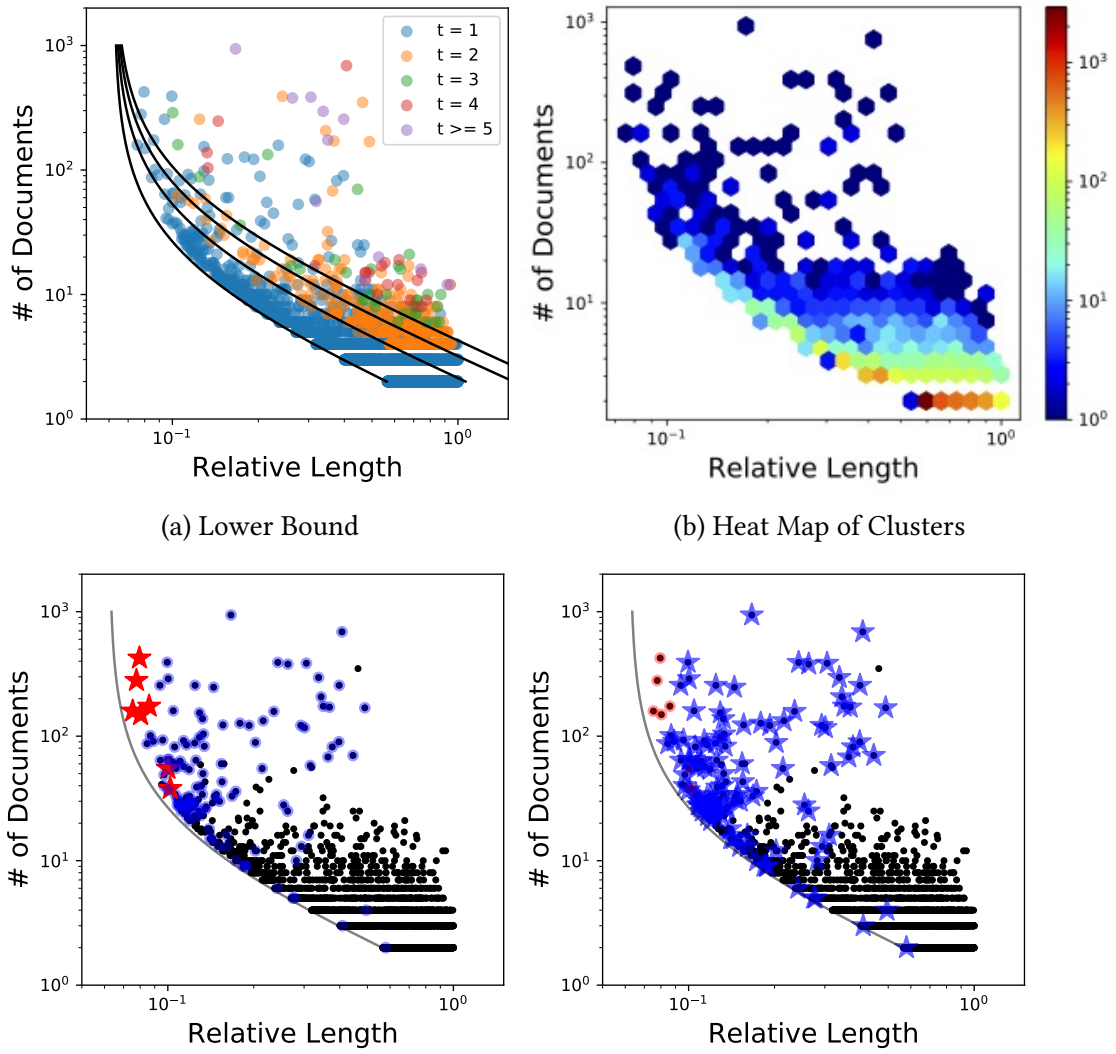
As shown in Table 3.7, we find that 23 Spanish tweets are encoded by the given template. The first 22 ones are exact duplicates, but the last one contains three different words. InfoShield-fine automatically determines that representing those different terms as unmatched results, rather than as a slot, gives a smaller total cost. We can easily spot anomalies within clusters by using the template; the last tweet will have a lower compression rate than all other tweets.

In Table 3.8, we find that all the tweets are talking about the most popular weekly stories. While the first half of all tweets are almost identical, with minor syntax differences, the second half describes the particular stories, which all differ. InfoShield-fine then detects the second half of each sentence as a slot, which we expect to have different content in each tweet. This will help researchers pay attention to the most worth-studying parts.

3.6.5.2 HT Data

In Table 3.9, we show an example template from the HT domain. Unfortunately, we must censor the text to protect potential HT victims, so we only provide the highlighting from the template. For the slots, we give a description of the type of text they represent.

Notice that slots tend to include consistent user-specific information. For example, the second slot, if not empty, always discusses time. With a quick glance, a domain expert can easily find this data, rather than looking at a longer wall of text. For the HT domain, interpretability



(a) Lower Bound

(b) Heat Map of Clusters

(c) Spam Clusters (red stars)

(d) Trafficking Clusters (blue stars)

Figure 3.6: *Perpetrators seem separable*, thanks to our features: (a) shows all clusters (circles) and the lower bounds (black lines) – points are above the lower bound, as expected. (b) heatmap of the same: most points are close to the lower bound. (c) emphasizes the spam clusters as red stars, and (d) emphasizes the HT clusters as blue stars. Note that the majority of spam and HT clusters (red and blue stars) sit apart from the benign clusters.

is key: law enforcement will only have to read one template, rather than each cluster member individually, to determine if this cluster is suspicious.

The slots also contain messy data: i.e. while each slot has a specific purpose in Table 3.9, the text can be in multiple formats, i.e. "until 9pm" vs. "9 P.M", etc. Work could be done to automatically extract and process the information within each slot, but this is beyond the scope of this chapter.

Slot Insertion Deletion Substitution

T_1		**sismo**	**richter**		**km al sureste de puerto escondido oax lat lon pf km**
#1		sismo	richter		km al sureste de puerto escondido oax lat lon pf km
Omit21 Identical Tweets as #1 ...					
#23	sismologico	sismo	magnitud	loc	km al sureste de puerto escondido oax lat lon pf km

Table 3.7: *INFOSHIELD is language-independent:* Spanish template from Twitter dataset.

Slot Insertion Deletion Substitution

T_1	**the**	**mostpopular**	**most popular**	**stories on pr daily this week from**	*	**are**	*
#1	the		most popular	stories on pr daily this week from	instagram to mr t and...		
...							
#14	the		most popular	stories on pr daily this week from	new cover photo rules...		
#15	the	mostpopular		stories on pr daily this week from	whimsical words to hillarys texts...	are	this weeks mos httptcoymwflapn
...							
#27	the	mostpopular		stories on pr daily this week from	understanding sopa to dating...	are	the httptcoploce

Table 3.8: *INFOSHIELD detects slots:* template from Twitter dataset.

Slot Insertion Deletion Substitution

T_1	**omitted for victim's safety**			
#1	(empty)	time	(empty)	(empty)
#2	personal description	time	(empty)	cost
#3	(empty)	time	(empty)	cost
#4	personal description	(empty)	preferences	cost
...18	similar ads			

Table 3.9: *Slots contain user-specific information:* template from HT dataset.

3.6.5.3 Relative Length

Next, we consider the relative length, to further investigate the clusters detected by INFOSHIELD. How does the relative length of a micro-cluster change as a function of the number of documents? Do we notice any differences between the relative lengths of spam clusters vs. HT clusters? Using the Cluster Trafficking dataset, we illustrate the lower bound of relative

length versus number of documents per cluster in Figure 3.6a, where the black lines from left to right denote the lower bound of clusters with one to four templates. For example, the clusters with two templates (orange dots) cannot be on the left side of the second black line. As shown in Figure 3.6b, most clusters are concentrated by the lower bound, meaning that they do not have high numbers of documents. Further analysis surprisingly finds that spam and HT clusters follow patterns in this space. As shown in Figure 3.6c, most spam clusters (red stars) have small relative length with a high number of documents; in Figure 3.6d, there are two patterns of HT clusters (blue stars): (1) the near-duplicate clusters with a high number of documents (but slightly lower than spam clusters), (2) the outlier clusters that lie far from the lower bounds.

3.6.6 Q3 – Robust

How sensitive is InfoShield-coarse to the length of n-grams we use to calculate tfidf scores? We run an experiment on one of the datasets we used for our timing experiments, which contains 100,000 tweets by sampling all tweets from 50% legitimate accounts and 25% social spambot #1 accounts, and 25% social spambot #3 accounts. We detail the results in Figure 3.7.

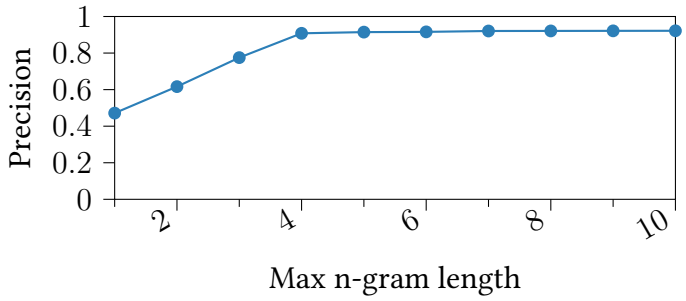


Figure 3.7: *5-grams are enough:* Precision stabilizes after $n = 4$.

3.6.7 Q4 - Incremental

How does DeltaShield compare to InfoShield? We run experiments comparing the effectiveness and efficiency of these methods.

3.6.7.1 DeltaShield-coarse

How do the document-term graphs generated by DeltaShield-coarse compare to the ones generated by InfoShield-coarse? The main difference between these algorithms is the approximation of tf-idf scores in DeltaShield-coarse. To measure the impact of this approximation, we compute the ARI and Homogeneity score (HOM) [91] between the cluster labels produced by DeltaShield-coarse and InfoShield-coarse. A high score signifies that the coarse clusters generated by DeltaShield-coarse are very close to the original coarse clusters generated by InfoShield-coarse. We run this experiment for both HT and both Twitter datasets, as shown in Table 3.10.

Twitter Data			Trafficking Data		
	Test Set #1	Test Set #2		Trafficking10k	Cluster Trafficking
ARI	99.1	99.9	ARI	97.1	99.2
HOM	99.9	99.9	HOM	96.1	96.5

Table 3.10: *DeltaShield-coarse is near-perfect:* we get almost exactly the same clustering for all datasets when processing ads incrementally.

We see that all metrics are high, meaning that we don't lose much information by using DeltaShield-coarse and proccessing ads incrementally.

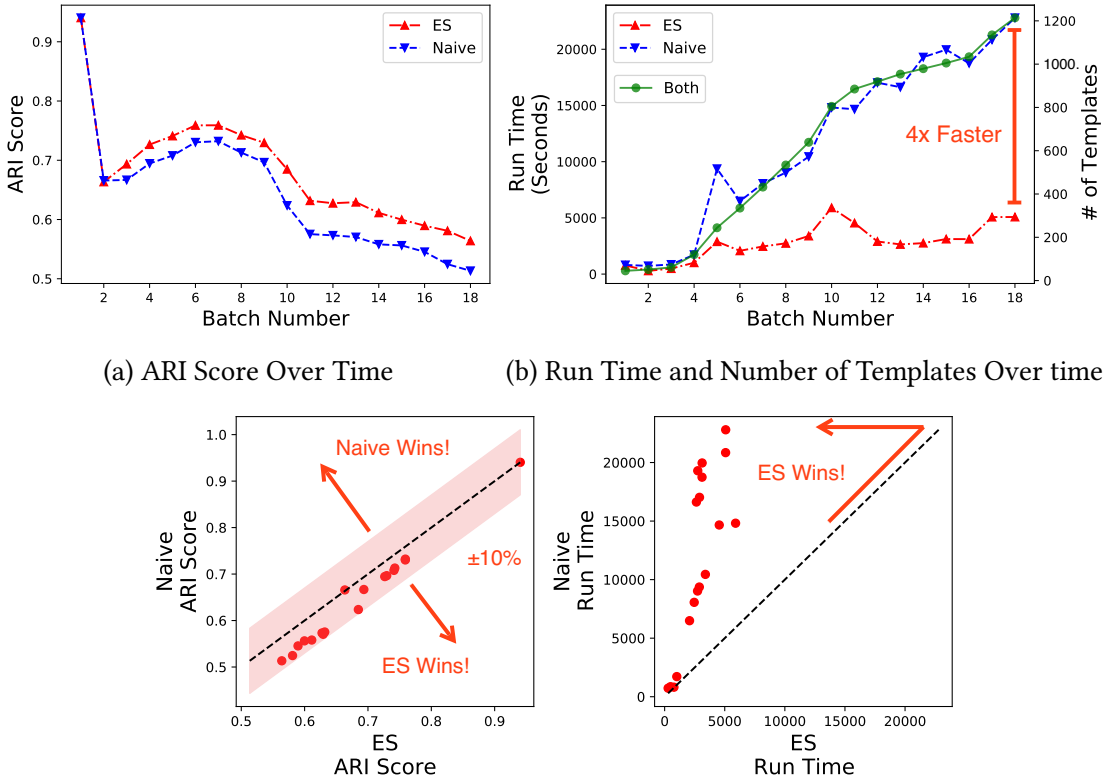
3.6.7.2 DeltaShield-fine

Here we compare the effectiveness and efficiency of DeltaShield-fine. We first compare Algorithm 3.5 (Naive) and Algorithm 3.6 (ES) to demonstrate that the ES method not only outperforms Naive one, but also dramatically decreases the running time. We then study the choice of update frequency, which results in a trade-off between effectiveness and efficiency.

In this experiment, we test an extreme case with the Cluster Trafficking dataset to truly reflect the difference between methods. The dataset is considered as a one big cluster and separated into 18 batches where each batch contains about 2000 advertisements. Note that InfoShield-coarse is not used in this experiment so that we can stress test DeltaShield-fine with a large number of templates. Since our goal of incremental version is to output as close to the non-incremental one, ARI score is used as the effectiveness metric here, where the ground truth is clustering labels generated by InfoShield-fine. It is worth noting that this extreme case will not happen if InfoShield-coarse is still implemented, meaning the ARI score is expected to be low. Our empirical result shows that the average number of templates in one cluster generated by InfoShield-coarse is about 3, which is largely smaller than the number in our experiment (as shown in Figure 3.8b, it is already more than 200 after batch number 4).

ES vs. Naive The ARI scores over time are shown in Figure 3.8a, where we can find the ARI score of the ES method is always higher than the Naive method after the second batch. As depicted in Figure 3.8b, as the number of templates (green line) grows over time, the running time of fitting the templates increases linearly as well. If we compute the slope by the number of templates and running time, the slope of the Naive method is 18, while the one of the ES method is 3, which is 6 times smaller than the Naive method. In Figure 3.8c, the ES method achieves a result with only 10% difference comparing to the Naive method, which is more less a tie. In Figure 3.8d, we find that the ES method always outperforms the Naive method more clearly in terms of run time.

Update Frequency Next, we study the trade-off between effectiveness and efficiency. We mainly compare DeltaShield-fine with update frequency every batch and every three batches. In Figure 3.9a, we find that as the number of incoming batches increases, the gap between two methods increases as well. Nevertheless, the running time of updating every three batches shown in Figure 3.9b is 1.4 times and 2.8 times faster than the one of updating every batch and



(a) ARI Score Over Time

(b) Run Time and Number of Templates Over time

(c) ARI Score: ES vs. Naive

(d) Run Time: ES vs. Naive

Figure 3.8: *DeltaShield-fine Preprocess-ES wins:* (a) shows the ARI score of Preprocess-ES remains higher compared to Preprocess-Naive over time. (b) demonstrates that run time is highly correlated with the number of templates, and shows Preprocess-ES is much faster. (c) and (d) show that Preprocess-ES outperforms Preprocess-Naive in terms of ARI score and run time respectively, where each data point denotes the result from one batch.

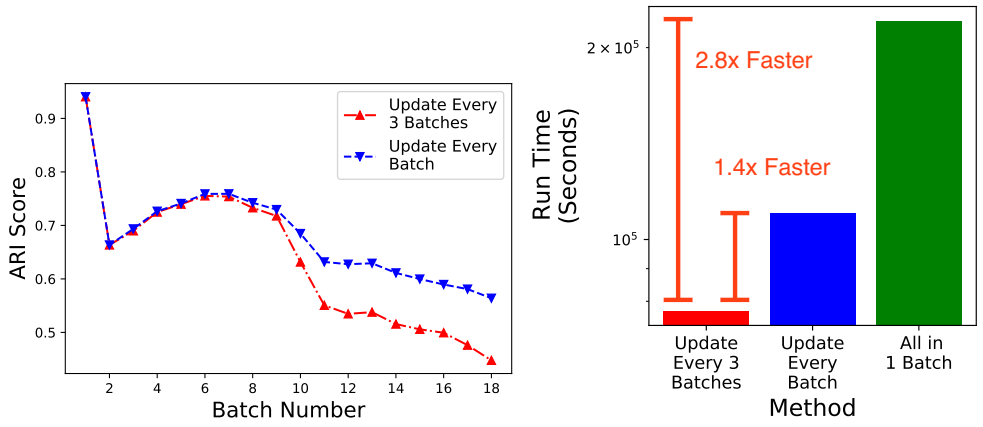
InfoShield-fine, respectively. It will substantially mitigate the expensive overhead when the number of clusters and templates are large, which is especially important to the law enforcement where every second counts for them.

We notice that the low update frequency will slightly hurt the performance. However, we keep in mind that this experiment stress tests DeltaShield-fine, since the number of templates in one coarse cluster is much larger than it will be if we first use DeltaShield-coarse. Alternatively, an end user can consider doing the recomputation of all data periodically, depending on their idle time.

3.7 Discussion and Discoveries: InfoShield at Work

We note that InfoShield has the following advantages:

Advantage 1. *InfoShield is* general, *using no language-specific or domain specific features.*



(a) ARI Score Over Time (1 batch $= 20K$ ads)

(b) Total Run Time

Figure 3.9: *DeltaShield-fine offers strong trade-off:* Even with large update frequency, the accuracy of DeltaShield-fine remains high. (a) shows large update frequency loses effectiveness increasingly over time. (b) shows increasing the update frequency leads to a much lower run time.

In fact, the Twitter data includes tweets in Spanish, Italian, English, and Japanese, and we use no language-specific features in our methodology. In InfoShield-coarse, we automatically let tf-idf penalize common words, so there is no need to include stop-words in our algorithm. Note that the template in Table 3.7 is in Spanish, while the template in Table 3.8 is in English. This makes our method very powerful; it can be run on text in almost any language, or on other text data such as DNA strings.

Advantage 2. *InfoShield is* extensible: *the goal of minimizing the total cost is separate from the algorithms we propose to do so.*

In fact, one could replace InfoShield-coarse and InfoShield-fine with similar algorithms achieving the same end goal of pre-clustering and minimizing the total cost. We propose the algorithms above because they are scalable, and effective on real data.

Advantage 3. *InfoShield does not require any user-defined parameters.*

By using *Consensus-Search* to find the optimal algorithm, we remove the need for user-defined parameters in InfoShield-fine.

3.8 Conclusions

We presented InfoShield, which finds small clusters of near-duplicates in a collection of documents like escort ads for human trafficking detection, and visualizes the micro-clusters in a clear manner.

The main contributions of the method are that it is:

- **Practical**, through scalability and using the MDL principle to be parameter-free,
- **Interpretable**, providing a clear visualization and summarization of clusters, and

- **Generalizable** and independent of domain (Twitter, HT), as well as of language (English, Spanish etc), and
- **Incremental**, by processing new documents on-the-fly, without having to recompute on historical documents.

In the future, we're interested in extending our work on human trafficking detection through spatio-temporal analysis, to understand the movement of possible traffickers through space and time, as well as visualization, so that domain experts can easily interact with the results of our algorithms.

Reproducibility: Code is open-sourced here: https://github.com/catvajiac/InfoShield-Incremental. The twitter datasets are public [30]. The Trafficking10K dataset is available after NDA – email Cara Jones (cara@marinusanalytics.com).

Part II

Visualization towards fighting HT

Chapter 4

TrafficVis: Visualization for Labeling

This chapter is based on work published at *IEEE VIS 2022* [PDF] [Video].

4.1 Introduction

Human trafficking (HT) for forced sexual exploitation is a pervasive societal problem that affects over 4.8 million people world-wide [84], and the majority of HT victims are advertised on online escort websites [86]. However, at-will sex workers also post on these sites, so investigators are focused on finding organized HT rings in these groups, separating them from ads of at-will workers. There is one critical insight to detecting HT; since traffickers entirely control the ad content for their victims [79, 86], ads posted by the same trafficker tend to be similar.

However, the problem is more complex; with the help of InfoShield, domain experts have recently discovered additional modus operandi (M.O.s) in escort ads. For example, SPAM ads with fake contact information flood escort websites, and SCAM ads ask for prepayment and don't provide any services. Our list of updated known M.O.s can be found in Table 4.1.

M.O.	Description of advertised person / behavior
AT-WILL	an independent sex worker freely posting without coercion.
TRAFFICKING	an HT victim who is being forcibly posted, usually as part of a group.
SPAM	computer-generated, eye-catching groups with fake contact information
SCAM	pretends to be at-will, requires upfront deposits from buyers, then disappears.
MASSAGE	a spa or massage parlor offering explicit services, may or may not include HT.

Table 4.1: **Updated known M.O.s from experts**: InfoShield helped to discover and understand *new* behaviors.

These M.O.s have made HT detection more difficult for investigators. According to experts, SPAM ads are particularly problematic due to their volume and eye-catching language. While

Figure 4.1: **Analyzing online escort ads using TraffickVis:** we show one meta-cluster, i.e. micro (text) clusters connected using metadata, on real data. Some text blurred for privacy. **1.** Human trafficking domain expert uses *Micro-cluster panel* to drill down to specific micro-cluster data and associated ads. **2-3.** Expert uses *Timeline panel* and *Map panel* to investigate metadata, noticing inconsistent posting time and regional geographic spread, ruling out spam and scam. **4.** Expert uses *Text panel* to quickly find telling signals; differences between ads in a micro-cluster are highlighted. **5.** Finally, the expert confidently labels the meta-cluster for each modus operandi (M.O.), deciding on *benign* (at-will sex worker), with a small chance of *trafficking*.

a few stumbled-upon examples of these M.O.s exist, their patterns are not yet well known, and it is currently prohibitively time-consuming for domain experts to label clusters so that researchers can analyze them.

At present, the only labeled examples we have are from domain experts stumbling upon groups haphazardly, which is not a sustainable practice for better evaluation as these algorithms progress over time.

While InfoShield does provide micro-clusters for experts to label, there is no intuitive way for them to interact with and label the results. Developing useful visualizations for HT results is challenging due to the multimodal aspect of the data; clusters involve large amounts of text, spatio-temporal posting behaviors and metadata, all of which contain insights that influence final labeling decisions.

We propose TraffickVis, an interactive application for domain experts to visually inspect suspicious meta-clusters (micro-clusters connected with metadata) and label their likelihood to be a particular M.O. Developed through months of participatory design with domain experts, TraffickVis provides coordinated views in conjunction with carefully chosen backend algorithms to effectively show spatio-temporal and text patterns to a wide variety of anti-HT stakeholders and was evaluated by practitioners from multiple domains. In particular, TraffickVis makes the following contributions:

1. **High-impact label generation:** Curating human-generated labels is a notoriously difficult and time-consuming process. TraffickVis allows a variety of practitioners, including criminologists, investigators, and survivors, to label hundreds of ads in just a few clicks, enabling researchers to develop and evaluate better HT detection algorithms down the line
    - addr. *PC2: Few/Expensive Labels: enabling labels for future algorithm development
    - addr. *PC4: Expert interpretability: helping practitioners better understand results
2. **Estimated 10x Time-saving:** Through expert feedback, we find that it takes between 2-4 minutes for an expert to label clusters using TraffickVis. Experts estimate it would take at least 20-30 minutes to investigate clusters with any other method (see Section 4.6). This provides a vast speedup vs the current standard, manual labeling, finally enabling investigators, social workers, and other domain experts to sift through these ads efficiently.
    - addr. *PC3: Legal/Financial Limits: practitioners are already inundated with work, so they cannot label if it doesn't happen quickly.

Reproducibility: The code is open-sourced at https://github.com/catvajiac/TrafficVis. We also provide synthetic data to demonstrate how TraffickVis might be adopted while guarding against privacy risks, ensuring others can try our tool without compromising victims' safety.

4.2 Related Work

We first discuss previous work related to HT, then label generation in the machine learning and visualization communities.

4.2.1 Existing work on HT.

To our knowledge, no published work exists for HT visualization, but there are some known algorithms for HT detection.

Advertisement level solutions. A few HT detection methods focused on advertisement level classification, rather than a clustering task [5, 40, 58, 100]. Many of these methods relied on specific indicators to mark ads as suspicious, such as keywords indicating underage victims. However, due to the adversarial nature of HT, predefined features will not stay relevant over time. Supervised methods used text and image data to predict the suspiciousness of an ad [100, 108] on a particular dataset. Unfortunately, the dataset used in these algorithms has noisy and biased labels; ads containing the word "Asian" are significantly more likely to be flagged as HT, irrespective of whether they actually were or not. Also, these ads are old, and posting behavior has dramatically changed since then, especially given the fall of Backpage [93], which greatly disrupted the escort market. Most problematically, these methods cannot find *groups* of organized activity, which is problematic for investigators – if a particular trafficker is being investigated, they need to discover all ads that are relevant to understand the scope of HT and quickly provide help to victims.

Cluster level solutions. Some related work tries to find connections between ads. Some methods train binary classifiers to predict if two ads are connected [76, 87], while others uses local active search approach to retrieve connected ads [88]. Sentence-level embedding and hashing techniques have also been used to find groups of ads [68]. InfoShield [66], as described in Chapter 3, uses Minimum Description Length to create these templates, containing no black-box components.

Unfortunately, none of these published methods include interactive visualizations. Furthermore, even the cluster level solutions make the assumption that each text-based cluster represents a different organized activity. However, this is not always the case – many traffickers change the templates they use to write ads over time, or use different templates in different regions. We aim to exploit the connections between these text-based clusters in our visualization. In this chapter, we build our system upon the micro-clusters (text clusters) found by InfoShield.

4.2.2 Label generation systems.

Labeling is a crucial step to the development and evaluation of machine learning models on real-world data, but is also notoriously a labor and time-intensive process [15, 110]. Often, particularly for simple labeling tasks, crowdsourcing marketplaces such as Amazon Mechanical Turk (MTurk) are used to quickly generate labels [23]. However, since the profiles of workers are not well known, there are quality issues associated with MTurk [44, 55]. Furthermore, it is not well-suited to problems where significant domain knowledge is required.

Broadly, visualization can provide valuable knowledge to the analyst [28, 107], which can help the labeling process [14]. These labels are commonly used in two ways: either in real time to improve the performance of the current task, such as in active learning settings, or for the collection of data to be used in downstream tasks.

Active learning approaches. There is much work on active learning and the role of visualization in improving these algorithms [27]. Some systems focus on recommending the most probable labels based on semi-supervised models on a larger set of disjoint labels [33, 99]. The use of Interactive visual labeling (VIAL) systems [14], which are built over active learning algorithms, can improve their performance [15].

Labeling for downstream tasks. There are many approaches for labeling complex, multivariate data for downstream tasks. Some are more generic frameworks for multi-variate data [12, 51]. Others make custom interfaces for highly specialized data or tasks, such as motion-capture data [13] and image segmentation tasks [1].

In this work, we focus on the visualization of complex, multi-modal HT-specific data where the label generation is for a downstream task.

4.3 Design

Our goal is to build an interactive system for HT that lets a domain expert investigate and label possibly suspicious meta-clusters. TraffickVis is the result of 9 months of participatory design [94] in conjunction with domain experts. More specifically, we received weekly feedback from two domain experts: a Senior Research Scientist at *Marinus Analytics* with six years of analyzing escort ads for HT and extensive experience working in government, and an HT survivor and analyst with 20 years of experience helping trafficked minors on the street. Both of these domain experts have collaborated with local and federal agents and are familiar with the justice system, but give very different perspectives. Through extensive discussions with these domain experts, we distilled the following design considerations (C1– C3) for TraffickVis.

C1. Big Picture: *Connect micro-clusters into larger activities.* While traffickers often entirely control the ad content for their victims [86], over time they might make changes to the text or post multiple ad templates at once. Domain experts state that metadata (e.g., phone number, email address) can be useful in connecting micro-clusters into larger organized activities, which we call meta-clusters (formally defined in Section 4.5).

An important design principle of labeling systems is to allow users to see both high level and low level information [92]. While domain experts are generally interested in the behavior of the meta-cluster as a whole, they also would like the ability to drill down into a particular micro-cluster, particularly if it has different behavior than other micro-clusters in the meta-cluster.

C2. Multimodality: *Displaying complex, multimodal data.* With many previous labeling systems, the challenge was label recommendation, rather than visualizing complex, multimodal data. However, in our case, the challenge lies in effectively visualizing spatio-temporal and text data of meta-clusters to domain experts.

Each meta-cluster has many time series. The posting pattern of metadata fields, geographic locations, and micro-clusters can each be represented as a time-series through the lifetime of the meta-cluster. Furthermore, each micro-cluster within the meta-cluster has its own time-series for each of these fields.

Since many domain experts look for suspicious keywords in ads to determine the M.O. [3, 36], thoughtfully showing the text in each meta-cluster is important. In particular, our domain experts often mentioned the need to be able to drill down into particular ad text while still being able to see the overarching text patterns.

Given the importance of promoting rich interactions between the data and domain experts [37], we must enable them to navigate through this data efficiently.

C3. Usability: *Making the interface usable for practitioners:* Since some of our intended users will be investigators and social workers, all visual techniques must be easily understood by non-experts in visualization. How can we convey patterns intuitively, using methods that the average expert will understand?

Furthermore, through our domain experts, we've gathered that investigators like to see as much information at once as possible; they do not like applications that require lots of scrolling back and forth to see the results. However, we also must ensure we do not overwhelm the expert [60].

The above design considerations cover the features mentioned to us during our conversations with domain experts.

4.4 Method

How can we design our backend algorithms to address the design considerations synthesized in Section 4.3? Here, we will use the same labels, C1— C3, to specifically mention how our algorithms address these considerations.

4.4.1 InfoShield

We use InfoShield [66] (see Chapter 3, to create micro-clusters. As a reminder, InfoShield exploits the insight that similar ads are likely written by the same person. More specifically, InfoShield is comprised of two parts. First, *InfoShield-coarse*, which quickly creates micro-clusters by connecting ads that share common phrases (up to 5-grams) with a high *term frequency inverse document frequency (tf-idf) [56]* score. Then, *InfoShield-fine* uses the Minimum Description Language (MDL) [89] principle to generate a template for each micro-cluster, aligning ads to find similar phrases and highlight the differing ones through insertions, deletions, or substitutions. InfoShield also finds *slots* — portions of the template that differ for most ads. Slots often contain information specific to that ad, such as name, contact time, or available hours. A visual example of this process is shown in Figure 4.2. InfoShield also ranks micro-clusters using the relative length metric r (compression ratio of the micro-cluster using the calculated template), which we will use in Section 4.4.3. We chose InfoShield be-

cause it is scalable, achieving near-linear performance on the number of ads processed and explainable, which justifies the decision to create the group to investigators.

Figure 4.2: *Pipeline for* INFOSHIELD*:* Taking crawled ads as input, INFOSHIELD-coarse groups these ads into micro-clusters, and INFOSHIELD-fine highlights the common phrases in each ad by finding a common template.

4.4.2 Meta-Clustering: C1 (Big Picture)

Since micro-clusters are constructed only using text features, multiple micro-clusters can actually be part of the same activity. Therefore, we connect micro-clusters (c_i) into *meta-clusters* (M_j) based on extracted metadata – images, emails, phone numbers, and social media accounts. We consider two micro-clusters c_1, c_2 to be part of the same meta-cluster M_j if two ads $a_m \in c_1, a_n \in c_2$ share at least one metadata field. Figure 4.3 shows an example of how six micro-clusters can be connected into three meta-clusters.

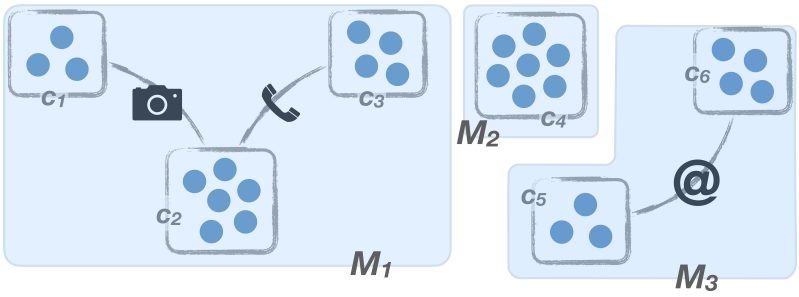


Figure 4.3: *From micro-clusters* (c_i) *to meta-clusters* (M_j)*:* By incorporating metadata – images, phone numbers, and social media accounts – we combine 6 micro-clusters into 3 meta-clusters, each of which are part of the same group.

This addresses consideration C1 (Big Picture) since we are connecting micro-clusters into larger organized activities. We consider these metadata to be hard connections because of their nature; it is very unlikely that two unrelated micro-clusters are using the same contact information or the same exact image. If we were using metadata fields where the connections were less

certain, running a clustering algorithm on this constructed metadata graph could have been an appropriate next step.

4.4.3 Ranking: C3 (Usability)

We would like domain experts to both view and label the most suspicious clusters first, so that we can get useful labels while saving practitioner's time, addressing consideration C3 (Usability). Domain experts consider some suspicious signals to be (a) a large number of ads and micro-clusters in the meta-cluster, and (b) very similar text. Large numbers of ads and micro-clusters are considered suspicious since they hint at the existence of a large organized group, with too many ads and clusters to be an individual escort. Similar text can also be considered a suspicious signal since traffickers often use the same template to advertise many victims. Therefore, we devise a ranking heuristic to prioritize types of meta-clusters with those behaviors. Since we observe the number of ads and number of micro-clusters in meta-clusters to be Pareto distributed, we will scale these values logarithmically. In order to capture text similarity, we consider the relative length metric (that is, compression ratio) r given from InfoShield, which measures the goodness of compression for a particular micro-cluster. r is close to 1 if the compression is bad, and smaller if the compression is good, signifying high similarity among the ads of the micro-cluster. More specifically, for a given meta-cluster, let N be the number of ads, M be the number of micro-clusters, and let $R = \{r_1, r_2, ..., r_M\}$ be the relative length scores for each micro-cluster. We give each meta-cluster a suspiciousness score s by

$$s(N, M, R) = \frac{\log N + \log M}{\frac{1}{M}\sum_{i=1}^{M} r_i}. \tag{4.1}$$

This metric will prioritize meta-clusters that have a large number of ads and micro-clusters, that also have good compression. Since the compression ratios R are positive and less than 1, micro-clusters with more similar text will boost the score. Finally, we present the meta-clusters in TraffickVis from high to low score. For the few labeled clusters that we have, we do observe that this metric ranks national-level HT rings and SPAM meta-clusters first.

4.5 Iterative Design

Once we construct meta-clusters, how can we visualize them in a way that addresses our design considerations from Section 4.3? We will describe each part of the interface through a usage scenario. As we introduce the features of TraffickVis, we will annotate which design consideration(s) C1— C3 it addresses. We then present a scenario based on the real experiences of crime analysts, inspired by actual comments given by experts on the presented data during solicited feedback (see Section 4.6).

4.5.1 The User Interface

First, we will quickly describe each part of the interface, and further elaborate on how each panel is used in Section 4.5.2.

The top banner shows basic statistics for the meta-cluster. The *Micro-cluster panel* shows the posting behavior of the top 10 micro-clusters with the highest number of ads throughout the lifetime of the meta-cluster, as seen in Figure 4.6 (left). This addresses C1 (Big Picture) by allowing users to see the posting behavior of the micro-clusters and the overall meta-cluster simultaneously. By hovering over a particular cell, a tooltip displays the number of ads per day in that micro-cluster. A multi-select dropdown above the *Micro-cluster panel* (Figure 4.6 (left)) allows users to select a particular subset of micro-clusters, which will update all panels and text in the rest of the interface. This feature further addresses C1 by letting users customize exactly which micro-clusters they can drill down into. By deselecting all micro-clusters, all meta-cluster data will once again be displayed in all panels.

The *Timeline panel* (Figure 4.4) shows the usage of metadata and the number of micro-clusters with posted ads per day over the lifetime of the meta-cluster. By hovering over any date, the time-series values will be displayed. Since the number of micro-clusters is a feature derived from InfoShield, it is displayed separately. The *Map panel* (Figure 4.5) also shows the geographic spread of the meta-cluster or selected micro-clusters. A tooltip shows the number of ads posted in each location. These panels help us display complex, spatio-temporal data usefully, addressing C2 (Multimodality).

The *Text panel* (Figure 4.6) allows domain experts to scroll through the text templates generated by InfoShield, as shown in Figure 4.6 (top), which give a general sense of the phrasing for each micro-cluster. If any micro-clusters are selected, the *Text panel* will show the individual ads for those particular micro-clusters, as shown in Figure 4.6 (bottom), highlighting any deviations from the template as insertions, substitutions, or slots, as designated by InfoShield. This panel helps us display complex text data usefully and drill down into individual micro-clusters when needed (C1, C2).

The *Labeling panel* (Figure 1.5) lets the domain expert quickly label the meta-cluster on a scale of 1 (very unlikely) to 5 (very likely) for each possible M.O. We use sliders and a discrete scale, rather than a continuous input probability, for ease of use, addressing C3 (Multimodality). Upon clicking the 'Next meta-cluster' button, these labels are saved to a CSV file and a new meta-cluster is displayed.

4.5.2 Usage Scenario: Analyst finding a massage parlor cluster with suspected HT

We present a usage scenario to illustrate how each panel can work together to help an analyst (e.g., investigator, social worker) use TraffickVis to investigate a meta-cluster. This scenario is based on expert feedback solicited on the meta-cluster depicted in Figures 4.1, 4.4, 4.5, and 4.6.

First, the analyst sees high-level statistics on the top banner, observing that for this 200 ad cluster, there are a lot of images posted, which often correlates with organized behavior. She then moves to the *Micro-cluster panel* (Figure 4.6 left) to inspect the individual micro-clusters. The analyst may choose to further investigate the consistent volume of ads in micro-cluster c_1 during the last few months of the meta-cluster using the multi-select dropdown just above the

Micro-cluster panel. When she does, the entire interface will populate with that micro-cluster's geographic, temporal, and text data.

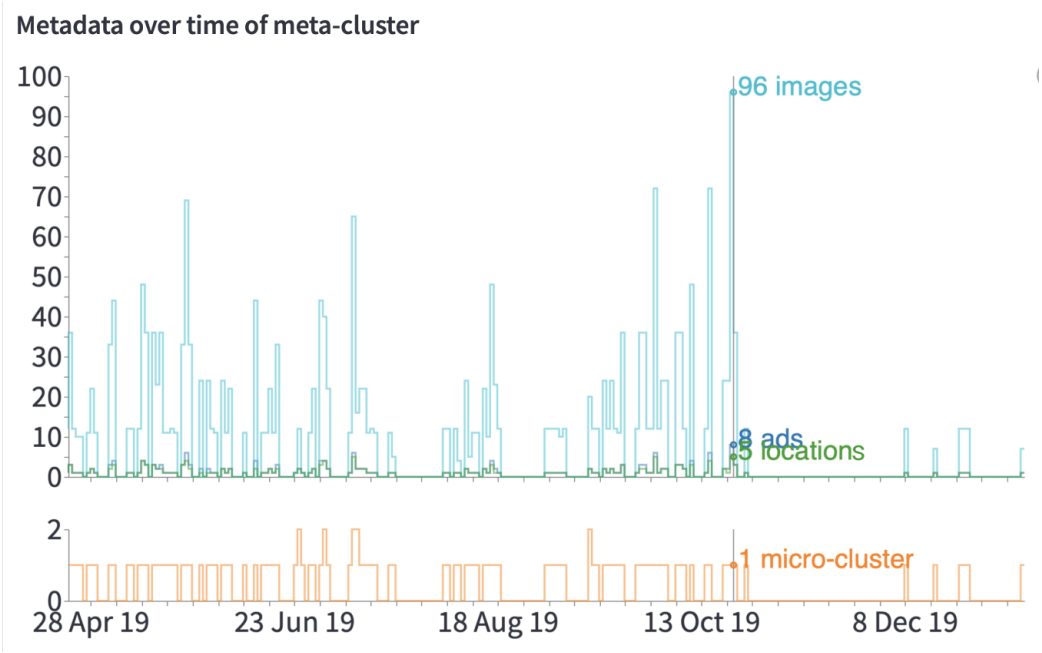


Figure 4.4: **Irregular spikes in *Timeline panel***, indicating to experts that this is not script-generated posting behavior, but rather human-generated, ruling out SPAM and SCAM labels.

Next, the analyst can look at metadata usage and the number of micro-clusters per day in the *Timeline panel* (Figure 4.4). She may notice the somewhat inconsistent posting over many months, with some "hot spots". This does not look like script-generated posting behavior, which makes the label SPAM less likely. She may notice that there are few unique locations per day, which also supports that the ads are not scripted. Using the *Map panel* as shown in Figure 4.5, she can investigate which locations are most popular, noticing that there is some regional spread focused on bigger cities in the Midwest and East Coast. This could be indicative of TRAFFICKING or AT-WILL behavior, with an individual or group circling between cities likely to have many customers, which rules out SPAM and SCAM.

Next, she inspects the text of each ad in the scrollable *Text panel* (Figure 4.6). If a particular meta-cluster is not selected, then only the InfoShield templates for each micro-cluster will be shown. This way, she can compare the differences between the wording in these micro-clusters. By selecting a micro-cluster, then the *Text panel* will change, showing the template and the individual ads for that particular micro-cluster, and highlighting where the individual ad text differs from the template. As found by InfoShield, blue represents insertions, deletions, and substitutions, and red represents parts of the ad that differ from most other ads in the micro-cluster. Looking at c_2, the analyst might notice the following interesting text features (T1-T4), as annotated in Figure 4.6.

T1. **Social media presence:** sign of a legitimate person (TRAFFICKING or AT-WILL). Handles blurred for privacy.

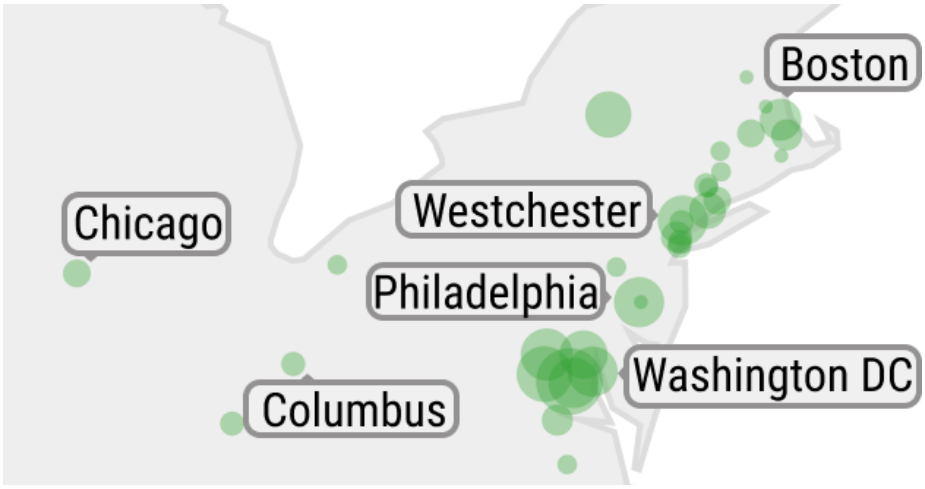


Figure 4.5: **Meta-cluster focuses on big cities**: ads are focused on bigger cities in the Midwest and East Coast. Size represents the number of ads posted in that location. This could be indicative of TRAFFICKING or AT-WILL workers circling between cities with many customers.

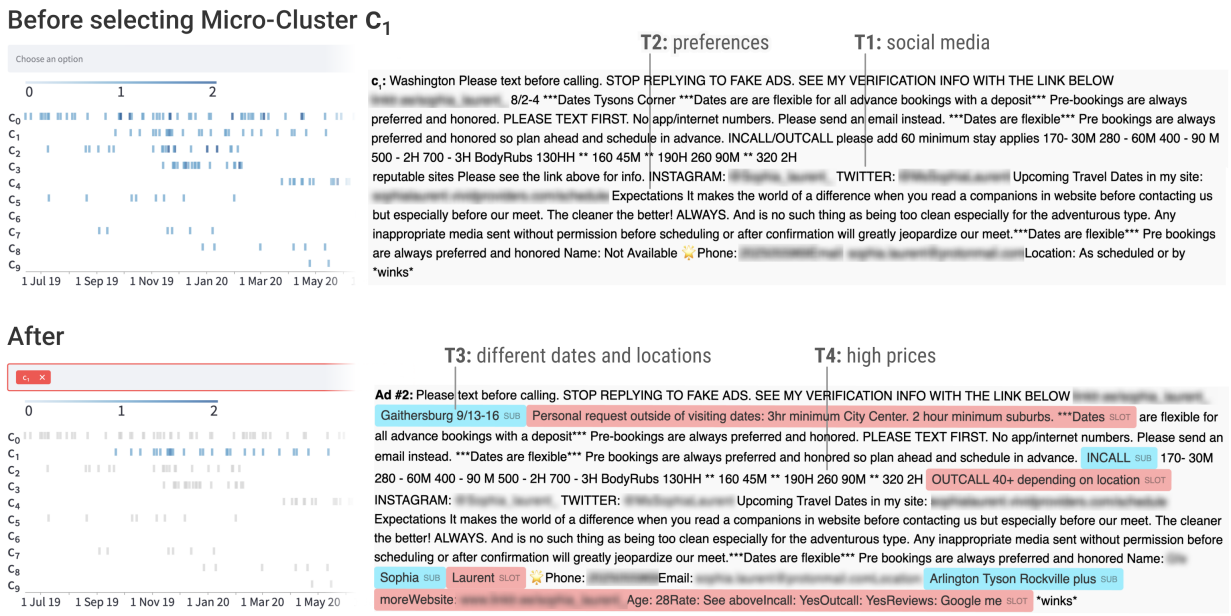


Figure 4.6: **Drilling down into specific micro-clusters**: annotations correspond to *text features* T1-T4 from Section 4.5.2. **Top:** *Micro-cluster panel* shows the posting activity for all micro-clusters. *Text panel* shows the template text for micro-cluster c_1. **Bottom:** upon selecting micro-cluster c_1, *Micro-cluster panel* updates to highlight c_1 and *Text panel* shows the individual ads with differences highlighted. As found by InfoShield, blue highlights represent substituted phrases, and red highlights represent parts of the ad that differ from most other ads in the micro-cluster (known as *slots*). Some sensitive text is blurred.

T2. **Preferences:** asks for cleanliness and no unsolicited photos: likely AT-WILL sex work, but TRAFFICKING still possible.

T3. **Different dates & locations:** possible sign of traveling TRAFFICKING or AT-WILL sex worker.

T4. **High prices:** usually a sign of a popular AT-WILL worker.

Finally, after inspecting the information in each panel, the analyst uses the *Labeling panel* (Figure 1.5) to confidently label the meta-cluster. Given the regional geographic spread and that many signals point to the ad not being script generated, the analyst labels this meta-cluster as likely AT-WILL (an at-will sex worker), with a small chance of TRAFFICKING based on the suspicious keyword.

4.5.3 Iterative Design Process

TRAFFICVIS was developed through 9 months of participatory design. We started our design by discussing the interest of visualization for fighting HT with our domain experts, getting their sense of what the basic needs of investigators and crime analysts would be, as well as the perspective of a survivor. Then, over the span of several weeks, we iterated over a few possible sketches of TRAFFICVIS. Each week, domain experts would give us feedback that would change our final design. As the sketches were implemented on real data, we iterated over many possible encodings of the data with domain experts. We give some examples of iterations for a few panels below.

4.5.3.1 *Micro-cluster panel*

We considered network-based representations of the data; nodes would be micro-clusters or individual ads, and edges would be metadata fields or keywords. However, we decided against it due to the hairball effect – most ads are connected with similar keywords and metadata, resulting in many clique-like structures. Since the relationships between micro-clusters were not useful to show, we focused on the timelines between each micro-cluster, as is shown in *Micro-cluster panel*. This helps us address C1 (Big Picture), since users are able to quickly decide if they want to drill down into lower-level information about each micro-cluster.

4.5.3.2 *Text panel*

Here, we display the results of InfoShield, which detects five types of regions; *constant strings* which are the same in most ads, *insertions, deletions, and substitutions*, and finally *slots*, or places where the text differs in most ads. Originally, we considered displaying output the same way as described in InfoShield, where constant strings were highlighted in yellow, insertions in green, deletions in grey, substitutions in blue, and slots in red. However, this representation became very visually crowded, and the average investigator does not care about the differences between slots and insertions. Therefore, we changed the representation to not highlight constant strings, make insertions/substitutions/deletions light blue, and slots red. This helps with C3 (Usability), since we are making the design less complex for domain experts that do not need this specific information.

Also, we were originally displaying all of the text for all micro-clusters in this scrollable panel, which ended up being overwhelming and cumbersome for domain experts, since they had to scroll down very far to get to some ads. Instead, we decided to only show the text templates from InfoShield if no micro-cluster was selected, and let the user decide which actual ads they wanted to see by using the multi-selector. This way, we make it easy to drill down into actual ads, addressing C1 (Big Picture).

4.5.3.3 Drilling down into specific micro-clusters

Domain experts specifically asked for this feature in order to drill down into specific mirco-clusters if needed. At first, we added a selector to *Micro-cluster panel* itself, which enabled users to click on a particular row to select a micro-cluster. However, this ended up being unintuitive, since domain experts would not realize that anything would happen if they did click on each row, so this feature was not being used. Therefore, we created an explicit dropdown with explanatory text. Also, in this way, it was easy to implement mutli-selection without asking the domain expert to remember keyboard shortcuts. Since some investigators may not be as familiar with common keyboard shortcuts, this makes it easier for them to use TraffickVis (C3).

4.6 Evaluation

We solicited expert feedback from domain experts to answer the following questions about the efficacy of TraffickVis for inspecting and labeling suspicious meta-clusters. Q1 – Q3 directly correspond to our Design Considerations C1– C3 introduced in Section 4.3.

Q1. Evaluating C1 (Big Picture): Is the distinction between meta-clusters and micro-clusters useful to experts? How do they interact with the clustering results?

Q2. Evaluating C2 (Multimodality): How do experts interact with metadata plots? How do these plots influence their labeling decisions?

Q3. Evaluating C3 (Usability): Do the solicited experts, which have varying backgrounds, all find it easy to use? How do they expect other types of experts would react to the design (i.e. other investigators, government employees, etc.)?

Q4. Which features of TraffickVis are most important to experts? Are there any *insights about the labeling process* that we gain from seeing how experts look at the data?

Q5. How quickly can experts label meta-clusters? Do they believe it will significantly *speed up the labeling process*?

4.6.1 Intuition Behind Setup

There are few domain experts in HT that analyze escort ads. This not only makes it challenging to solicit them for feedback, but also tasks us with making our study as efficient as possible as to make the best use of their very limited time. We did not want each expert to take more than approximately an hour of their time giving feedback. Since the current solution for domain

experts is manual labeling, we decided there is little point in A/B testing. Any clustering and visualization would provide speedup vs. manual labeling. Instead, we focus on asking experts to label more meta-clusters and soliciting feedback on the tool as a whole.

4.6.2 Solicited experts

We asked domain experts with various experience in HT to participate. We recruited four experts; For brevity, we will use E1, E2...E4 to refer to Expert 1, Expert 2, ...Expert 4. E1 and E2 are from *Marinus Analytics*, a Pittsburgh-based startup focused on fighting HT and the providers of our data. E3 is an HT survivor that now helps rescue trafficked minors on the street. E4 is a criminology master's student studying escort advertisements and HT.

The average time our experts were involved in studying HT varied greatly, from 1 to 20 years. On average, experts rated themselves as a 4.3 ± 1.15 out of 5 on their expertise in labeling escort ads, and a 3.6 ± 0.58 out of 5 on their experience with AI and clustering algorithms.

Some of our domain experts had extensive experience with looking for cases themselves. E2 and E3 discussed using keyword searches and various statistical techniques to investigate clusters. In their experience with investigators, they often start with a tip and try to build a case manually using street knowledge. Experts say some officers may look for online ads to try and glean some information, but it's difficult to find in a large set of unorganized ads on a webpage.

4.6.3 Dataset used

We used a random sample of escort ads given to us by *Marinus Analytics*. These ads were crawled from multiple common escort websites which are suspected to contain organized activity. Then, we ran InfoShield on these ads, followed by our meta-clustering algorithm. We then manually picked 10 meta-clusters that differed in their spatio-temporal distributions, number of posted ads, and text templates (i.e. length, use of emojis, presence of suspected TRAFFICKING keywords), to increase the likelihood of differing labels. We chose 10 in an effort to limit the time taken for each interview to no more than one hour.

4.6.4 Procedure

Our protocol was approved by the IRB. Each expert signed a consent form before the interview was conducted. All experts were interviewed separately over Zoom. All experts' movements on the interface and their audio were recorded. Experts got access to the interface by the interviewer sharing their screen and letting the expert interact with it using Zoom's remote control feature. Each interview started with a 5-minute introduction, outlining the structure of the interview. Then, each expert was asked the questions about their background in HT, analyzing escort ads, and whether they have any insights to investigators. The exact wording of these questions can be found in the supplemental material.

We then gave a 5 minute tutorial on TRAFFICVIS, making sure the expert had common definitions for all M.O.s. Then, we let the expert explore the interface using Zoom remote control and label the 10 clusters in our dataset. The labeling options were 1: Very unlikely, 2: Unlikely, 3: Unsure, 4: Likely, and 5: Very likely. Experts were encouraged to think aloud

as questions and comments arose, and to verbally explain the clues that led them to their final decision. If they had not previously explained their thought process, upon finalizing their labels for a meta-cluster, the interviewer would ask them to quickly explain why. We recorded the elapsed time to label each meta-cluster as the moment the interface loaded a new meta-cluster to the moment they clicked the 'next meta-cluster' button.

Once experts finished labeling all clusters, we would end the session with a few exit questions asking for feedback about TrafficVis, which can be found in the supplemental material. Finally, the expert was asked to complete a quick questionnaire offline, which can also be found in the supplemental material.

4.6.5 Results and Design Lessons

Experts had overwhelmingly positive feedback on TrafficVis. They predominantly looked at the text to identify the behavior of clusters, but used the geographic spread and timelines to supplement their thinking. Often, specific keywords would jump out at them. Based on their feedback, we distilled some central design lessons. We show the number of experts that commented on each lesson, without being prompted, in Figure 4.7. Next, we elaborate on the design lessons we learned (L1 – L6) and how they answer our questions Q1 – Q6.

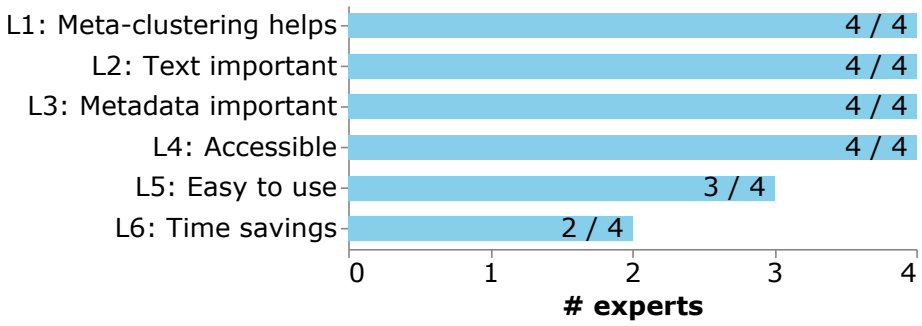


Figure 4.7: **Design Lessons from Expert Feedback:** the number of experts that explicitly commented on each design lesson, without being prompted.

L1: Meta-clustering helps – (Q1)

All experts enjoyed that the ads were clustered together, rather than looking at them individually as is typically done in the current approach, manual labeling. The distinction between micro-clusters and meta-clusters was also appreciated by the experts. E1 mentions that they

> "*liked the ability to look at the meta-cluster and then select some of the micro-clusters and see some of the ads within those...[to] be able to drill down*".

E3 spent much of their time drilling down into particular micro-clusters to see how they varied, saying

> "*being able to look at the individual micro-clusters really helps.*"

E4 also mentioned the distinction between meta and micro-clustering as useful, saying that they liked

"*the fact that you could go within certain ads within the micro-clusters*"

L2: Text is important – (Q2, Q4)

Text was the most defining feature. All experts really enjoyed the text clustering, spending the majority of their time interacting with TraffickVis on the *Text panel*. E4 says they

"*liked the ad text...I used that a lot in how I was labeling.*"

For each meta-cluster, every expert cited text as part of their reasoning for selecting their labels. For most meta-clusters, all experts looked at the text the majority of the time. In particular, the presence or absence of certain keywords were telling. When looking at the first meta-cluster, E2 mentions

"*I'm looking for keywords I usually see when I look at this stuff*".

We distilled the most commonly cited text indicators of TRAFFICKING.

- Mention of exotic ethnicities (E1, E3)
- Scarcity: girls are constantly changing, new in town, or leaving soon (E1, E3)
- Multiple girls advertised (E1, E2)
- Particular keywords, redacted by request of the participating domain experts (E1 – E4)
- Offering high-risk services (E1 – E4)

Experts also commented that short ads or differing fonts in ads are possible indicators of SPAM or SCAM (E1, E2, E4).

L3: Metadata provides useful insights – (Q2, Q4)

Experts commonly referenced various metadata properties as influencing their labeling decisions. All experts interacted with all panels during their labeling. E1, E2, E3 explicitly mentioned looking at the number or similarity of phone numbers and E1-E4 all mentioned the geographic spread when explaining their rationale for labeling particular meta-clusters. E3 referenced the temporal distribution multiple times, saying

"*since you have multiple ads a day, this is likely not an individual escort...[they] post a ton of websites on one day, but spread over a year*" as part of their reasoning.

E4 particularly liked seeing the geographic spread:

"*the geographical spread is very good too, it's a very good indicator especially when you're labeling spam and scam, and also trafficking...*"

Experts commented that seeing an explicit location offered in the ad text is an indicator against spam or scam (E2, E3, E4).

L4: Accessible – (Q3)

TraffickVis is useful for many anti-HT stakeholders. All experts commented that TraffickVis could be used by domain experts and investigators. E1 believes that TraffickVis is

"*really powerful for finding large organized crime groups.*"

E1 suggested it would likely be

"*more relevant to larger national law enforcement groups than local [law enforcement], but [they] think it would be helpful in building cases or showing relationships.*"

E3 mentioned that a huge benefit of TraffickVis was the curation of meta-cluster labels, stating that TraffickVis is

> "*useful for labeling, for supervised training which is currently very difficult...and also for verifying whether the underlying algorithms are correct.*"

E3 was particularly excited about the possibility for investigators and attorneys to use TraffickVis, stating that

> "*it would help investigators retrace their steps from jury or prosecution during testimony...it would really add to explainability and justification for why that individual is being indicted...[or] targeted.*"

E2, E3, and E4 all explicitly mentioned that they'd like to label more clusters with TraffickVis.

L5: Easy to use – (Q3)

Broadly, experts liked the interface, saying they were "very impressed by the tool" (E2, E4) and that it's "easy to use" (E3, E4). E2 and E3 particularly enjoyed the one page layout, mentioning that "you didn't have to jump to another page to record your responses" and "the layout is nice, don't change it".

L6: Time savings – (Q5)

TraffickVis makes labeling possible. E3 commented on the possible time savings as compared to Marinus' escort ad exploration software:

> "*it's quick...even with TraffickJam it would take 20-30 minutes per cluster to try and figure out what is going on. And then you wouldn't even be able to label the ads. This is a huge advantage over the way things are currently done. For law enforcement officers they have no way...they have to do everything manually, there's no way.*"

E2 also mentioned the time savings, saying

> "*I could really see this speeding up scanning ads, especially if you've got one cluster of ads and you see another cluster in the same geographical area, even over the same time period.*"

Experts consistently need about 2-3 minutes to provide labels with TraffickVis, with an average of 2 minutes and 36 seconds. The distributions of the time taken to label, per expert and per meta-cluster, are shown in Figure 4.8, in comparison to E3's estimation of 20-30 minutes. This confirms a central motivating point of TraffickVis: the current solution, manual labeling, is so time-intensive, that it is rarely ever done. With TraffickVis, it's *feasible* to solicit labels from domain experts.

4.6.6 Distribution of labels

The final labels, averaged among all experts, are shown in Figure 4.9 for each meta-cluster. Circles represent the predominant label for each meta-cluster, while tick marks represent all other labels. For example, in MC1, our experts gave an average score of 1, 1.25, 3, 2.5 and 4.25 to SPAM, SCAM, TRAFFICKING, AT-WILL and MASSAGE labels respectively. This strongly indicates

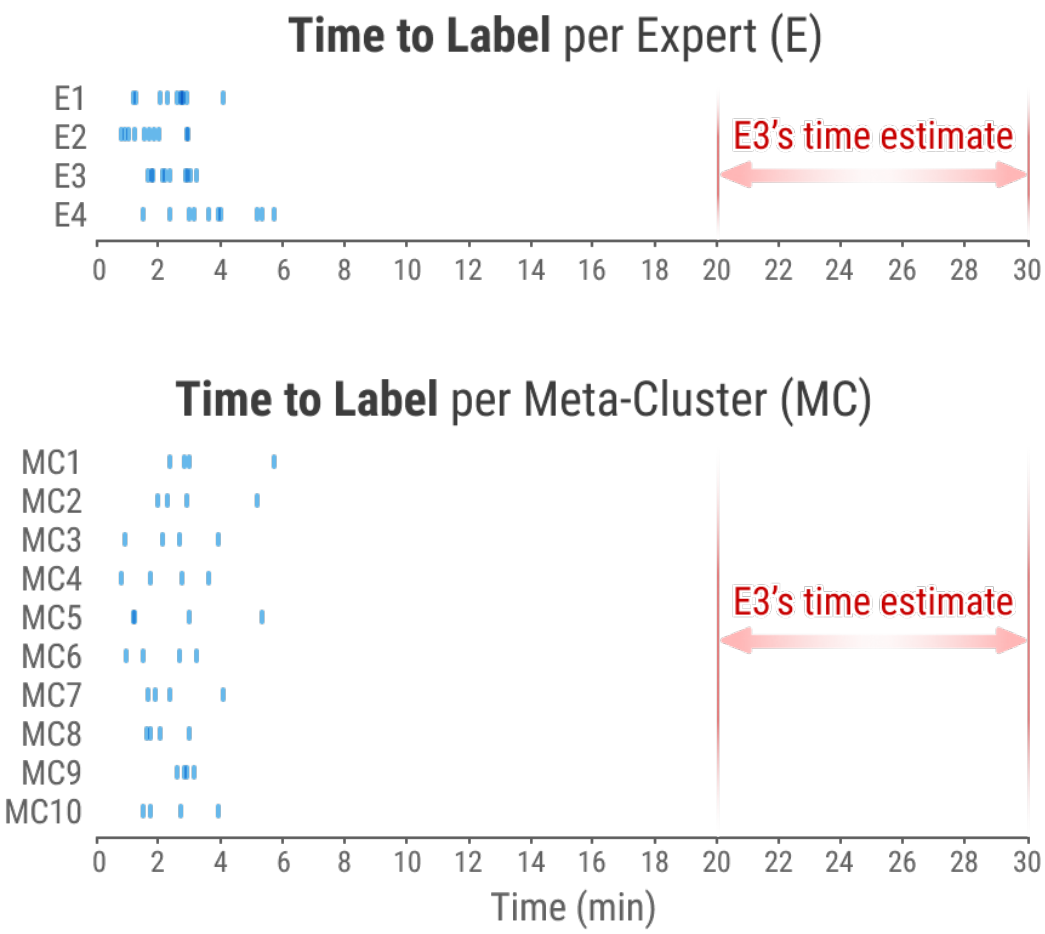


Figure 4.8: TRAFFICVIS **is fast:** Experts consistently need about 2-4 minutes to provide labels, while E3 estimates any other method would take at least 20-30 minutes. (top) labeling times by expert, (bottom) shows labeling times by meta-cluster.

that MC1's most likely label is MASSAGE. Looking at the highest-valued label for each meta-cluster (circles), we see that we ended up with 6 AT-WILL, 3 TRAFFICKING, and 1 MASSAGE.

4.6.6.1 Post-interview Questionnaire

The results of the post-interview questionnaire can be seen in Figure 4.10. We note that all users had a positive experience with TRAFFICVIS and see it implemented in practice.

4.7 Limitations and Future Work

4.7.1 Improvements to Algorithms and UI Design

Experts seemed confident in the results of our algorithms. However, we would like to integrate more features into TRAFFICVIS, such as an automated analysis of the spatial trajectories of

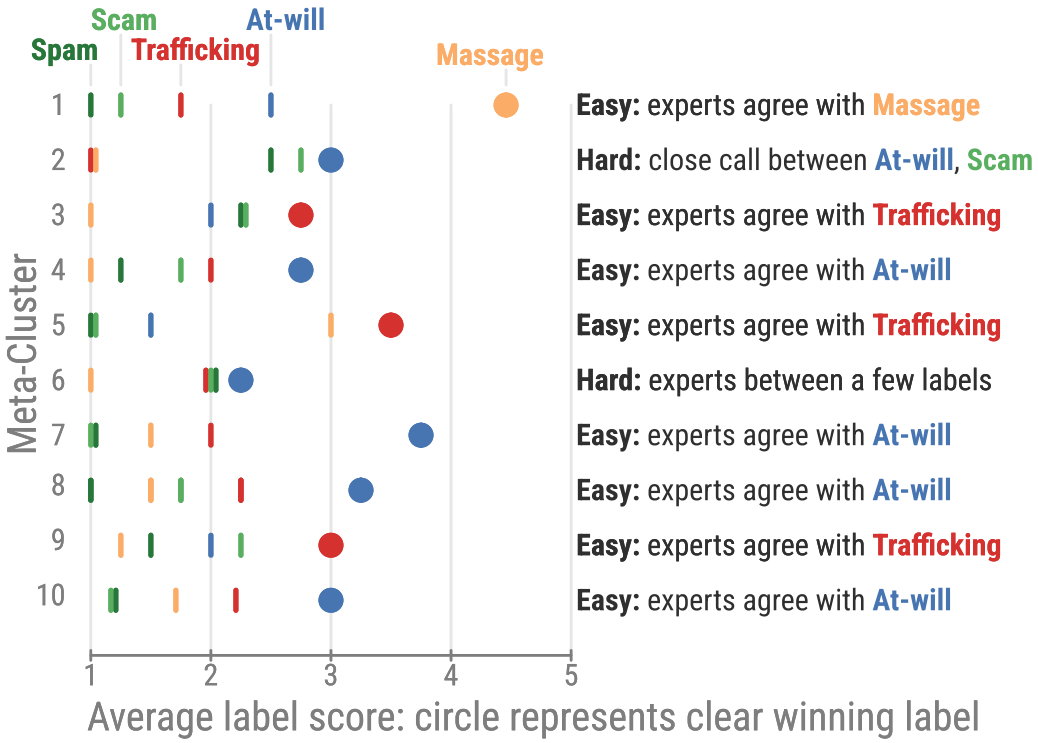


Figure 4.9: **Final labels**: averaged scores among all experts, for each meta-cluster. Circles represent clear winning labels. Experts usually agreed on one label, except for a few meta-clusters that are close calls (2, 6). For both these meta-clusters, at least one expert called it difficult to label based on the given information.

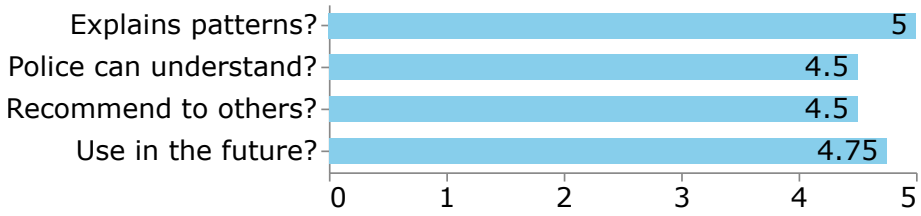


Figure 4.10: **Experts loved** TRAFFICVIS**:** results on a scale of 1 (strongly disagree) to 5 (strongly agree). Full questions can be found in Appendix A.

meta-clusters over time. If a particular meta-cluster shows ads moving across the US over time or circling back to the same few locations again and again, this could be indicative of a traveling HT ring. Experts could use these patterns to better inform their labeling.

While we don't currently have access to sensitive image data, only the hash codes for those images, experts often look at image data to inform their labels. In particular, E3 noted that including image data would have made it easier to label some meta-clusters (i.e. MC2, MC6). If we get access to image data in the future, TRAFFICVIS could analyze it to help experts in various ways. For example, we could estimate the number of distinct people advertised in a

particular meta-cluster, where a large number of possible victims would be indicative of an organized HT ring.

In terms of the UI design, we got two minor pieces of feedback. E2 asked for larger font sizes and E3 was interested in adding an additional label: "possible", which would fall between "3: Unsure" and "4: Likely". Any substantial improvements to UI design would arise if we implemented some of the algorithmic improvements mentioned above.

In the future, we would also like to incorporate more than just escort ads in our analysis. Many ads have connections to social media websites, such as Twitter, Instagram, Facebook, and OnlyFans accounts. We would like to collect some of this data and incorporate it into our algorithms and visualizations. In particular, Instagram and Facebook are the most common platforms for soliciting escort services [83], but their terms of service do not allow for crawling data. It's also against OnlyFans terms of service to crawl data, leaving only Twitter as a possible source. Since we see few Twitter handles mentioned in escort ads compared to the other aforementioned platforms, we believe the benefit of incorporating Twitter data would be marginal at best.

4.7.2 Societal Impact and Practical Use

There has been much backlash in the media as well as academia about the use of black box technologies in policing and investigative efforts [10, 81]. Frequently in these cases, investigative efforts are based on predictions of illegal activity derived from AI or other algorithms that do not provide explainability and that prosecuting attorneys do not understand, which is dangerous to act upon without verification. Since TraffickVis is specifically designed to detect and visualize organized crime groups, it has the potential to be an ideal tool to explain how one arrived at the decision that a case was a part of an organized crime group.

According to E3, TraffickVis could be justifiably used in court because no black box algorithms were utilized. The visual presentation of individual ads in the *Text panel* shows exactly how the ads are connected. For example, in Figure 4.1, we can see that the majority of the ad content is the same, excepting some details such as dates and locations.

One of the largest concerns we have with building algorithms and systems for fighting HT is to make sure that we are not stepping on the liberties of at-will sex workers, who also post escort ads on these websites. While large clusters of text similarity generally signal organized crime groups and not individual workers, we are conscious that they may appear in a meta-cluster. Since TraffickVis does not outwardly classify any meta-clusters as HT cases, only highlighting some possibly suspicious ones, we put the onus on domain experts to make the final decision.

However, even without TraffickVis, an investigator could look at any of these ads online, set up a fake appointment with a real escort worker, and arrest them for prostitution at any time. We have to be very careful about which officers will get access to this software and data, and we are working with *Marinus Analytics* to ensure that anyone with direct access to a running instance of TraffickVis would be highly vetted. Furthermore, we've seen an encouraging trend towards practitioners taking a victim-centered approach to HT; US cities have been decriminalizing prostitution in the past few years [11, 80]. By vetting the practitioners users of TraffickVis to only officials that are clearly invested a victim-centric approach, we

ensure that TraffickVis does not contribute to stepping on the liberties of at-will sex workers. We also have ensured that we have stakeholders in multiple affected populations: not just the perspective of *Marinus Analytics* and practitioners but also of HT survivors by including an HT survivor's perspective throughout the design and feedback of TraffickVis.

4.7.3 Using our labels for downstream tasks.

The labeling design of TraffickVis was intentionally chosen to be flexible for the expert, allowing them to rate on a scale of 1–5 for each label. However, this causes some difficulties for us to post-process these labels before they are used in downstream tasks, particularly because the labels are not disjoint. Downstream classification of meta-clusters will be difficult since the same meta-cluster could be labeled in 25 different ways, and not all differences between labels are insightful – the difference between a '4: Likely' and '5: Very Likely' may not be very meaningful. Furthermore, a meta-cluster could simultaneously be AT-WILL and MASSAGE, or TRAFFICKING and MASSAGE. To handle this, we can do a few different things: (a) threshold the label scores, i.e. an average score of 3.5 or higher indicates the meta-cluster falls under that label, else it does not, (b) choose to not predict certain labels that overlap with others, i.e. MASSAGE, or (c) treat our downstream task as prediction rather that classification.

4.7.4 Reproducibility and Application to Other Domains

Unfortunately, we cannot make the data publicly available to protect the safety of potential victims. However, even with public data, our study could only be reproduced by somebody able to solicit HT experts. Within these parameters, we have done what we can to make TraffickVis reproducible; the code for InfoShield and TraffickVis are open-sourced with synthetic data.

TraffickVis has specifically been designed for labeling suspicious meta-clusters of escort ads for HT and other organized activity. However, TraffickVis could be applied, with small modifications, as a cluster labeling solution for other domains. For example, coordinated disinformation campaigns on social media have become a pervasive issue in the last few years [22, 105], causing many tech companies to implement algorithms to find and flag suspicious users online [38]. One could use TraffickVis's pipeline to quickly label clusters of similar social media posts, using relevant metadata such as images and usernames. Additionally, we may be able to use InfoShield directly, as it was also successful finding organized, bot-like behavior in Twitter data (see Chapter 3.6).

4.8 Conclusion

Facilitating the retrieval of high-quality labels for complex, multimodal data can be a challenging task. TraffickVis is a system designed to visualize this type of data for the HT problem, making the following major contributions:

1. **High-impact,** being accessible to a variety of anti-HT stakeholders, including criminologists, domain experts, and investigators (see Section 4.6);

2. **Label generation**, finally providing a way to generate high-quality cluster labels, which will be used for further algorithm development;
3. **Time-saving**, granting a huge speedup over manual labeling, according to feedback from domain experts.

TraffickVis shows that even with such complex data, we can still design an interface that lets domain experts quickly see big patterns while simultaneously allowing them to drill down into specific entries when needed.

Through the process of soliciting expert feedback, we naturally curated a dataset labeled by TraffickVis that will enable further algorithm development towards M.O. detection, allowing investigators to quickly find meta-clusters of ads that actually represent real HT cases. Domain experts and *Marinus Analytics* can use TraffickVis to label additional micro-clusters over time and more quickly find new patterns in known M.O.s. This process can also enable researchers to continually develop and evaluate M.O. detection algorithms, spam filters, and more, as we see emerging trends in escort ads over the years to come.

Reproducibility: The code and synthetic data is open-sourced at https://github.com/catvajiac/TrafficVis.

Chapter 5

TrafficBoard: Visualization for Kickstarting Investigations

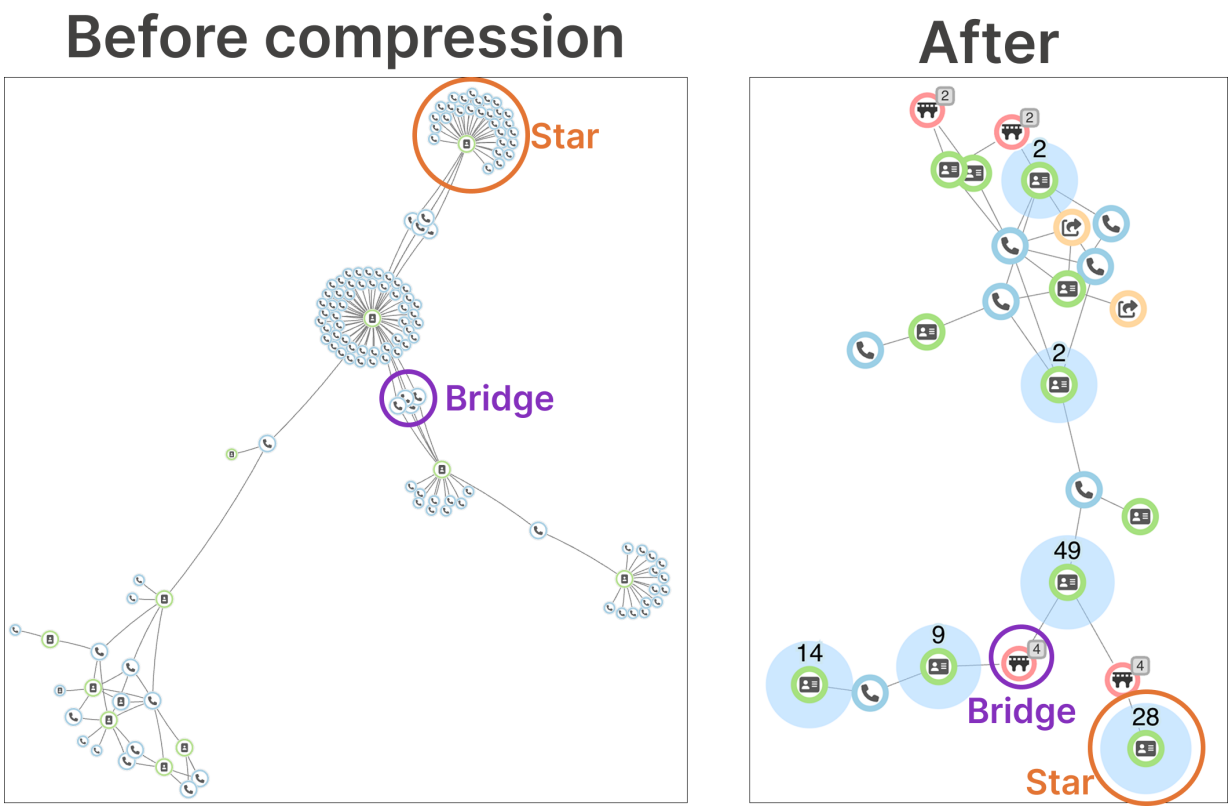


Figure 5.1: TrafficBoard improving evidence graph visualization.

This chapter is based on preliminary work presented at IEEE VIS 2023 [106], followed by consulting work done with Marinus Analytics to improve evidence graph visualization. The results are being implemented into TrafficJam, their main software platform.

5.1 Problem and General Approach

Analysts, such as investigators, criminologists, or NGO workers, often utilize connections between metadata to analyze suspicious clusters. Our previous work on HT evidence visualization (Chapter 4) has stayed away from node-link diagrams, as these graphs contain very densely connected regions that make them unreadable. However, investigators have been using similar visualization techniques for decades, even creating analog versions called Anacapa charts [49, 98]. As such, Marinus Analytics is trying to add evidence-graph visualization to their software platform, TrafficJam.

This chapter describes the formative study we ran with users of TrafficJam to better understand how they use evidence graphs and the resulting graph-visualization, TrafficBoard, which has the following advantages.

1. **Formative Study with Actual Practitioners:** To guide the design of TrafficBoard, we ran a formative study with 10 actual users of TrafficJam across North America and Europe to better understand what role graphs play in their analysis, and where bottlenecks with graph-based analysis in TrafficJam currently are.
2. **Intuitive Graph Summarization & Layout:** After identifying the bottlenecks, we address the two immediately actionable problems: reducing cognitive overwhelm and improving scalability, by summarizing and visualizing the evidence graph greedily using known substructures.

Impact: This work is currently being implemented into TrafficJam. The code can be found at https://github.com/catvajiac/graphvis.

5.2 Previous Attempts for Evidence Graph Visualization

Previously, Marinus Analytics implemented a three-dimensional representation of node-link diagrams in an attempt to mitigate edge congestion [25] through interactivity. More specifically, this tool relied on users panning and zooming to change the camera's field of view, but many practitioners still struggled to determine which nodes were truly connected to each other, leading to the tool's eventual deprecation.

Node-link diagrams are widely used for graph visualization tasks, but face significant scalability challenges as the size of the network increases [24]. A primary issue is edge congestion, often referred to as the "hairball effect", where excessive edge crossings and node occlusions result in a visually impenetrable mass that obscures the underlying topological structure [7], which grows disproportionately with graph size. While alternative visualization methods such as adjacency-matrix representations [16, 24] avoid the hairball effect, practitioners prefer node-link diagrams due to their familiarity with this representation in other tools, such as i2 Analyst Notebook [53].

5.3 Formative Study with Actual Marinus Users

How can we best understand what kinds of problems practitioners are facing with graph visualization in Marinus' TrafficJam software? We ran a formative study with 10 current users of TrafficJam across the United States, Canada, the United Kingdom, and Ireland to determine how they currently use graph-based visualization tools in their analysis and what pitfalls they run into. We first describe the study design in more detail, then summarize the results.

5.3.1 Study Design

In conjunction with domain experts, we first determined a set of three high-level Questions (hereby referred to as **Q1** – **Q3**) to better understand the use of graph-based tools in HT detection.

Q1. **Understanding the Current Workflow:** How are evidence graphs useful to practitioners in HT detection? How should the graphs be constructed?

Q2. **Determine Context of Tool Use:** Are graph-based visualizations used solely as an exploratory tool, or are they also used to summarize results to others? Who else might interact with these tools?

Q3. **Identify Bottlenecks:** What is limiting about current graph-based tools? What added functionality would help you the most?

To answer each question, we designed a set of 12 targeted prompts for participants to answer during hour-long interviews, which were then refined after a mock interview with an additional domain expert. Table 5.1 summarizes each prompt and which Q1 – Q3 it was designed to answer. The full phrasing of these questions can be found in Appendix B.

	Summary of Prompt	**Q1**	**Q2**	**Q3**
1	Why use graphs? What node types are useful?	✓		✓
2	What tools do you currently use?	✓	✓	
3	What are the pain points?	✓		✓
4	What patterns indicate HT?	✓		✓
5	How are tools used in court / beyond investigation?		✓	
6	Examples of previously prosecuted cases?	✓	✓	✓
7	Are false links a large issue?	✓		✓
8	Are clusters of nodes interesting?	✓		✓
9	What makes one node more relevant than another?			✓
10	Is data the bottleneck for pursuing cases?		✓	✓
11	Add additional data to graph?	✓		✓
12	Collaboration with other practitioners?	✓	✓	

Table 5.1: **Answering High-Level Questions:** A summary of each prompt and its corresponding Questions. The full phrasing of these questions can be found in Appendix B.

5.3.2 Takeaways from Practitioners: Answering Q1– Q3

Practitioners had much to say about how they use graphs in their work, as well as other useful pieces of information about how they search for and prosecute HT. We summarize these findings in two parts: domain-specific takeaways and graph-specific takeaways.

5.3.3 Domain-Specific Takeaways

We summarize domain-specific takeaways into five main themes.

1. **HT in North America looks very different to HT in Europe (addr. Q1, Q2, Q3).** While the majority of HT in North America is domestic, the opposite is true in Europe, with the majority of victims being foreign-born and therefore, uncomfortable talking to government officials or law enforcement. To help mitigate this issue, many police forces are employing liaisons for at-will workers and HT victims to better serve these populations.
2. **Government relationships with at-will workers greatly vary per country, affecting fighting HT (addr. Q2, Q3).** Both the United Kingdom and Ireland seemed to have more positive relationships with at-will workers than in North America. Specifically in Ireland, wellness checks, free STD screenings and contraception are offered, and at-will workers have government representatives to call if they need assistance.
3. **Once a lead is found, lack of data isn't the bottleneck – the legal system is. (addr. Q2, Q3)** Once a lead is generated and an investigation started, practitioners can usually easily find evidence through privileged information, such as cell-phone and email records, bank statements, and more. Unfortunately, the real bottleneck comes in the legal system, where most cases are never prosecuted, or if they are, it's only using the more "straightforward" crime of money laundering, which almost always co-occurs with HT, but does not require survivors to testify. While "victimless prosecution" is theoretically possible in the United Kingdom, investigators lament that practically, no case is prosecuted without survivors testifying, which can be difficult and dangerous for some survivors.
4. **Signals of SPAM in North America are signals of TRAFFICKING elsewhere. (addr. Q1, Q2, Q3)** Surprisingly, practitioners in the United Kingdom described what we would describe as SPAM in North America, namely the same phone number advertised in multiple locations per day, to represent TRAFFICKING in the UK. This likely occurs because the physical distances are smaller between major cities in the UK, compared to North America.
5. **Geographic Movement Over Time is suspicious (addr. Q1, Q3).** Investigators in the UK and Ireland mentioned that victims are cycled between properties in various locations, which are often owned by the traffickers themselves. In particular, they look for movement in and out of rural areas, where the presence of new people advertising online can be more easily spotted.

5.3.4 Graph Visualization Takeaways / Design Goals

In addition to the domain knowledge we gained, we identified three main takeaways (hereby referred to as **T1**– **T3**).

T1. 7 / 10 participants: Graphs are overwhelming (addr. Q1, Q3) Participants struggle to interpret patterns with so many nodes and edges on the screen, causing many to ignore the feature. While multiple previous solutions were tried, including a 3d interactive graph model, these ended up being overly confusing for participants, particularly for such densely connected subregions of graphs.
Our remedy: With a combination of graph summarization methods and clever visualization techniques, we can reduce the cognitive overwhelm on the practitioner.

T2. 6 / 10 participants: Scalability makes visualization infeasible (addr. Q1, Q3). The current tool cannot process and does not visualize graphs above a certain size, particularly affecting participants in densely populated regions of the UK, Canada, and the US where evidence graphs tend to be larger.
Our remedy: Use graph summarization to reduce the number of nodes needed to be shown.

T3. 5 / 10 participants: Better integration with TrafficJam filters (addr. Q1, Q2, Q3). Participants currently struggle to understand how nodes in an evidence graph correspond to other visualizations in TrafficJam.
Our remedy (ongoing): Ongoing engineering effort from Marinus Analytics, as it includes other parts of TrafficJam.

Since **T3** is a software engineering effort, we focus on addressing **T1** and **T2** in the remainder of this chapter.

5.4 Resulting Interface: TrafficBoard

How can we (a) quickly summarize evidence graphs for TrafficJam on-the-fly and (b) improve visual encodings to mitigate expert overwhelm? Upon construction of these evidence graphs, we notice two clear structure types in the data, namely, bridges and stars. Pictoral representations of both structures can be found in Figure 5.2.

- *bridge supernodes*, representing a group of nodes who all only connected to the same two neighbors, and
- *star supernodes*, representing a group of nodes that only connect to one central node.

Since these structures are so well-defined, we can greedily look for and compress nodes, processing them in order of degree.

5.4.1 Visualization Techniques: Stars as "Halos"

Instead of representing each individual node in a star, we instead use a "halo" structure surrounding the supernode. The halo maintains a similar look to the fanning out of individual

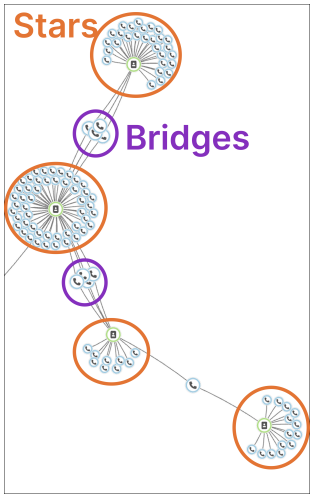


Figure 5.2: **Evidence graphs show clear structure:** because of the way Marinus Analytics is constructing these graphs, there are clear structures that can be summarized, namely, stars and bridges.

nodes in a force-directed layout, keeping it intuitive for experts, but vastly reduces the number of nodes needed to be drawn. Halo size scales logarithmically with the number of nodes it represents. Colors are based on the types of nodes compressed.

5.5 Ongoing Implementation & Recommended Evaluation

While TrafficBoard addresses takeaways **T1** and **T2**, we cannot satisfy **T3** or properly test the graph layout's improvements to Marinus' TrafficJam before integration, which requires a long-term engineering effort which is ongoing as of the writing of this thesis. Once TrafficBoard has been integrated, we recommend running a Randomized Control Trial (RCT) with the formative study participants.

5.5.1 Recommended Study Design

To evaluate the utility of TrafficBoard as a component of Marinus' TrafficJam software, we recommend a within-subjects design where each participant completes two blocks of cluster investigation, one using TrafficJam alone, and the other using TrafficJam+TrafficBoard. Meta-clusters from known HT groups will be split into two non-overlapping datasets, dataset A and dataset B. Participants are then randomly assigned to one of two groups: Group 1 uses TrafficJam alone to analyze dataset A and TrafficJam+TrafficBoard to analyze dataset B, while Group 2 uses TrafficJam+TrafficBoard to analyze dataset A, then TrafficJam alone to analyze dataset B, labeling the meta-clusters. Then, we can analyze the final labels to determine (a) if TrafficJam+TrafficBoard helps practitioners be more effective in labeling clusters, and (b) if TrafficJam+TrafficBoard helps practitioners more quickly identify potential cases.

This study design has the following advantages:

- **Neutralizes “Task Learning”:** If TrafficBoard is inherently helpful, the participants in Group 1 should show a significant jump in accuracy when adding TrafficBoard. Conversely, if TrafficBoard is confusing, Group 2 might actually perform better when they switch back to the previous interface.
- **Reduces “Participant Noise”:** Some investigators are naturally more "eagle-eyed" than others. By having the same person use both tools, you measure the delta in their individual performance rather than comparing two different people.

While TrafficBoard is still being implemented into Marinus Analytics’ software, TrafficJam, initial feedback from experts has been optimistic, leaving us hopeful about its potential to simplify case-building using evidence graphs for practitioners across North America and Europe.

Chapter 6

Discussion

6.1 Ethical Considerations

While proactive approaches can help victims get out of exploitation, these tools will only be effective when used by governments with a *victim-centric* and *at-will worker-centric* approach. While the algorithms described in this thesis are publicly available for research use, data access is extremely restricted to ensure that any personal information about those advertised is not used irresponsibly. To mitigate potential harm, Marinus Analytics only provides data access to agencies, governments, and researchers that consistently show this commitment.

6.2 Beyond HT Detection: Using InfoShield for Survivor Corroboration

While the scope of this thesis has been Trafficking Detection, as it is the most visible pain point, we must ensure anti-HT efforts eventually extend beyond this phase to address the root causes of this complex social issue, as well as help survivors rebuild their lives once they are out of exploitation. In Figure 6.1, we identify two additional phases that victims and survivors go through where technology could help.

Figure 6.1: **The Trafficking Pipeline doesn't start or stop with detection:** while beyond the scope of this thesis, there are other parts of this complex social issue that we can focus on.

6.2.1 Ongoing Extension: Towards Rebuilding

In conjunction with collaborators at McGill University and Montreal-based NGO, La Sortie [62], InfoShield is being used as part of a search / retrieval-based interface that survivors can use to gather corroborating evidence towards them being trafficked, which is used in Canada for debt forgiveness and other social programs. More specifically, a survivor provides a series of dates, locations, phone numbers, or any other information they remember, which are matched with InfoShield micro-clusters to help survivors find the advertisements describing them. This corroboration can then be used to void any financial penalties survivors might face for not having paid bills, etc while being exploited.

6.2.2 Future Direction: Towards Stopping Recruitment

This thesis specifically focuses on finding HT once the victims are already being trafficked, primarily due to the available data and domain experts. However, there are other problems to solve in this space, e.g., what if we tried to stop *recruitment* of victims into HT?

In the US, recruitment commonly happens through social media, often through private messaging through Facebook, Instagram, and Snapchat, [102]; the issue is so pervasive that Meta was fined for recruitment of a minor into HT through Facebook Messenger [54].[1]

Social media companies, since they have access to this data, could extend content moderation efforts to include flagging behavior or accounts that exhibit predatory or grooming behavior to limit the reach of traffickers to their intended targets.

Relevant Research Questions.

With access to social media posts or private messaging data, how could corporations with HT recruitment issues flag users that are potential traffickers, so that they can be manually verified and banned from the platform, reducing their potential for harm? How should they incorporate the data's multimodality (posts, comments, likes, images, video, etc.)?

6.2.3 Future Direction: Incorporating additional data into finding HT.

InfoShield and DeltaShield were designed for unstructured text documents because they create the best data source for suspicious signals of HT that is *practically* available at the time of writing this thesis. With the addition of privileged data (i.e. cell-phone records / transcripts, email content, bank statements etc), we could likely provide higher quality evidence to experts when examining suspicious cases. More feasibly, it may be possible to incorporate data from social media websites into our methods, since they are known hubs for both recruitment and advertising of HT [103]. However, as of the writing of this thesis, it is infeasible for us to crawl

[1]Traffickers often look at social media websites to find people in vulnerable positions financially and socially, i.e. that have few social ties to family and are not stably housed, then use those websites to recruit these people, often under false pretenses of a relationship or job opportunity.

this data ourselves as it violates many platforms' Terms of Services, or in the case of OnlyFans, is financially prohibitive.

According to our domain experts, there are certain features of the images that can be indicative of HT, such as the presence of a the same brand or tattoo on multiple HT victims. As of the time of writing, Marinus Analytics has not provided images because of the danger of de-anonymizing possible victims or sex workers, as well as the legal complications of the university storing these images on their servers, particularly since some images contain Child Sexual Abuse Material (i.e. explicit photos of minors). With these issues resolved, it is possible that the addition of images could vastly improve our algorithms. However, it will likely introduce yet another complication – using images might also spuriously connect advertisements that should not be connected, as our experts have mentioned that some posters on escort websites will steal images from other ads / use common photos, which will need to be handled.

Relevant Research Questions.

1. If we had access to private information such as cell-phone records or bank statements, how can we augment InfoShield or DeltaShield to include this data? How do we augment the current text-based clustering to include non-text features that can be generated from private data?
2. If we had access to social media websites, how could we incorporate the multimodal data (i.e. text, images, videos, comments) into InfoShield and DeltaShield?
3. If we had access to images, what features will we try to extract? Should we connect advertisements with images that seem to advertise the same person? Same tattoo? Same hotel background? How do we extract those entities from images?
4. How would any of these data sources impact the ranking metric in InfoShield and DeltaShield, which is currently primarily based on text compression efficacy?
5. If we did have access to these data sources, how could we incorporate them into TraffickVis and TraffickBoard? Should image data be directly visible in the tools, or would statistics on extracted features be enough for the expert to label a case or start an investigation?

6.3 Generative AI for Human Trafficking Detection

Large language models (LLMs) are being applied to a variety of problems with great success. As of the writing of this thesis, the potential application to the HT domain has been limited, particularly since most LLMs include guardrails against generating explicit content for obvious safety reasons. Even when guardrails aren't an issue, performance isn't always as intended when models aren't trained on escort ad data; while we used previous versions of ChatGPT to generate a synthetic HT dataset to promote academic reproducibility in the HT domain [77], the ads generated read more like mini romance novels than actual escort advertisements. In

addition, the dataset can no longer be generated using the same prompts with later GPT models, due to changes in guardrails since the dataset was published.

While practitioners have concerns towards using LLMs directly for HT detection due to explainability issues and the chance of hallucinations of "hard evidence", we could envision LLMs being used to assist in summarizing or annotating already-found evidence in visualizations or dashboards, pointing practitioners to the specific characteristics of a meta-cluster that they might be most interested in. More generally, for high stakes applications like HT detection, we recommend only including generative models where humans are directly in-the-loop and have the available data to quickly and easily refute a potential hallucination.

6.4 Practical Lessons Learned

In the process of working closely with stakeholders on a variety of projects, including the chapters in this thesis, we encountered many stumbling blocks that are common with real-world problems. We summarize a few lessons learned in dealing with these stumbling blocks below.

Lesson #1: Talking to practitioners and stakeholders often is vital, even about lower-level details. We can almost always learn something new.

It can be tempting, after initially formulating the research problem with practitioners, to iterate on possible solutions with other researchers, and only include practitioners once the work is more finalized, especially when lower-level details are difficult to communicate. However, including practitioner's regular feedback has two advantages: not only can practitioners help guide design decisions, researchers also directly benefit from trying to summarize the current state of the work to non-experts. Personally, I often found it worked similarly to "rubber-duck debugging".

Lesson #2: No labeled data isn't a non-starter, but it might require creative solutions.

Even for unsupervised approaches, labels are still eventually needed for proper evaluation. Unfortunately, many impactful, real-world problems don't come with gold-standard labeled datasets, but this does not mean we cannot make progress towards these problems, we might just need to alter our approach, as we did with TraffickVis in this thesis.

Lesson #3: Custom, "simpler" models are sometimes more appropriate for a real-world problem.

While state-of-the-art ML models may work for many tasks, they may not always be appropriate, especially in compute-constrained environments. In our case, we found state-of-the-art text embedding methods to be less performant than InfoShield, not to mention less scalable. In addition, in high-stakes environments, concern for potential harm, complex models may not be preferred by practitioners if they do not meet desired interpretability criteria.

Chapter 7

Conclusion

This thesis creates a pipeline enabling analysts, social workers, and investigators to (a) find possible cases of human sex trafficking (HT) using publicly accessible data,(b) understand why the lead is suspected HT, and (c) quickly decide whether the case is worthy of further inspection. We identified and addressed four practical challenges with HT detection:

*PC1: Dirty Data breaks standard assumptions of online text content.

*PC2: Few/Expensive Labels are difficult to procure for HT.

*PC3: Legal/Financial Limits affect possible model choices.

*PC4: Expert interpretability makes it difficult for experts to incorporate models.

Additionally, through the lens of finding HT online, this thesis makes the following contributions to the intersection of applied machine learning, interactive data visualization:

Part 1: Algorithms for finding suspected HT in escort ads

Chapter 3: InfoShield and DeltaShield find microclusters of suspicious ads, outperforming the state-of-the-art all while addressing a more realistic setting.

- Addressing Challenges:
  - InfoShield is unsupervised (addr. *PC1: Data),
  - uses principled methods that are more easily interpretable (addr. *PC3: Legal), and
  - provides a summary template for each microcluster of similar ads (addr. *PC4: Interpret).
- Impact:
  - In the news: InfoShield highlighted on local Pittsburgh news channel WPXI, Carnegie Mellon University, and McGill University.
  - InfoShield is also being integrated by *Marinus Analytics*.

Part 2: Visualization for data labeling and interpretation

Chapter 4: TraffickVis provides a **10x** speedup in labeling time vs. manual labeling (addr. *PC3: Legal), the previous standard and was highly rated in expert feedback.

- Addressing Challenges:
  - By carefully visualizing results from InfoShield with other relevant data in an interactive dashboard (addr. *PC1: Data, *PC4: Interpret),
  - practitioners can finally see all the information available in the ads that can help them label the cluster (addr. *PC2: Labels) as HT, at-will, or other.
  - TraffickVis provides a **10x** speedup in labeling time vs. manual labeling (addr. *PC3: Legal), the previous standard and was highly rated in expert feedback.
- Impact:
  - Best Poster Honorable Mention VIS 2021.
  - Best Paper Honorable Mention VIS 2022.
  - TraffickVis was used to curate a small labeled dataset that can be used for evaluation.

Chapter 5: TraffickBoard summarizes connections between pieces of evidence in a particular case to help practitioners kickstart their consideration or possible investigation of a potential case.

- Addressing Challenges:
  - After a formative study of 14 practitioners across 4 countries in North America and Europe (addr. *PC3: Legal),
  - TraffickBoard summarizes and redesigns standard graph-based layouts to help interpretability of HT evidence graphs (addr. *PC4: Interpret)
  - and uses interactivity to help users more efficiently explore graph-based evidence data, which is error-prone (addr. *PC1: Data).
- Impact:
  - TraffickBoard is currently being integrated by *Marinus Analytics*.

This thesis shows that the intersection of ML and visualization can lead to tools that have massively positive impacts on systemic problems in today's society, as long as these tools are crerated through a **human-forward** approach that includes stakeholders at every step of the process.

Appendix A

TrafficVis Questions

[5 mins] Introduction and basic parameters.

Verify the following points:

1. the participant has read the offline document and has no further questions
2. the participant knows they will be recorded, and short quotes (related to their feedback on the user study only) might be included in the results section of the paper

[10 mins] Introductory questions

1. How many years have you studied human trafficking?
2. On a scale of 1: not an expert to 5: expert – what is your level of comfort with looking at online escort ads for signs of human trafficking / other organized activity?
3. On a scale of 1: not an expert to 5: expert – what is your level of comfort with AI / clustering algorithms?
4. Do you actively look for cases yourself? If so, what is your current process for finding these cases? Are there any particular algorithms / techniques that you use?
5. (If applicable) In your experience with police officers / federal law enforcement, how do they look for cases?

[30 mins] Using the interface to label

- Ask participants to label all 10 clusters
- Tell them to "speak aloud": asking any questions / commenting as they go

[15 mins] Conclusion

Open-ended questions:

1. What were your top three favorite things about the tool?
2. What three features would you like changed / added?

3. Do you find the tool easy to use? Are there any sticking points?
4. How do you see this tool being used in practice?

Offline questions (to be filled in Google form):

1. How likely are you to use this software in the future?
2. Would you recommend it to a colleague?
3. How well does this tool help explain observed patterns?
4. How well do you think this tool will help a police officer, defense attorney, etc understand these clustering results?

Appendix B

Formative Study Questions

Note: to more closely match the language practitioners use, instead of the terms *graph*, *node*, and *edge*, these questions use the terms *link chart*, *entity*, and *link*, respectively.

1. Why use link charts? Which types of entities actually help with building a case?
2. How are you currently looking at those connections? Which tools (TrafficJam, i2) do you use?
3. What are the pain points you see when investigating links between entities?
4. What patterns are most indicative of human sex trafficking? Does the timeline of entities play a factor? Does the geographical spread of entities matter?
5. What type of supporting documentation is needed in court that relates to link charts? Do you use them in court at all?
6. Are there examples of ground truth you can provide, i.e. past cases that have already been prosecuted?
7. Should a connection between entities ever be dropped/ignored? E.g., if a connection is older, or false?
8. Do you look for subgroups (clusters) of entities within a particular case? Could these groups overlap or would they be separate?
9. Do you care about all the entities equally? What makes an entity more important? If it's well-connected (hub)? Are there entities that are similar enough to be combined?
10. How common is it that you don't have enough data to pursue a case?
11. What data sources do you use (other than data from TrafficJam)? Public, private?
12. How much collaboration is there with other practitioners when investigating a lead? With other police officers? With other domain experts or stakeholders?

Bibliography

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, pages 859–868. Computer Vision Foundation / IEEE Computer Society, 2018. 49

[2] Faraz Ahmed and Muhammad Abulaish. A generic statistical approach for spam detection in online social networks. *Comput. Commun.*, 36(10-11):1120–1129, 2013. 34, 36

[3] Hamidreza Alvari, Paulo Shakarian, and J. E. Kelly Snyder. A non-parametric learning approach to identify online human trafficking. In *ISI*, pages 133–138. IEEE, 2016. 50

[4] Hamidreza Alvari, Paulo Shakarian, and J. E. Kelly Snyder. Semi-supervised learning for detecting human trafficking. *Security Informatics*, 6(1):1, 2017. 14

[5] Hamidreza Alvari, Paulo Shakarian, and JE Kelly Snyder. Semi-supervised learning for detecting human trafficking. *Security Informatics*, 6(1):1, 2017. 7, 48

[6] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: ordering points to identify the clustering structure. In *SIGMOD*, pages 49–60, 1999. 15, 17

[7] Benjamin Bach, Nathalie Henry Riche, Christophe Hurter, Kim Marriott, and Tim Dwyer. Towards unambiguous edge bundling: Investigating confluent drawings for network visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):541–550, January 2017. 68

[8] Geoffrey J Barton and Michael JE Sternberg. A strategy for the rapid multiple alignment of protein sequences: confidence levels from tertiary structure comparisons. *Journal of molecular biology*, 198(2):327–337, 1987. 15

[9] Regina Barzilay and Lillian Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL*, 2003. 15, 17

[10] Yavar Bathaee. The artificial intelligence black box and the failure of intent and causation. *Harvard Journal of Law & Technology*, 31:889, 2018. 64

[11] Juliana Battaglia. Baltimore will no longer prosecute drug possession, prostitution and other low-level offenses. https://www.cnn.com/2021/03/27/us/baltimore-prosecute-prostitution-drug-possession/index.html, 2021. 64

[12] David Beil and Andreas Theissler. Cluster-clean-label: an interactive machine learning approach for labeling high-dimensional data. In *VINCI*, pages 5:1–5:8. ACM, 2020. 49

[13] J. Bernard, Eduard Dobermann, Anna Vögele, Björn Krüger, Jörn Kohlhammer, and Dieter W. Fellner. Visual-interactive semi-supervised labeling of human motion capture data. In *Visualization and Data Analysis*, 2017. 49

[14] J. Bernard, Matthias Zeppelzauer, Michael Sedlmair, and Wolfgang Aigner. A unified process for visual-interactive labeling. In *EuroVA EuroVis*, 2017. 49

[15] Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Dieter W. Fellner, and Michael Sedlmair. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE Trans. Vis. Comput. Graph.*, 24(1):298–308, 2018. 48, 49

[16] Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, Madison, WI, 1983. Translated by William J. Berg. 68

[17] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146, 2017. 15, 34

[18] Vanessa Bouche. *An empirical analysis of the intersection of organized crime and human trafficking in the United States*. National Criminal Justice Reference Service, Office of Justice Programs, 2017. 1

[19] Stephen E. Boyd. Letter from Stephen E. Boyd, Assistant Attorney General, U.S. Department of Justice, to Robert W. Goodlatte, Chairman, House Committee on the Judiciary. U.S. Department of Justice Views Letter, February 2018. Regarding H.R. 1865, Allow States and Victims to Fight Online Sex Trafficking Act of 2017. 7

[20] Sergey Brin, James Davis, and Héctor García-Molina. Copy detection mechanisms for digital documents. In *SIGMOD*, page 398–409, 1995. 15

[21] Thomas Brinkhoff, Hans-Peter Kriegel, and Bernhard Seeger. Efficient Processing of Spatial Joins Using R-Trees. In *SIGMOD*, pages 237–246, 1993. 15

[22] David A. Broniatowski, Amelia Jamison, SiHua Qi, Lulwah Alkulaib, Tao Chen, Adrian Benton, Sandra Crouse Quinn, and Mark Dredze. Weaponized health communication: Twitter bots and russian trolls amplify the vaccine debate. *American Journal of Public Health*, 108:1378–1384, 2018. 65

[23] Michael D. Buhrmester, Tracy Nai Kwang, and Samuel D. Gosling. Amazon's mechanical turk. *Perspectives on Psychological Science*, 6:3 – 5, 2011. 48

[24] Michael Burch, Kiet Bennema Ten Brinke, Adrien Castella, Sebastiaan Peters, Vasil Shteriyanov, and Rinse Vlasvinkel. Dynamic graph exploration by interactively linked node-link diagrams and matrix visualizations. *Visual Computing for Industry, Biomedicine, and Art*, 4(1):1–14, 2021. 68

[25] M Sheelagh T Carpendale and X Rong. Examining edge congestion. In *CHI'01 Extended Abstracts on Human Factors in Computing Systems*, pages 115–116, 2001. 68

[26] Deepayan Chakrabarti, Spiros Papadimitriou, Dharmendra S. Modha, and Christos Faloutsos. Fully automatic Cross-associations. In *KDD*, 2004. 16

[27] Mohammad Chegini, J. Bernard, Philip Berger, Alexei Sourin, Keith Andrews, and Tobias Schreck. Interactive labelling of a multivariate dataset for supervised machine learning

using linked visualisations, clustering, and active learning. *Vis. Informatics*, 3:9–17, 2019. 49

[28] Min Chen and Amos Golan. What may visualization processes optimize? *IEEE Trans. Vis. Comput. Graph.*, 22(12):2619–2632, 2016. 49

[29] Edwin KP Chong and Stanislaw H Zak. *An introduction to optimization*. John Wiley & Sons, 2004. 25

[30] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *WWW*, page 963–972, 2017. 14, 33, 35, 43

[31] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. Dna-inspired online behavioral modeling and its application to spambot detection. *IEEE Intell. Syst.*, 31(5):58–64, 2016. 13, 34, 36

[32] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Botornot: A system to evaluate social bots. In *WWW*, page 273–274, 2016. 13, 14, 34, 36

[33] Michael Desmond, Michael J. Muller, Zahra Ashktorab, Casey Dugan, Evelyn Duesterwald, Kristina Brimijoin, Catherine Finegan-Dollak, Michelle Brachman, Aabhas Sharma, Narendra Nath Joshi, and Qian Pan. Increasing the speed and accuracy of data labeling through an AI assisted interface. In *IUI*, pages 392–401. ACM, 2021. 49

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019. 15

[35] Chris H. Q. Ding and Xiaofeng He. Principal component analysis and effective k-means clustering. In *SIAM DM*, pages 497–501, 2004. 15, 17

[36] Artur Dubrawski, Kyle Miller, Matthew Barnes, Benedikt Boecking, and Emily Kennedy. Leveraging publicly available data to discern patterns of human-trafficking activity. *Journal of Human Trafficking*, 1(1):65–85, 2015. 50

[37] John J. Dudley and Per Ola Kristensson. A review of user interface design for interactive machine learning. *ACM Trans. Interact. Intell. Syst.*, 8(2), June 2018. 50

[38] Clare Duffy. Youtube is cracking down on anti-vaccine misinformation. https://www.cnn.com/2021/09/29/tech/youtube-vaccine-misinformation/index.html, 2021. 65

[39] Sara Elmanarelbouanani and Ismail Kassou. Authorship analysis studies: A survey. *Int. Journal of Computer Applications*, 86, 12 2013. 15

[40] Saeideh Shahrokh Esfahani, Michael J. Cafarella, Maziyar Baran Pouyan, Gregory J. DeAngelo, Elena Eneva, and Andy E. Fano. Context-specific language modeling for human trafficking detection from online advertisements. In *ACL*, 2019. 7, 48

[41] Saeideh Shahrokh Esfahani, Michael J. Cafarella, Maziyar Baran Pouyan, Gregory J. DeAngelo, Elena Eneva, and Andy E. Fano. Context-specific language modeling for human trafficking detection from online advertisements. In *ACL (1)*, pages 1180–1184. Association for Computational Linguistics, 2019. 14

[42] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996. 15, 17

[43] Amy Farrell, Jack McDevitt, and Stephanie Fahy. Understanding and improving law enforcement responses to human trafficking, final report. *US Department of Justice*, 2008. 1

[44] Karën Fort, Gilles Adda, and Kevin Bretonnel Cohen. Last words: Amazon mechanical turk: Gold mine or coal mine? *Computational Linguistics*, 37:413–420, 2011. 48

[45] Maria Giatsoglou, Despoina Chatzakou, Neil Shah, Alex Beutel, Christos Faloutsos, and Athena Vakali. Nd-sync: Detecting synchronized fraud activities. In *PAKDD (2)*, volume 9078, pages 201–214, 2015. 14

[46] Nathaniel Gleicher. *How We Respond to Inauthentic Behavior on Our Platforms: Policy Update*, 2019. 14

[47] Peter Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007. 16

[48] A. Guttman. R-Tree : A dynamic Index Structure for Spatial Searching. In *SIGMOD*, pages 47–57, Boston, MA, 1984. 15

[49] Walter R Harper and Douglas H Harris. The application of link analysis to police intelligence. *Human Factors*, 17(2):157–164, 1975. 68

[50] Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954. 14

[51] Andreas P. Hinterreiter, Peter Ruch, Holger Stitz, Martin Ennemoser, Jürgen Bernard, Hendrik Strobelt, and Marc Streit. Confusionflow: A model-agnostic visualization for temporal analysis of classifier confusion. *IEEE Trans. Vis. Comput. Graph.*, 28(2):1222–1236, 2022. 49

[52] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985. 34

[53] i2 Group. i2 Analyst's Notebook. https://i2group.com/i2-analysts-notebook, 2026. Accessed: April 29, 2026. 68

[54] *In re Facebook, Inc.* 625 S.W.3d 80 (Tex. 2021). 75

[55] Panagiotis G. Ipeirotis, Foster J. Provost, and Jing Wang. Quality management on amazon mechanical turk. In *HCOMP '10*, 2010. 48

[56] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60(5):493–502, 2004. 14, 50

[57] Vani Kanjirangat and Deepa Gupta. A study on extrinsic text plagiarism detection techniques and tools. *Journal of Engineering Science and Technology Review*, 9:150–164, 10 2016. 15

[58] Mayank Kejriwal, Jiayuan Ding, Runqi Shao, Anoop Kumar, and Pedro A. Szekely. Flagit: A system for minimally supervised human trafficking indicator mining, 2017. 7, 14, 48

[59] Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Towards Parameter-Free Data Mining. In *KDD*, 2004. 16

[60] Todd Kulesza, Margaret M. Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *IUI*, pages 126–137. ACM, 2015. 50

[61] Aayushi Kulshrestha. *Detection of Organized Activity in Online Escort Advertisements*. McGill University (Canada), 2021. 2, 15

[62] La Sortie. La sortie - supporting women who have experienced sex trafficking, 2026. Accessed: 2026-05-01. 75

[63] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, page II–1188–II–1196, 2014. 15, 34

[64] Christopher Lee, Catherine Grasso, and Mark F Sharlow. Multiple sequence alignment using partial order graphs. *Bioinformatics*, 18(3):452–464, 2002. 15, 25

[65] Meng-Chieh Lee, Catalina Vajiac, Aayushi Kulshrestha, Sacha Levy, Namyong Park, Cara Jones, Reihaneh Rabbany, and Christos Faloutsos. InfoShield: generalizable information-theoretic human-trafficking detection. In *2021 37th IEEE International Conference on Data Engineering (ICDE)*. IEEE, 2021. 16

[66] Meng-Chieh Lee, Catalina Vajiac, Aayushi Kulshrestha, Sacha Levy, Namyong Park, Cara Jones, Reihaneh Rabbany, and Christos Faloutsos. INFOSHIELD: generalizable information-theoretic human-trafficking detection. In *ICDE*, pages 1116–1127. IEEE, 2021. 48, 50

[67] L. Li, O. Simek, A. Lai, M. Daggett, C. K. Dagli, and C. Jones. Detection and characterization of human trafficking networks using unsupervised scalable text template matching. In *IEEE Big Data*, pages 3111–3120, 2018. 14, 17

[68] Lin Li, Olga Simek, Angela Lai, Matthew P. Daggett, Charlie K. Dagli, and Cara Jones. Detection and characterization of human trafficking networks using unsupervised scalable text template matching. In *IEEE BigData*, pages 3111–3120. IEEE, 2018. 48

[69] Ming-Ling Lo and Chinya V. Ravishankar. Spatial Joins Using Seeded Trees. *SIGMOD*, pages 209–220, May 24-27 1994. 15

[70] Vitor Martins, D. Fonte, Pedro Rangel Henriques, and Daniela da Cruz. Plagiarism detection: A tool survey and comparison. *OpenAccess Series in Informatics*, 38:143–158, 01 2014. 15

[71] Yasuko Matsubara, Yasushi Sakurai, and Christos Faloutsos. AutoPlait: automatic mining of co-evolving time sequences. In *SIGMOD*, pages 193–204, 2014. 16

[72] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017. 15, 17, 34

[73] M. Mehta, R. Agrawal, and J. Rissanen. SLIQ: A fast scalable classifier for data mining. In *EDBT*, Mar. 1996. 16

[74] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 15, 34

[75] Chirag Nagpal, Kyle Miller, Benedikt Boecking, and Artur Dubrawski. An entity resolution approach to isolate instances of human trafficking online. In *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP*, pages 77–84, 2017. 14

[76] Chirag Nagpal, Kyle Miller, Benedikt Boecking, and Artur Dubrawski. An entity resolution approach to isolate instances of human trafficking online. In *NUT@EMNLP*, pages 77–84. Association for Computational Linguistics, 2017. 48

[77] Pratheeksha Nair, Javin Liu, Catalina Vajiac, Andreas Olligschlaeger, Duen Horng Chau, Mirela Cazzolato, Cara Jones, Christos Faloutsos, and Reihaneh Rabbany. T-net: weakly supervised graph learning for combatting human trafficking. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, pages 22276–22284, 2024. 15, 76

[78] National Institute of Justice. Unconventional wisdom: Research shakes up assumptions about sex trafficking clues in online escort ads. https://nij.ojp.gov/topics/articles/unconventional-wisdom-research-shakes-assumptions-about-sex-trafficking-clues, May 2024. Accessed: 2024-05-22. 6

[79] United Nations News. Traffickers abusing online technology, un crime prevention agency warns. https://news.un.org/en/story/2021/10/1104392, 2021. 45

[80] Olivia O'Connell. Manhattan to stop prosecuting prostitution, dismissing cases dating back decades. https://www.independent.co.uk/news/world/americas/manhattan-prostitution-prosecution-cyrus-vance-b1835256.html, 2021. 64

[81] Renata M O'Donnell. Challenging racist predictive policing algorithms under the equal protection clause. *NYUL Rev.*, 94:544, 2019. 64

[82] U.S. Department of Justice. What is human trafficking. https://www.dhs.gov/blue-campaign/what-human-trafficking, 2023. 1

[83] United Nations Office on Drugs and Crime. Global report on trafficking in persons. https://www.unodc.org/documents/data-and-analysis/tip/2021/GLOTiP_2020_15jan_web.pdf, 2020. 64

[84] International Labour Organization. Global estimates of modern slavery: forced labour and forced marriage. *International Labour Organization*, 2017. 1, 45

[85] International Labour Organization. Global estimates of modern slavery: forced labour and forced marriage. *International Labour Organization*, 2021. 1

[86] Polaris. 2020 us national human trafficking hotline statistics. https://polarisproject.org/2020-us-national-human-trafficking-hotline-statistics/, 2021. 8, 45, 49

[87] Rebecca S. Portnoff, Danny Yuxing Huang, Periwinkle Doerfler, Sadia Afroz, and Damon McCoy. Backpage and bitcoin: Uncovering human traffickers. In *KDD*, pages 1595–1604. ACM, 2017. 14, 48

[88] Reihaneh Rabbany, David Bayani, and Artur Dubrawski. Active search of connections for case building and combating human trafficking. In *KDD*, pages 2120–2129. ACM, 2018. 14, 48

[89] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, September 1978. 8, 16, 50

[90] Jorma Rissanen. A Universal Prior for Integers and Estimation by Minimum Description Length. *Ann. Statist.*, 11(2):416–431, 1983. 20

[91] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics. 39

[92] Stephanie Rosenthal and Anind K. Dey. Towards maximizing the accuracy of human-labeled sensor data. In *IUI '10*, 2010. 49

[93] Lisa Lambert Sarah N. Lynch. https://www.reuters.com/article/us-usa-backpage-justice/sex-ads-website-backpage-shut-down-by-u-s-authorities-idUSKCN1HD2QP, 2018. 48

[94] Douglas Schuler and Aki Namioka. *Participatory design: Principles and practices*. CRC Press, 1993. 49

[95] Neil Shah, Hemank Lamba, Alex Beutel, and Christos Faloutsos. The many faces of link fraud. In *ICDM*, pages 1069–1074. IEEE Computer Society, 2017. 14

[96] Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. Combating fake news: A survey on identification and mitigation techniques. *ACM TIST*, 10(3):1–42, 2019. 14

[97] Siwei Shen, Dragomir R Radev, Agam Patel, and Güneş Erkan. Adding syntax to dynamic programming for aligning comparable texts for the generation of paraphrases. In *COLING/ACL*, pages 747–754, 2006. 15, 17

[98] Malcolm K Sparrow. The application of network analysis to criminal intelligence: An assessment of the prospects. *Social networks*, 13(3):251–274, 1991. 68

[99] Yunjia Sun, Edward Lank, and Michael A. Terry. Label-and-learn: Visualizing the likelihood of machine learning classifier's success during data labeling. In *IUI*, pages 523–534. ACM, 2017. 49

[100] Edmund Tong, Amir Zadeh, Cara Jones, and Louis-Philippe Morency. Combating human trafficking with multimodal deep models. In *ACL (1)*, pages 1547–1556. Association for Computational Linguistics, 2017. 7, 48

[101] Edmund Tong, Amir Zadeh, Cara Jones, and Louis-Philippe Morency. Combating human trafficking with multimodal deep models. In *ACL*, pages 1547–1556, Vancouver, Canada, 2017. 14, 17, 33, 34, 36

[102] UNODC. Global report on trafficking in persons. *United Nations publication Sales No. E.20.IV.3*, 2020. 75

[103] UNODC. Global report on trafficking in persons. *United Nations publication Sales no.: E.23.IV.1*, 2022. 1, 75

[104] UNODC. Global report on trafficking in persons. *United Nations publication, Sales no.: E. 24.*, 2024. 8

[105] Joshua Uyheng and Kathleen M. Carley. Bots and online hate during the COVID-19 pandemic: case studies in the United States and the Philippines. *Journal of Computational Social Science*, 3(2):445–468, November 2020. 65

[106] Catalina Vajiac, Duen Horng, † Chau, Andreas M. Olligschlaeger, Pratheeksha Nair, Meng-Chieh Lee, Mirela Teixeira Cazzolato, Reihaneh Rabbany, and Christos Faloutsos. Trafficboard: Digital spatio-temporal pinboard for human trafficking detection. *Poster, IEEE VIS*, 2023. 68

[107] Jarke J. van Wijk. The value of visualization. *VIS 05. IEEE Visualization, 2005.*, pages 79–86, 2005. 49

[108] Longshaokan Wang, Eric Laber, Yeng Saanchi, and Sherrie Caltagirone. Sex trafficking detection with ordinal regression neural networks. *arXiv preprint arXiv:1908.05434*, 2019. 7, 48

[109] Chao Yang, Robert Chandler Harkreader, and Guofei Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Trans. Inf. Forensics Secur.*, 8(8):1280–1293, 2013. 34, 36

[110] Lei Zhang, Yan Tong, and Qiang Ji. Active image labeling and its application to facial action labeling. In *ECCV*, 2008. 48

[111] Xinyi Zhou and Reza Zafarani. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5):1–40, 2020. 14